

**TRAFİK KAZALARININ
MEKANSAL ANALİZİNDE KULLANILACAK
BİR YAZILIMIN GELİŞTİRİLMESİ**

Hasan Korkut ÖZKAN

**DOKTORA TEZİ
KAZALARIN ÇEVRESEL VE TEKNİK ARAŞTIRMASI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**HAZİRAN 2011
ANKARA**

Hasan Korkut ÖZKAN tarafından hazırlanan “TRAFİK KAZALARININ MEKANSAL ANALİZİNDE KULLANILACAK BİR YAZILIMIN GELİŞTİRİLMESİ” adlı bu tezin Doktora tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Kürşat ÇUBUK

.....

Tez Danışmanı, İnşaat Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Kazaların Çevresel ve Teknik Araştırması Anabilim Dalında Doktora tezi olarak kabul edilmiştir.

Doç. Dr. Hülâgü KAPLAN

.....

Şehir Bölge Planlama Anabilim Dalı, G.Ü.

Yrd. Doç. Dr. Kürşat ÇUBUK

.....

İnşaat Mühendisliği Anabilim Dalı, G.Ü.

Doç. Dr. Mehmet Ali AKÇAYOL

.....

Bilgisayar Mühendisliği Anabilim Dalı, G.Ü.

Yrd. Doç. Dr. Ebru ARIKAN ÖZTÜRK

.....

İnşaat Mühendisliği Anabilim Dalı, G.Ü.

Yrd. Doç. Dr. Yusuf DEMİREL

.....

İnşaat Mühendisliği Anabilim Dalı, G.Ü.

Tarih: 21/06/2011

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Doktora derecesini onamıştır.

Prof. Dr. Bilal TOKLU

.....

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Hasan Korkut ÖZKAN

TRAFİK KAZALARININ MEKANSAL ANALİZİNDE KULLANILACAK BİR YAZILIMIN GELİŞTİRİLMESİ

(Doktora Tezi)

Hasan Korkut ÖZKAN

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Haziran 2011

ÖZET

Trafik kazalarının mekansal analizleri ile elde edilebilecek veriler, kazaların önlenmesi amacı ile gerçekleştirilen çalışmalar için oldukça değerlidir. Bu çalışmada kaza noktalarının kendi arasındaki ilişkiyi (otokorelasyon), mekansal sınıflar ve kazaların yol ağı ile olan ilişkisini sorgulayabilmek için; trafik kazalarının mekansal analizinde kullanılacak bir metot geliştirilmiş ve metoda uygun bir yazılım (Span) üretilmiştir. Geliştirilen yazılım; düzlem ve yol ağı üzerinde nokta desen analizlerini gerçekleştirebilmekte, en yakın komşu indeksi, global, çapraz ve lokal K değerlerini hesaplayabilmektedir. Yazılım, birim testlerini takiben gerçek veriler ile, Ankara Batıkent bölgesinde meydana gelen trafik kazalarının mekansal özelliklerinin dönemsel farklılıklarını belirlemek üzere test edilmiştir. Bu test sonuçları ile, Batıkent bölgesinde okul çevresinde alınan trafik tedbirlerinin kazaların dağılımı üzerindeki anlamlı etkisi olduğu bulunmuştur ve bu sonuç literatüre önemli bir katkı sağlayan orijinal bulgudur.

Bilim Kodu : 911.1.014
Anahtar Kelimeler : Bilgisayar yazılımı, Coğrafi Bilgi Sistemi, Kaza analizi, Mekansal Analiz, Nokta Desen Analizi
Sayfa Adedi : 186
Tez Yöneticisi : Yrd. Doç. Dr. Kürşat ÇUBUK

**DEVELOPMENT OF A SOFTWARE APPLICATION FOR
SPATIAL ANALYSIS OF TRAFFIC ACCIDENTS**

(PhD. Thesis)

Hasan Korkut ÖZKAN

**GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY**

June 2011

ABSTRACT

The data, generated with spatial analysis of traffic accident, is valuable for traffic accidents preventing practices. In this study; a methodology and application software (Span) developed for querying autocorrelation between traffic accident point, correlation between traffic accident and spatial classes and also road network to spatial analyze of traffic accidents. Developed application software should make of point pattern analysis and calculate nearest neighborhood index, global, cross and local Ripley K values in both on planner and road network. After unit tests the application software tested with real world data to identification of periodic changes on spatial characteristics' of Batıkents' accidents. Batıkent study results shows the application software should be generate very valuable and new information like significant effects of traffic safety practices for school area on distribution traffic accidents in school area.

Science Code : 911.1.014

**Key Words :Computer software, Geographic Information System,
Accident analysis, Spatial analysis, Point pattern analysis**

Page Number: 186

Adviser : Asist. Prof. Dr. Kürşat ÇUBUK

TEŞEKKÜR

Çalışmalarım esnasında değerli yardımlarını esirgemeyen danışmanım Yrd. Doç. Dr. Kürşat ÇUBUK'a, bilimsel çalışmalar ve trafik güvenliği konusunda beni yönlendiren Doç.Dr. Süleyman IŞILDAR'a ve tüm çalışma süresince destek veren kardeşim Muhammed ÖZKAN'a ve aileme teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZET	iv
ABSTRACT	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	xi
ŞEKİLLERİN LİSTESİ	xii
HARİTALARIN LİSTESİ.....	xiv
SİMGELER VE KISALTMALAR.....	xv
1. GİRİŞ	1
2. MEKANSAL ANALİZ.....	10
2.1. Veri Tipleri.....	11
2.1.1. Nominal tip veriler.....	11
2.1.2. Sıralı tip veriler	12
2.1.3. Aralık tipi veriler	12
2.1.4. Sürekli tip veriler	12
2.2. Mekansal Nesneler.....	13
2.2.1. Nokta.....	13
2.2.2. Çizgi.....	13
2.2.3. Poligon.....	14
2.3. Mekansal İşlemler.....	14
2.4. Tanımlayıcı Nokta Desen Hesaplamaları	15
2.4.1. Noktalararası uzaklığın standart sapması	16

	Sayfa
2.4.2. Ortalama merkez.....	17
2.4.3. Mekansal standart sapma elipsi	19
2.4.4. Çarpıklık	20
2.4.5. Basıklık	20
2.5. Mekansal Analiz Yöntemleri	21
2.6. Veri Esaslı Nokta Desen Analizleri	22
2.6.1. Kare analizleri.....	24
2.6.2. Çekirdek yoğunluk tahmini	25
2.6.3. En yakın komşu indeksi.....	27
2.6.4. K fonksiyonu	29
2.7. Nokta Desenlerinin Modellenmesi	35
2.8. Mekansal Veri Madenciliği.....	38
3. MEKANSAL ANALİZ ARACI	42
3.1. Veri Gereksinimleri	43
3.1.1. Kaza öncesi veriler	44
3.1.2. Kaza anı verileri.....	45
3.1.3. Kaza sonrası veriler	46
3.1.4. Analiz öncesi veriler	47
3.1.5. Analiz verileri	48
3.1.6. Analiz sonuçları	48
3.1.7. Analiz gösterim verileri	49
3.2. Fonksiyonel Gereksinimler	50
3.2.1. Mekansal analiz çalışmasına ait verilerin saklanması	50

Sayfa

3.2.2. Kaza verilerinin saklanması.....	52
3.2.3. Kazaların filtrelenmesi.....	52
3.2.4. Yol katmanının üretilmesi	52
3.2.5. Trafik kazalarının mekansal dağılım sonuçlarının üretilmesi ve gösterimi	53
3.3. Kullanıcı Etkileşimi	55
3.4. Uygulama Yazılımı Gerçekleştirme Süreci	63
4. BATIKENT’TE MEYDANA GELEN ÖLÜMLÜ ve YARALAMALI TRAFİK KAZALARININ MEKANSAL NİTELİKLERİNDEKİ DÖNEMSEL DEĞİŞİKLİKLER.....	65
4.1. Materyal	65
4.2. Metot	70
4.3. Bulgular.....	71
5. SONUÇ VE ÖNERİLER	86
KAYNAKLAR	90
EKLER.....	94
EK-1 Dönemsel çapraz K grafikleri (alışveriş merkezleri).....	95
EK-2 Dönemsel çapraz K grafikleri (sağlık merkezleri)	100
EK-3 Dönemsel çapraz K grafikleri (okullar).....	105
EK-4 Geliştirilen yazılımında kullanılan veri tipleri	110
EK-5 Geliştirilen yazılımın mimarisi.....	118
EK-6 Geliştirilen uygulama yazılımının mekansal analiz fonksiyonları	132
EK-7 Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu.....	142

Sayfa

EK-8 Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu	155
ÖZGEÇMİŞ	186

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Mekansal nesnelere uygulanan analiz türleri.....	21
Çizelge 2.2. İlişkilendirme için kullanılacak örnek kaza döküm tablosu	39
Çizelge 2.3. Örnek verilere dayanan ilişkilendirme çalışması sonuç tablosu.....	40
Çizelge 3.1. Mekansal analiz fonksiyonlarının analiz öncesi veri ihtiyaç tablosu.....	47
Çizelge 4.1. Yıllara göre Batıkent’te meydana gelen ölümlü ve yaralamalı trafik kazalarının kazaya karışanlara ve sonuçlarına göre dağılımı.....	72
Çizelge 4.2. Yıllara göre Batıkent’te meydana gelen ölümlü ve yaralamalı trafik kazalarına karışan araçların sayısı ve yüzdeleri	73
Çizelge 4.3. Yıllara göre Batıkent’te meydana gelen ölümlü ve yaralamalı trafik kazaları arasındaki mesafelere ait tanımlayıcı istatistikler.....	74
Çizelge 4.4. Yıllara göre Batıkent’te meydana gelen ölümlü ve yaralamalı trafik kazaları arasındaki mesafe dağılımı basıklık değerleri ve dağılım	76

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Nokta desen dağılımları	22
Şekil 2.2. Kare analizi	24
Şekil 2.3. Ripley K fonksiyonunun uygulanışı	31
Şekil 2.4. Ripley K fonksiyonu ve rastlantısal sonuçlar ile kıyaslanması	33
Şekil 3.1. Span uygulaması menü yapısı akış diyagramı	55
Şekil 3.2. Span uygulama yazılımı açılış formu	56
Şekil 3.3. Span uygulama yazılımı veri düzenleme ekranı	57
Şekil 3.4. Span uygulama yazılımı kaza verisi giriş ekranı	58
Şekil 3.5. Span uygulama yazılımı kaza verisi giriş ekranı	59
Şekil 3.6. Span uygulama yazılımı tanımlayıcı nokta desen istatistikleri ekranı.....	60
Şekil 3.7. Span uygulama yazılımı mekansal analiz ekranında global K değerlerinin grafik ile gösterimi	61
Şekil 3.8. Span uygulama yazılımı mekansal analiz ekranında yerel K değerlerinin harita ile gösterimi	62
Şekil 4.1. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının aylara ve yıllara göre dağılımı	72
Şekil 4.2. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının düzlem ve yolağı üzerindeki mesafe dağılımlarını gösterir histogramlar .	75
Şekil 4.3. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının düzlem ve yolağı üzerindeki en yakın indeksleri	77
Şekil 4.4. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının yıllara göre düzlemde Global K değerlerini gösterir grafik	77
Şekil 4.5. Batıkent'te 2007 yılında meydana gelen ölümlü ve yaralamalı trafik kazalarının yol ağı üzerinde Global K değerlerini gösterir grafik.....	78

Şekil	Sayfa
Şekil 4.6. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının sağlık merkezleri çevresinde kümelenme tespit edilen dönemlere ait Çapraz Ripley K değerlerini gösterir grafik	79
Şekil 4.7. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının yıllara göre sağlık merkezleri esas alınarak hesaplanan Çapraz K değerlerini gösterir grafik	79
Şekil 4.8. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının alışveriş merkezleri çevresinde kümelenme tespit edilen dönemlere ait Çapraz K değerlerini gösterir grafik	80
Şekil 4.9. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının yıllara göre alışveriş merkezleri esas alınarak hesaplanan çapraz K değerlerini gösterir grafik	81
Şekil 4.10. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının okullar çevresinde kümelenme tespit edilen dönemlere ait Çapraz K değerlerini gösterir grafik	81
Şekil 4.11. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının yıllara göre okullar esas alınarak hesaplanan çapraz K değerlerini gösterir grafik.....	82

HARİTALARIN LİSTESİ

Harita	Sayfa
Harita 4.1. Mekansal analiz çalışmasında ele alınan ölümlü ve yaralamalı trafik kazalarının yol ağı üzerinde gösterimi.....	67
Harita 4.2. Mekansal analiz çalışmasında ele alınan yol ağı ve çalışma alanı.....	68
Harita 4.3. Mekansal analiz çalışmasında ele alınan mekansal sınıflar ve çalışma alanı.....	69
Harita 4.4. Batıkent’te meydana gelen ölümlü ve yaralamalı trafik kazalarının yıllara göre kaza merkezlerini gösterir harita.....	83
Harita 4.5. Batıkent’te meydana gelen ölümlü ve yaralamalı trafik kazalarının yıllara göre yol ağı üzerindeki kümelenmelerini gösterir haritalar	84

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
kern	Çekirdek fonksiyonu
SQL	Yapısal sorgulama dili
WGS	Dünya Jeodezi Sistemi
WHO	Dünya Sağlık Örgütü

1. GİRİŞ

Verdikleri zarar dikkate alındığında modern toplumların en önemli sorunlarından birinin trafik kazaları olduğu bilinmektedir. Günümüzde endüstrileşme ile artan ulaşım ihtiyacı; kara, deniz, hava yolu veya boru hatları üzerinden yapılan taşımacılık ile karşılanmaktadır. Yük ve yolcu taşımacılığının türler arasında dağılımı ve ulaşım ihtiyacının boyutu ülkeden ülkeye farklılık gösterse de; karayolunda meydana gelen trafik kazaları yaygın ve dramatik sonuçları ile sadece Türkiye'nin değil diğer ülkelerin gündemlerini de meşgul etmektedir.

Dünya Sağlık Örgütü (WHO) verilerine göre, 2002 yılında dünyada meydana gelen ölümlerin % 2,1'inin sebebi trafik kazalarıdır. Aynı çalışmada sadece 2002 yılında meydana gelen trafik kazaları neticesinde oluşan sakatlıklar sonucu kaybedilen iş gücünün 38 milyon yıl olduğu hesaplanmıştır[1].

Türkiye'de son on yıl içerisinde Emniyet Genel Müdürlüğü ve Jandarma Genel Komutanlığı sorumluluk bölgesinde meydana gelen karayolu trafik kazaları incelendiğinde; kaza sonucu meydana gelen ölümlerde azalma olduğu görülmektedir. Ancak 2001 yılında Türkiye'de trafik kaza sayısı 442960 iken 2010 yılında bu sayı 1104388 ulaşmıştır. Kaza sonucu yaralanmaların sayısı da 2001 yılında 116202 iken 2010 yılında 211496'e çıkmıştır [2, 3, 4].

Türkiye'de resmi kurum ve kuruluşlar kanun ve yönetmeliklerle kendilerine verilen görevler kapsamında, trafik güvenliğini arttırmak ve trafik kazalarına engel olmak üzere; yol kullanıcılarının bilinçlendirilmesine, kazaların sıklıkla meydana geldiği yer ve zamanlara yönelik denetimler gerçekleştirilmesine, altyapıya yönelik mühendislik çalışmalarına, acil ve ilk yardım hizmetlerinin yaygınlaştırılmasına yönelik çalışmalar yürütmektedirler.

Trafik güvenliğine yönelik denetleme ve alt yapı gibi uygulamaların yapılacağı yer hakkında mekansal tanımlama ve sınırlamada bulunulması gerekmektedir. Özellikle

altyapıya yönelik iyileştirici mühendislik faaliyetlerinin yüksek maliyetleri projeleri, bütçe ve kaynak kullanımı açısından sınırlandırmaktadır.

Altyapıya yönelik iyileştirici faaliyetlere daha az kaynak ayrılmasının nedeni; ulaştırma alanında yatırıma yönelik çalışmaların, iyileştirme çalışmalarından daha fazla politik kabul görüyor olmasıdır.

Kurumların trafik güvenliğini sağlamaya yönelik faaliyetlerine tahsis edilmiş kaynakların sınırlı olması, projelerin önceliklendirilmesini ve ilgi grupları için özelleştirilmesini gerektirmektedir.

Önceliklendirme ve özelleştirme çalışmalarında; kaza sonucu oluşan can ve mal kaybını önlemede, alınan önlemler ile kazalar arasındaki ilişki bilimsel metotlar ile sorgulanmaktadır. Kazalar ile kaza oluşumu arasındaki ilişki nedensellikte ele alınamadığı gibi, ilişkisellik ile de ele alınamamaktadır.

Trafik güvenliğinin artırılması, altyapıya yönelik iyileştirici faaliyetlerin yanında, yatırım çalışmaları için de bir unsur olarak ele alınmalıdır. Trafik kazaları ile tasarım unsurları arasındaki ilişkinin irdelenmesi ve tespitler ışığında tasarımın güncellenmesi daha makul maliyetli ve güvenli uygulamaların ortaya koyulmasını sağlayacaktır.

Bu konuda kullanılacak bir mekansal analiz yazılımında temel olarak; kazaların birbirleri ve diğer kaza değişkenleri ile arasındaki mekansal ilişkiyi tanımlaması ve kazaların mekansal dağılımı üzerindeki etkileri konu edinen hipotezleri test edilebilmesi beklenmektedir.

Geleneksel kaza analizlerinde kaza öncesi ve kaza anına ait veriler ile kaza ve sonuçları arasındaki ilişki değerlendirilmektedir. Bu çalışmalarda kazaların sebep ve sonuçlarını açıklamak üzere geliştirilen hipotezler, hipotez testleri ile denenir. Kaza analizlerinde kurulan sıfır hipotez belirlenen örneklem ile ana kütle ortalaması

arasındaki farkın anlamlı olup olmadığının sorgulanması şeklindedir. Mekansal olmayan analizlerde, geliştirilen hipoteze uygun modelde unsurlar arası bir ilişkinin olmadığı kabul edilerek çalışmaya başlanır.

Oysa Waldo Tobler; mekansal olarak her şeyin birbiri ile ilintili olduğunu, ancak birbirine yakın olan şeylerin uzak olanlara göre daha ilintili olduğunu ortaya koymuştur [5]. Bu ifade coğrafyanın birinci yasası olarak bilinir.

Tobler yasası dikkate alınarak yapılan trafik kaza analizleri neticesinde, trafik güvenliğine etkiyen unsurlar daha geniş bir ölçekte incelenmekte algılanmakta ve anlayışta nedensellikten çok ilişkiselliğe dayanan farklı bir açı elde edilmektedir. Mekansal olarak tanımlanabilen verilerin analiz çalışmalarında kullanılması ve bu verilerin temsil ettiği unsurlar ile trafik kazalarının dağılımı üzerindeki etkinin sorgulanması trafik kazalarının mekansal analizinin temelini oluşturur.

Trafik kazalarının belirli bir parametre uzayında (bu çalışmada mekanda), yine o uzayın parametreleri cinsinden dağılımına yönelik bir olasılık fonksiyonu mevcuttur. Kazaların mekansal analizi bu fonksiyon parametrelerinin irdelenmesidir.

Literatürde mekansal analizin bir olayı açıklamak üzere kullanılışına John Snow'un 1854 yılında Londra meydana gelen kolera salgını üzerinde yaptığı çalışmalarda rastlanmaktadır. Snow kolera hastalarının yaşadıkları yerleri Londra haritası üzerinde işaretleyerek, hastaların şehre su sağlayan pompalar etrafında yoğunlaştığını görmüştür. Daha sonra gerçekleştirdiği çalışmalarda, atık suların nehre döküldüğü yerdeki pompalar çevresinde hastalığın şiddetinin de arttığını, ölüm oranlarının yükseldiğini belirlemiştir. Bu çalışma bir fenomenin mekan verileri ile ilişkilendirilerek açıklandığı literatüre geçmiş ilk mekansal analiz çalışmasıdır [6].

Suç analizinde ve salgın hastalıkların incelenmesinde mekansal analiz metotlarının kullanımı yaklaşık yüz elli yıllık bir geçmişe sahiptir. Trafik kazaları üzerine

gerçekleştirilen çalışmalarda mekansal analiz metotlarının kullanımının ise yeni bir uygulama olduğu görülmektedir.

Black tarafından 1991 yılında gerçekleştirilen çalışmada trafik kazalarının oluşumunun trafik yoğunluğu ile ilişkili olarak, zaman ve mekanda kümелendiği görülmüş; kar, sis gibi doğal çevresel faktörlerin yanında karayolu ağının karakteristiğini yansıtan giriş, çıkış noktaları, yetersiz tasarım ve bakım gibi faktörlerin de etkisi ile kazaların desenler meydana getirdiği tespit edilmiştir [7].

1995 yılında Levine ve arkadaşları tarafından trafik kazaları noktasal olaylar olarak ele alınmış;

- Çevre özelliklerinin kıyaslanması,
- Karayolu sistemi ile kazalar arasındaki ilişkinin araştırılması,
- Kazaların belirli alan veya koridorlarda sosyal ve ekolojik açıdan incelenmesi gerçekleştirilmiş,
- Yerel trafik kazası desenlerinin küresel ölçekte bir başka desen oluşturup oluşturmadığı kontrol edilerek, kazaların mekansal dağılımları konu edinilmiştir. Levine bu çalışmada kazayı en yakın kavşağa atayarak değerlendirmiştir. Mekansal veri işleme metotları kullanılarak gerçekleştirilen çalışmada, kazaların yol kesitlerine atanması kaza ile çevresel faktörler arasındaki ilişkiyi daha tanımlayıcı yaparken, modelin tahmin edebilme yeteneğini sınırlandırmıştır [8].

Jones ve arkadaşları tarafından nokta desen analizi uygulanarak geliştirilen kaza modelinde ele alınan; yaş, kazazede türü ve kazazede sayısı değişkenlerinin model üzerindeki etkisi göz ardı edildiğinde, kazaya ait diğer değişkenlerin kümелendiği tespit edilmiştir [9].

Yamada; 2004 yılında yayınlanan çalışması ile; kazaların mekansal dağılımlarının yol ağı dikkate alınarak incelenmesi neticesinde daha fazla açıklayıcılığa sahip bulgulara ulaşılabileceğini ortaya koymuştur [10].

Miaou ve arkadaşları ise 2003 yılında, yerleşim bölgelerindeki trafik kaza riskinin belirlenebilmesi için Poisson dağılımını esas alarak Bayes yöntemini kullanmıştır. Kaza riski haritasının oluşturulmasında kazaları, yol ağı üzerinde bulundukları konuma (kavşak içerisinde, kavşağa yakın, kavşak haricinde, otoyollarda) göre değerlendirmişler ve yol ağını kaza riskine göre sınıflandırmışlardır [11].

Noland ise; 2004 yılında arazi kullanımını (iş ve nüfus yoğunluğu), yol niteliğini, demografik faktörleri ve trafik akışını dikkate alarak trafik kazalarını açıklamaya çalışmıştır. Trafik kaza değişkenin sayıma dayalı olması nedeni ile normal dağılım yerine Poisson dağılımını kullanarak bir regresyon modeli oluşturmuştur [12].

Brüksel dışında meydana gelen trafik kazalarının analizine yönelik, 2003 yılında Eckhardt ve Thomas tarafından yapılan çalışmada ise her 100 metrede kaza olma riskini ortaya koyan bir model oluşturulmuştur. Çalışma neticesinde; il, ilçe, mahalle, köy gibi idari yapılanmalar dikkate alınarak kazaların yoğunluğunun belirlenmesinin ve değerlendirilmesinin zorluğuna dikkat çekilmiştir [13]. Sürücü davranışı, ulaşım verileri gibi diğer faktörlerin ele alınmadığı çalışmada, geliştirilen model ile çevre faktörlerinin kaza sayısındaki değişimin %5'ni, altyapının ise kaza sayısındaki değişimin %21'ni açıkladığı görülmüştür [13].

Aguero ve Jovanis ise 2006 yılında yayınladıkları çalışmalarında 1996-2000 yılları arasında Pensilvanya'da meydana gelen trafik kazalarının zaman ve mekan üzerindeki dağılımını ilçeler bazında Bayes yöntemini kullanarak açıklamaya çalışmışlar ve kazaların dağılımında negatif binom dağılımını test etmişlerdir [14].

Aryaija ve arkadaşları tarafından 2007 yılında gerçekleştirilen çalışmada ise doğrusal modelleme yapılarak; arazi kullanımının çocuk yayaaların karıştığı trafik kazaları üzerindeki etkisi değerlendirilmiştir [15].

Meliker ve arkadaşları 2004 yılında yayınladıkları çalışmada; alkol satan yerler ile alkolle ilişkili trafik kazaları arasında bir ilişkiyi mesafe analizlerine dayandırarak açıklamışlardır [16].

Türkiye’de yapılan çalışmalar da, uluslar arası çalışmalarla eşgüdümlü olarak trafik kazalarının modellenmesine ve kara noktaların belirlenmesine yöneliktir.

Geyik, tarafından 2001 yılında kaza kara noktalarının tespiti ve değerlendirilmesine yönelik yeni bir yöntemin oluşturulması hakkında bir yüksek lisans tezi hazırlanmıştır [17].

Bektaş ise 2002 yılında daha sınırlayıcı bir mekanı ele alarak trafik kazalarının modellenmesine yönelik bir doktora tezi sunmuştur [18].

Yazar tarafından 2006 yılında yapılan çalışmada ise, turizmin kazalar üzerindeki zamansal ve mekansal etkisi araştırılarak tanımlayıcı mekansal istatistikler ile kazaların mekansal dağılımları incelenmiştir [19].

Bu çalışma ile ele alınan mekansal desen analiz yöntemleri, daha çok hastalıkların epidemik özelliklerinin veya bitki örtüsünün mekansal dağılımının araştırılmasında kullanılan yöntemlerdir.

2004 yılında Turnbull, bitki tohumlarının rekabet ve kolonizasyonunu konu alan çalışmasında bitkilerin mekansal desenlerini araştırmıştır [20].

Besag ve Newell ise 1991 yılında, nadir verilerin yığılması kontrol için geliştirilen algoritmayı makro anlamda hastalıkların mekansal dağılımına uygulamıştır [21].

Trafik kazalarının mekansal dağılımlarının incelenmesine yönelik 2005 yılında Güvenal, Çabuk ve Yavuz tarafından yapılan çalışmada ise, coğrafi bilgi sistemleri

kullanılarak kazaların taban haritaları üzerinde gösterimi yapılmış, kaza yoğunluğu verisi (genellikle bir nokta etrafında üretilen tampon bölgede yer alan kaza sayısı) karayolunun yapısal özellikleri ile ilişkilendirilmiştir [22].

Trafik kazaları üzerine gerçekleştirilen literatür araştırması neticesinde mekansal verilerin analiz çalışmalarında mekansal dağılımların Poisson, negatif binom veya Bernoulli dağılımları ile açıklandığı görülmektedir.

Kanuni uygulamalar ve düzenlemeler sonucunda trafik kazalarının nedenlerinin yol kullanıcılarının kusurlarına dayandırılması yönünde bir bakış açısının olduğu söylenebilir. Bu bakış açısı aynı zamanda kaza verilerinin toplanmasına ve değerlendirilmesine de yansımıştır.

Türkiye’de trafik uygulamaları kanuni açıdan 13.10.1983 tarihli ve 2918 sayılı Karayolları Trafik Kanunu ile düzenlenmiş ve çeşitli tarihlerde yapılan kanun hükmünde kararname ve kanunlar ile uygulamaya güncellik kazandırılmıştır.

Trafik kazalarının sonuçları hakkındaki bilgi ise; yaralamalı ve ölümlü trafik kazaları sonrasında trafik zabıtasının hazırlamış olduğu Kaza Tespit Tutanaklarına dayanmaktadır. Hazırlanan tutanaklar Türkiye’de ölümlü ve yaralamalı kazalar hakkında yapılan istatistikî çalışmaların temelini oluşturmaktadır.

Mevcut Kaza Tespit Tutanaklarının nedensellik esas alınarak hazırlanıyor ve üretiliyor olması, trafik güvenliğini çalışmalarında kaza verilerinin değerlendirebilmesi için ayrıca veri toplama metotları geliştirmesi gerektirmektedir.

Bu çalışmada, ile trafik kazalarının mekansal veriler de dikkate alınarak analiz edilmesi ile kazaları açıklamaya ve önlemeye yönelik yeni çözümlerin geliştirilmesi amaçlanmaktadır.

Tez çalışmasında, trafik kazalarını nedensellik ile ilişkisellik ile de açıklamak üzere; kaza noktalarının oluşturduğu desenleri inceleyen ve bu inceleme bulgularından anlamlı sonuçlar üretebilen trafik kazalarının analizinde kullanılabilecek yeni bir yöntem geliştirilmiştir. Çalışmada aynı zamanda metodu kullanan bir yazılımda üretilmiştir.

Üretilen yazılım; trafik kazalarının dağılımı üzerinde, kazaların kendisinin (oto korelasyon), mekansal sınıfların ve yol ağının etkisini ortaya koyarken, modelleme ve veri madenciliğinde kullanılabilecek ara verileri de üretmektedir.

Yazılım temel iki farklı analize imkan sağlamaktadır. Yazılım kullanılarak trafik kazalarının noktasal dağılım ve yoğunluklarının incelenmesine yönelik tanımlayıcı istatistiki veriler üretilmekte ve düzlem veya yol ağı üzerinde kazaların, birbirleri arasında veya kontrol değişkenleri aralarındaki ilişki kontrol edilebilmektedir.

Çalışma bu kapsamı ile; istatistik, bilgisayar mühendisliği ve kaza analizi disiplinlerinin kesişme noktasında, kazaların mekansal analizinde özgün bir metod kullanımını mümkün kılan bir uygulama yazılımının geliştirilmesi şeklinde tanımlanabilir.

Tez içerisinde; mekansal analiz yazılımının trafik güvenliği açısından önemi, nokta dağılım analizinin teorik zemini ve farklı uygulama şekilleri, yazılımın gereksinimleri ve yazılım tasarımı konularına değinilmiştir.

Geliştirilen uygulama yazılımının etkinliği ise; 2006, 2007 ve 2008 yıllarında Ankara Batıkent yerleşim bölgesinde meydana gelen ölümlü ve yaralamalı trafik kazalarının mekansal dağılımlarının analiz edilmesi ile ele alınmıştır. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının mekansal dağılımına yönelik bulgular, tez çalışması içerisinde değerlendirilerek yorumlanmıştır.

Geliştirilen uygulama yazılımının performans ve kullanımı ile ilgili hususlar ve daha sonraki çalışmalar için önerilen konular, tez çalışmasının sonuç ve değerlendirme kısmında belirtilmiştir.

Batıkent uygulamasında erişilen çapraz ilişkileri açıklayan grafikler ve uygulama yazılımının mimarisi ve temel sınıflarına ait kod dökümleri ise teze ek olarak verilmiştir.

2. MEKANSAL ANALİZ

Çevrede meydana gelen olguları açıklamak üzere merak ve ilginin artışı; inceleme ve araştırma çalışmalarında daha fazla veriyi dikkate almayı gerektirmiştir. Mekansal nitelik ve etkileşimler, Tobler yasası ile nedenselliğin inkar edilemez temellerini oluşturmaktadır.

Petrol, bakır, altın gibi yer altı kaynaklarına erişmek üzere hızla gelişen coğrafi istatistik (geostatistic) uygulamaları; aynı metotları sosyal bilimlerde de başarı ile kullanmak için araştırmacıları teşvik etmektedir.

Dünya'nın maden kaynaklarının yönetim ve işletiminde söz sahibi olan şirketler coğrafi istatistik metotlarını kullanarak Dünya kabuğunun kilometrelerce altında olan cevherlerin yerlerini belirlemekte kullandıkları modeller geliştirmektedir. Madencilikte elde edilen başarı; biyoloji, sosyoloji, politika, antropoloji, sağlık gibi alanlarda da mekansal özellikleri ve etkileşimleri dikkate alan mekansal modellerin üretilmesine neden olmuştur.

1970'de mekansal entropi konusunda başlayan çalışmalar, Cliff ve Ord'un 1981 yılında yayınladıkları "Spatial Processes: Models and Applications" kitabı ile yeni bir yön kazanmış ve eser mekansal analiz uygulamalarının basılı eğitim materyali haline gelmiştir [23].

Luc Anselin'in 1988'de yayınlanan "Spatial Econometrics: Methods and Models" isimli kitabı ise sosyal bilimler alanında bir ilke imza atarken, bu kitap sayesinde mekansal analiz çalışmaları Cliff ve Ord'un oto korelasyon yaklaşımından ayrılarak modellemeye [24] yönelmiştir. Bu değişim ile de, mekansal etkileşimlerin olguları açıklamaktaki kullanımı yaygınlaşmıştır.

Uygulamada; coğrafi istatistik ile mekansal veri analizi arasında farklılıklar görülmektedir.

Coğrafi istatistikte; bakır cevherinin kalitesi, petrol fiyat artışları gibi mekan ya da zamanla ilişkili süreklilik gösteren bir olgu ele alınır. İlişkiselilikleri eşitliklerle veya variogram modelleri ile belirlenen değişkenlerin bilinmeyen değerleri; kriging hesaplaması veya simülasyonlar ile tahmin edilir. Coğrafi istatistikte ele alınan örneklem ile olguyu açıklayan mekansal bir modelde süreklilik gösteren yüzey verisine ulaşılması hedeflenir. Örneğin her 500 metrede gerçekleştirilen toprak analizleri ile petrolü tutabilecek killi toprağın yeri öngörülerek muhtemel petrol rezervinin miktarı kestirilir.

Mekansal analizlerde ise incelenen veri; sayıma dayalıdır ve süreklilik göstermez. Örneğin bebek ölüm oranlarının illere göre dağılımı mekansal analizin inceleme alanındadır. Mekansal dağılım üzerine etki eden faktörler belirlenerek bağımlı değişkenin kestirimine ait coğrafi istatistikte kullanılan metotlara benzer metotlar kullanılır.

2.1. Veri Tipleri

Coğrafi istatistik ile mekansal analiz arasındaki temel farklılık açıklanmaya çalışılan olgunun veri tipi ile ilgilidir. Mekansal verilerin analiz ve gösteriminde verilerin hangi tipte tutuldukları önemlidir. Veri tiplerine göre analiz metotları değişiklik gösterir:

2.1.1. Nominal tip veriler

Nesnelerin çeşitli niteliklerini birbirlerine eşit veya eşit olmadıklarına göre ayıran veri tipleridir. Nominal tip veriler üzerinde aritmetik işlem yapmak mümkün değildir.

Mekansal analizlerde nominal tip veriler frekanslarına göre ayrılarak değerlendirilebilir. Örneğin trafik kazaları neticesinde ölüm veya yaralanma meydana gelmiş olması kazanın diğer bir niteliği olan konum bilgisi ile

eşleştirilebilir. Böylece elde olan diğer konum bilgileri ölümlü veya yaralamalı kaza olup olmadığına göre karşılaştırılabilir.

Nominal verilerin harita üzerinde gösterimi, farklılıkları ön planda olan renkler ile gerçekleştirilebilir.

2.1.2. Sıralı tip veriler

Eşitlikleri ve eşitliklerin kendi aralarındaki sıralamalarını tanımlayan verilerdir. Kazazedelerin eğitim seviyelerinin ifade edildiği veriler (eğitimsiz, ilkokul, lise ve üniversite gibi) sıralı verilere örnek olarak gösterilebilir.

Sıralı tip verilere ait değerler harita üzerinde bir rengin açık veya koyu tonları ile farklılaştırılarak temsil edilir.

2.1.3. Aralık tipi veriler

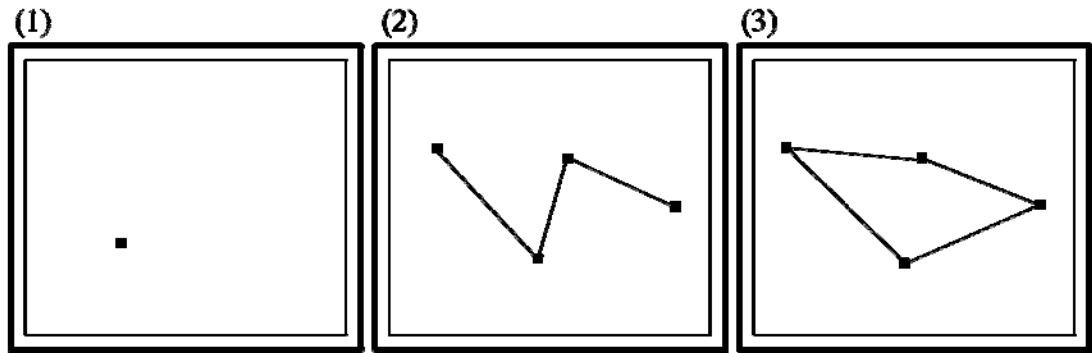
Ölçümler ile elde edilen değerleri sıraya sokarak aralıklar halinde tanımlayan verilerdir. Aralık tipi veriler için dört işlem kullanılarak aritmetik ortalama, ortanca ve tepe değerleri (mod) üretilebilir. Trafik kazalarının meydana geldiği mahalledeki hane nüfus sayısı aralık tipi veriye örnektir. Aralık tipi veriler harita üzerinde aynı rengin açık ve koyu tonlamaları ile gösterilebilir.

2.1.4. Sürekli tip veriler

Süreklilik gösteren ve mutlak değere sahip olan verilerdir. İnsanın yaşı veya toprakta bulunan kurşun miktarı örnek olarak verilebilir. Haritada gösterimleri farklı renkler kullanılarak gerçekleştirilebilir.

2.2. Mekansal Nesneler

Karşılaşılan olguyu açıklamak üzere; istatistiki çalışmalarda olduğu gibi, gerçekleştirilecek mekansal analizlerde de, veriler işlenmeden ve anlamlandırılmadan önce bir takım veri işlemlerine tabi tutulur. Mekansal analizler esnasında kullanılacak olan veriler; 3 tip mekansal nesneden en az birine atanarak vektörize edilir.



Şekil 2.1. Mekansal nesneler
1- Nokta, 2- Çizgi, 3- Poligon

2.2.1. Nokta

Mekan bilgisinin yatay ve dikey koordinatları ile düzlem üzerinde ifade edilmesidir. Trafik kazalarının olduğu yer, meteor çarpma noktaları, su kaynakları nokta ile temsil edilir. Alan, hacim, uzunluk gibi kendinden daha üst uzayda tanımlanan değerlere sahip değildir.

2.2.2. Çizgi

İki nokta arasındaki noktalar kümesidir. Ancak genellikle 2 noktanın konum bilgisi ile tanımlanır. Çalışmada yol ağını oluşturan bağlantı verileri, esasen poligon olmasına rağmen çizgi şeklinde değerlendirilmiştir.

2.2.3. Poligon

Çizgilerden meydana gelen köşelere ve alana sahip bütündür. Mahalleler, idari bölümlenmeler veya sorumluluk alanları harita üzerinde poligon ile gösterilir.

2.3. Mekansal İşlemler

Mekansal işlemler, mekansal nesnelerin birbirleri ile olan etkileşimlerini kontrol etmek üzere kullanılırlar. Bu işlemler nesnelerin mekansal tanımları üzerinden gerçekleştirilir.

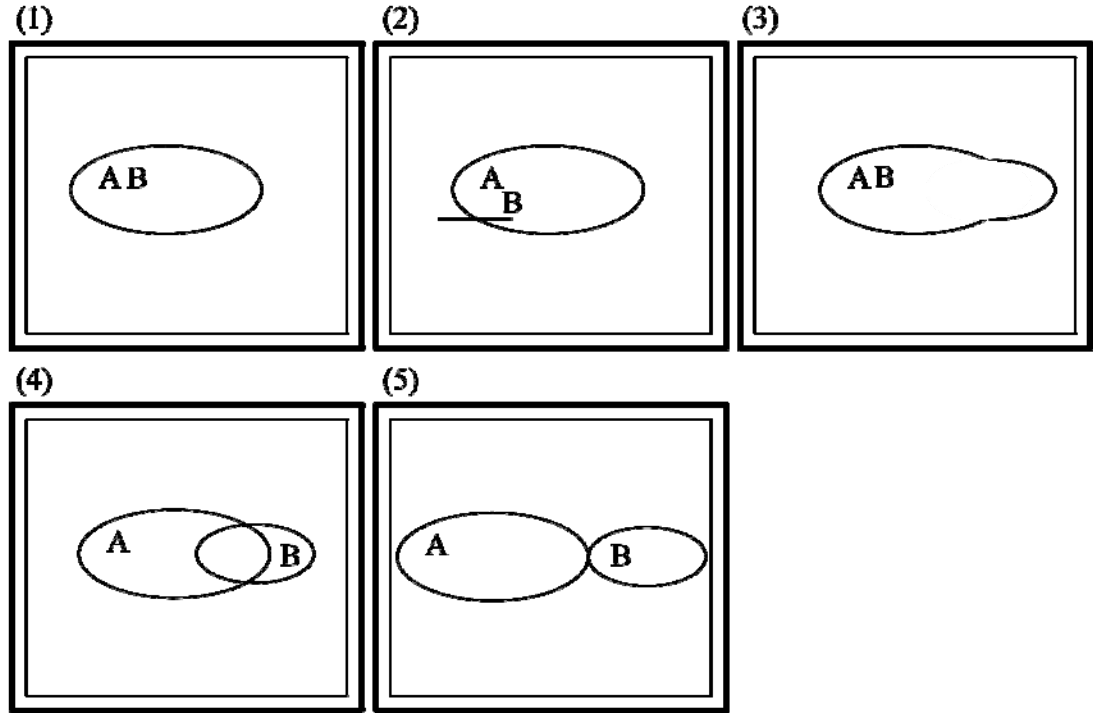
Mekansal nesnelerin geometrilerinin birbirlerine eşit olup olmadığının kontrolü eşitlik işlemi ile ortaya konulur. Mekansal nesnelerin ortak en az bir noktaya sahip olup olmadıklarının belirlenmesi için ise ayrıklık işlemi gerçekleştirilir.

Mekansal nesneler arasındaki ortak sınırların kaldırılarak aynı özniteliklerin atanması işlemine birleşme adı verilir.

Nesnelerin birbirleri üzerinde yer alan parçaların belirlenmesi kesişme olarak adlandırılır. Nesnelerin birbirlerine temas edip etmediklerinin kontrol edilmesi bağlantı işlemi ile gerçekleştirilir.

Veriler, mekansal işlemler ile analize daha uygun getirilebilir. Nokta şeklinde elde edilen bir veri poligona eşlenebilir veya bir alan için belirlenmiş öznitelik alan bir noktaya atanabilir.

Mekansal işlemler ile gerçekleştirilen veri birleştirme ve genelleme işlemi veriye uygulanabilecek olan mekansal analiz yöntemi çeşitliliği artırmaktadır. Ancak yapılan veri birleştirme ve genelleme işlemleri çoğu zaman geri dönülemez niteliktedir.



Şekil 2.2. Mekansal işlemler

1- Eşitlik, 2- Ayrılık, 3- Birleşme, 4- Kesişme, 5- Bağlantı

Birleştirme veya genelleme işlemleri neticesinde, sahip olunan veri detayı yitirilir ve veri kümesine analiz esnasında istenmeyen gürültü ve desenler eklenir [25].

Mekansal işlemler ile kaybolan detayın daha sonradan kazanılması interpolasyon gibi daha ileri mekansal işlemler gerçekleştirilmesini gerektirir. Ancak birleştirme işlemi ile kaybedilen detaydaki veri niteliğine hiçbir zaman erişilemez.

2.4. Tanımlayıcı Nokta Desen Hesaplamaları

Bir olayın mekansal olarak anlaşılmasında kullanılabilecek en temel mekansal hesaplama tekniği, gözlemlenen olgunun mekanda dağılımının tanımlayıcı nokta desen hesaplamaları ile ele alınmasıdır.

Tanımlayıcı nokta desen hesaplamaları, nokta ile tanımlanan veri kümelerinde noktalar arası uzaklığın dağılımının incelenmesi ile gerçekleştirilen işlemlerdir.

Noktalar arası uzaklığın; minimum, maksimum, ortalama, standart sapma, çarpıklık ve basıklık değerleri ile hesaplanır. Ulaşılan değerler, diğer veri kümeleri veya dağılımlar ile kıyaslanır.

Noktalar arası uzaklık; iki koordinat arasında Dünya'nın geometrik yapısını dikkate alınarak en kısa güzergahın uzunluğunun belirlenmesi ile bulunur. Bu hesaplamaya eğer varsa bağlantıları da tanımlayan bir ağ dahil edilebilir.

Bu çalışmada yol ağı noktalar arası uzaklık hesaplamalarına dahil edilmiştir.

2.4.1. Noktalararası uzaklığın standart sapması

Noktalar arası uzaklığın, noktalar arası ortalama uzaklıktan farkının kareler toplamının gözlem sayısına bölümünün karekökü noktalararası uzaklığın standart sapmasını verir. Bu değer noktalar arası uzaklığın ne kadar farklılık gösterdiğini belirler.

Standart sapmanın karesi (SE^2) ise varyansı verir. Standart sapma Eş. 2.1 ile hesaplanır:

$$SE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x})^2} \quad (2.1)$$

Eş. 2.1'de; x_{ij} , i ve j noktalar arasındaki uzaklığı, $\bar{x}$ noktalar arasında uzaklık ortalamasını, n gözlem sayısını ifade eder.

Çalışma süresinde noktalararası uzaklığı temsil eden veri kümesinin tamamı ele alınmış olduğu için standart sapma hesaplamasına Bessel düzeltmesi uygulanmamıştır. Veri kümesinden alınan bir örneklemin konu edinilmesi durumunda formül Eş. 2.2'de olduğu gibi değişiklik gösterecektir:

$$SE = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x})^2} \quad (2.2)$$

Eş. 2.2’de; x_{ij} , i ve j noktaları arasındaki uzaklığı, $\bar{x}$ noktalar arasında uzaklık ortalamasını, n gözlem sayısını ifade eder.

Normal dağılım gösteren bir veri kümesinde; ortalama değer + standart sapma ve ortalama değer – standart sapma aralıklarının içinde değerlerin %68,27’sinin kaldığı kabul edilir.

2.4.2. Ortalama merkez

Ortalama merkez, nokta dağılımlarının özellikle zamana göre dağılımlarındaki değişikliği görebilmek üzere önem taşır.

Ortalama merkez, X ve Y koordinat değerleri için ayrı ayrı hesaplanır.

$$\bar{x} = \frac{\sum_i X_i}{n} \quad (2.3)$$

$$\bar{y} = \frac{\sum_i Y_i}{n} \quad (2.4)$$

Ortalama merkez noktaların sahip olduğu öznitelikler ile ağırlandırılarak hesaplanması durumunda ağırlıklandırılmış ortalama merkez için $\bar{x}$ ve $\bar{y}$ koordinatı şu şekilde hesaplanır:

$$\bar{x} = \frac{\sum_i w_i X_i}{\sum_i w_i} \quad (2.5)$$

$$\bar{y} = \frac{\sum_i w_i Y_i}{\sum_i w_i} \quad (2.6)$$

Eş. 2.5 ve Eş. 2.6'da; $\bar{x}$ yatay ortalama merkezini, $\bar{y}$ dikey ortalama merkezini, w_i ağırlıklandırmada ele alınacak değişkenin i . ölçüm değerini, x_i i .nci noktanın dikey koordinatını, y_i i .nci noktanın yatay koordinatını ifade eder.

x yatay düzlemi için standart sapma ise Eş. 2.7'de belirtildiği gibi hesaplanır:

$$S_x = \sqrt{\sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n - 1}} \quad (2.7)$$

y dikey düzlemi için ise standart sapma ise şu şekilde hesaplanır:

$$S_y = \sqrt{\sum_{i=1}^n \frac{(y_i - \bar{y})^2}{n - 1}} \quad (2.8)$$

Eş. 2.7 ve Eş. 2.8'de $\bar{x}$ yatay düzlemdeki ortalama merkezini, $\bar{y}$ dikey düzlemdeki ortalama merkezini, x_i i .ci noktanın dikey koordinatını, y_i i .nci noktanın yatay koordinatını, n nokta sayısını toplamını ifade eder.

Ağırlıklandırılmış nokta dağılımlarında ağırlık için belirlenen değerde standart sapma hesaplamalarında göz önünde bulundurulur.

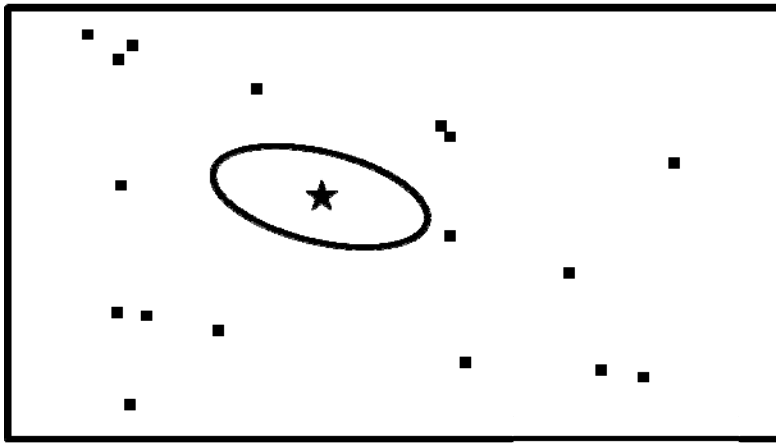
$$\sigma^{lw} = \sqrt{\frac{1}{\sum_i w_i} \sum_{i=1}^n w_i (x_i - \bar{x})^2 + \frac{1}{\sum_i w_i} \sum_{i=1}^n w_i (y_i - \bar{y})^2} \quad (2.9)$$

Eş. 2.9'da; σ^{lw} ağırlıklandırılmış mekansal standart sapmayı, $\bar{x}$ yatay düzlemdeki ortalama merkezini, $\bar{y}$ dikey düzlemdeki ortalama merkezini, w_i ağırlıklandırmada

ele alınacak değişkenin i .nci ölçüm değerini, x_i i .nci noktanın yatay koordinatını, y_i i .nci noktanın dikey koordinatını ifade eder.

2.4.3. Mekansal standart sapma elipsi

Nokta dağılımlarının herhangi bir yönde diğer yönlerden farklı olarak nasıl dağıldığını iki boyut üzerinde gösterir.



Şekil 2.3. Mekansal standart sapma elipsinin gösterimi

Mekansal ortalama değerlerinden standart sapmanın çıkarılması ile düzlemde bir elips üretilir.

Mekansal standart sapma elipsi özellikle ilişkilerin gözlemlenmesinde kullanılabilen başarılı bir gösterim şeklidir. Bu tanımlayıcı hesaplama trafik kaza noktalarının mekansal standart sapma eğrisi üzerinde gösterimi ile kazaların oluşumu ile ilişkili fiziksel bir yapı arasındaki bağı gözlemlenmesinde kullanılabileceği gibi, farklı veri setlerinin yönelik hesaplanarak Elipsin çapı ve ölçüleri üzerinden mekansal dağılımların kıyaslanmasına gidilebilir.

Mekansal standart sapma elipsi aynı zamanda yayılma fonksiyonuna sahip salgın hastalıklar gibi olayların yaygınlığı konusunda da kullanılabilir.

2.4.4. Çarpıklık

Nokta dağılımının x ve y düzlemlerinde göstermiş olduğu değişikliğin simetrik olmayışının ölçümüdür. Momentin 3. kuvveti olarak ifade edilir.

Eş. 2.10, n sayıda değere sahip veri kümesinin x düzlemindeki çarpıklığını verir:

$$g_1 = \frac{m_3}{m_2^{3/2}} = \frac{\sqrt{n} \sum_{i=1}^n (x_i - \bar{x})^3}{(\sum_{i=1}^n (x_i - \bar{x})^2)^{3/2}} \quad (2.10)$$

Eş. 2.10'da; x_i i.nci noktanın yatay koordinatını, $\bar{x}$ yatay düzlemdeki ortalama merkezini, m_3 yatay düzlemdeki üçüncü merkezsiz momenti ve m_2 yatay düzlem için örneklem varyantını verir.

Pozitif çarpıklıkta dağılımı gösteren grafikte grafiğin sağ tarafında yer alan kuyruk daha uzundur. Bu durum değerlerin grafiğin sol tarafına doğru yığıldığını gösterir. Negatif çarpıklıkta ise yığılım sağ tarafta olup, solda yer alana kuyruk sağa göre daha uzundur.

2.4.5. Basıklık

Standardize edilmiş dördüncü momenttir. Dağılımın grafik üzerinde gösteriminin basıklığı veya sivriligidir.

$$g_2 = \frac{m_4}{m_2^2} - 3 = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2)^2} - 3 \quad (2.11)$$

Eş. 2.11'de; x_i i.nci noktanın yatay koordinatını, $\bar{x}$ yatay düzlemdeki ortalama merkezini, m_4 yatay düzlemdeki dördüncü momenti, m_2 yatay düzlem için örneklem varyansını ifade eder.

2.5. Mekansal Analiz Yöntemleri

Mekan veriye sahip olduğu müddetçe; örneklemin mekansal niteliklerini veya diğer mekansal verileri dikkate alarak mekansal analiz gerçekleştirmek mümkündür.

Openshaw veriler üzerinde mekansal analiz yapma nedenlerini şu şekilde ifade etmiştir[25]:

- Mekansal ilişki ve desenleri açıklamak üzere geliştirilmiş hipotezin doğruluğunu test etmek,
- Mekansal desen veya ilişkiyi ortaya koymak,
- Karar verme sürecinde ve mekansal planlamalarda doğru sonuçlara erişmek.

Veri kümesinin temsil edildiği mekansal nesne tipine göre uygulanabilecek mekansal analiz türleri değişiklik göstermektedir.

Mekansal analiz türlerinden nokta deseninin araştırılmasına yönelik analizler, alan verisinin araştırılmasına yönelik analizlerden daha az yoğunlukta veriye ihtiyaç duyar. Bu durum analizin karmaşıklığını azaltırken, veri toplanmasının maliyetini de düşürür.

Çizelge 2.1. Mekansal nesnelere uygulanan analiz türleri

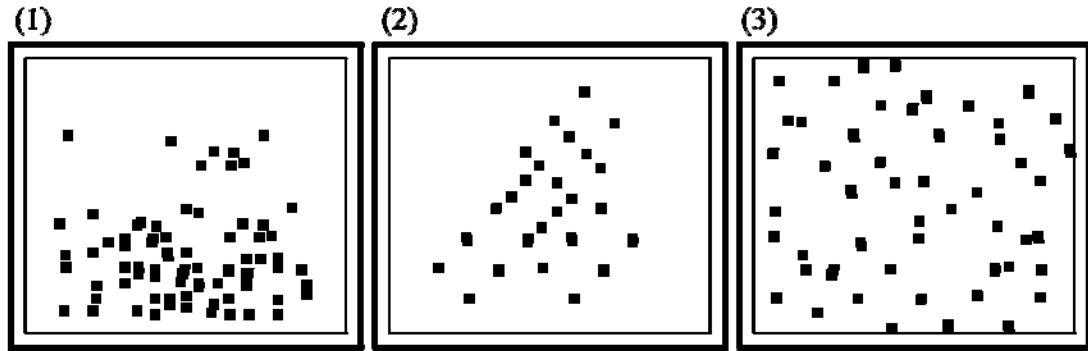
Mekansal Nesneler	Veri Esaslı Mekansal Analiz Türleri	Model Esaslı Mekansal Analiz Türleri
Nokta deseni (x,y)	Kare analizleri, Çekirdek tahmini, En yakın komşu uzaklığı, K fonksiyonu analizi,	Homojenlik ve heterojenlik, Poisson dağılımı modeli, Çok değişken dağılımı modeli,
Alan verisi (x,y,z)	Global mekansal ilişki ölçütleri (Moran I ve Geary C)	Mekansal regresyon modelleri
	Bölgesel mekansal ilişki ölçütleri (Moran's dağılım haritası)	Mekansal oto korelasyon artıkları için regresyon modelleri

Her ne kadar nokta verisini herhangi bir alana atayarak analiz çalışmalarını sürdürmek söz konusu ise de, eşleme veya atama işlemleri ele alınacak verinin detayının yitirmesine ve veri kümesindeki gürültünün artmasına sebebiyet verecektir.

2.6. Veri Esaslı Nokta Desen Analizleri

Veri esaslı nokta desen analizlerinde nokta dağılım sürecinin (P) tamamen rastlantısal olduğu kabul edilir. Dağılımın rastlantısal oluşu aynı zamanda diğer mekansal tanımlarında (yoğunluk, komşuluk gibi) rastlantısallığını gerektirir.

Veri esaslı nokta desen analizleri, bir olaya ait nokta veri tipindeki konum verisinin dağılımının açıklanmasına yönelik bir hipotez kurulması ve test edilmesi ile gerçekleştirilir. Analiz esnasında noktaların konumları incelenerek, nokta dağılımına sistematik olarak etki eden bir unsurun var olup olmadığı araştırılır. Etkinin varlığına göre nokta dağılımının; kümelenme, düzen veya rastgelelik oluşturup oluşturmadığına bakılır. Ulaşılan sonuca göre hipotez kabul veya reddedilir.



Şekil 2.4. Nokta dağılım desenleri

1- Kümelenme (heterojen dağılım), 2- Düzen (anizotropik dağılım),
3-Rastgelelik (homojen izotropik – poisson dağılımı)

Nokta desen analizlerinde noktaların kendi arasında etkileşiminin yanında başka mekansal sınıflarında nokta dağılımı üzerindeki etkisi araştırılabilir (Çapraz sorgulama). Semt pazarlarının trafik kazaları üzerinde etkili olup olmadığını araştırmak için; kazalar ile pazar yerleri arasındaki mesafe iki değişkenli nokta desen analizlerinde ele alınabilir. Değişkenlerin sayısının ikiden fazla olduğu durumlarda çok değişkenli nokta desen analizleri gerçekleştirilir.

Veri esaslı nokta desen analizlerinde ele alınan nokta dağılımı tektir. Bu sınırlama veri esaslı nokta desen analizlerinde nokta dağılımının homojen olduğunun ön kabul ile benimsenir. Heterojen ve anizotropik dağılımların ön kabul gördüğü çalışmaların gerçekleştirilmesi zordur.

Veri esaslı nokta desen analizlerinde; kare analizleri, çekirdek yoğunluk tahmini, en yakın komşu indeksi ve k fonksiyonu kullanılabilir.

Kare analizi ve çekirdek yoğunluk tahmini birincil düzeninin etkisini, en yakın komşu indeksi ve K fonksiyonu ise ikincil düzenin etkisini kontrol eder.

Veri esaslı nokta desen analizlerinde yoğunluk (λ) birincil etki olarak adlandırılır. Homojen dağılım gösteren nokta desenlerinde λ sabittir ve çalışma alanı noktaların sayısına bölünerek hesaplanır.

$$\lambda(s) = \lim_{ds \rightarrow 0} \left\{ \frac{N(ds)}{ds} \right\} \quad (2.12)$$

Eş. 2.12’de; ds noktaların bulunduğu alanı, N(ds) nokta sayısını, ds ise çalışma alanını ifade eder.

İkincil etki ise nokta sürecinde ele alınan iki farklı noktaya ait değerlerin birlikte değişimleri (kovaryans) ya da etkileşimleridir (korelasyon). İkinci etki yoğunluğu ise şu şekilde gösterilir:

$$\lambda(s_i, s_j) = \lim_{ds_i, ds_j \rightarrow 0} \left\{ \frac{N(ds_i), N(ds_j)}{ds} \right\} \quad (2.13)$$

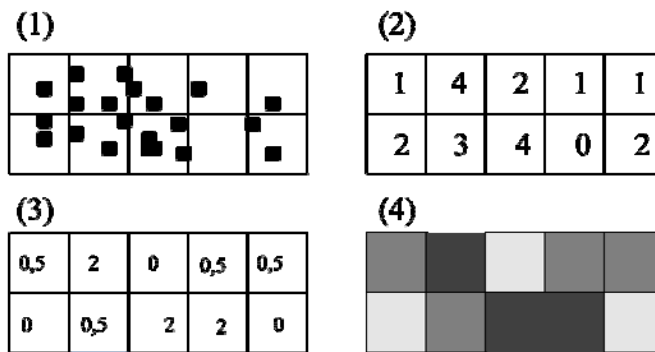
Eş. 2.13’de; ds_i ve ds_j ele alınan iki farklı noktanın bulunduğu alanları, $N(ds_i)$ ve $N(ds_j)$ bu alanlarda yer alan nokta sayısını ifade eder.

2.6.1. Kare analizleri

Nokta desen analizlerinde yoğunluğun kıyaslanması kare analizleri ile yapılabilir. Kare analizlerinde gözlemlenen yoğunluk ile rastgele dağılım halinde öngörülen yoğunluk kıyaslanır. Analiz esnasında çalışma alanı karelere bölünür ve her bir kare içinde kalan nokta sayısı tespit edilir. Karelerin içinde kalan noktaların sayımı ile elde edilen örneklemin dağılımı kontrol edilerek, nokta deseninin mekansal rastlantısallığa sahip olup olmadığı belirlenir.

Kare sayım örnekleminin varyansının ortalamaya bölümü ile elde edilen değer 1 olması durumunda dağılımın mekansal rastlantısallığa sahip olduğu tespit edilmiş olur. 1'den yukarı değerlerde kümelenme, 1'den aşağıda olan değerlerde ise düzen mevcuttur. $\frac{S^2}{\lambda}=1$ İse nokta desen rastlantısal, $\frac{S^2}{\lambda} > 1$ nokta desen kümelenmiş, $\frac{S^2}{\lambda} < 1$ ise nokta desen düzenlidir. S^2 varyansı, λ yoğunluğu ifade etmektedir. Kare analizleri ile nokta desenin rastlantısallığına yönelik sıfır hipotezi (H_0) χ^2 ile Eş. 2.14'te belirtildiği şekilde test edilir:

$$\chi^2 = (n - 1) \frac{S^2}{\lambda} \quad (2.14)$$



Şekil 2.5. Kare analizi

- 1- Nokta deseni, 2- Nokta sayımları, 3- Yoğunluk gösterimi,
4- $\frac{S^2}{\lambda}$ gösterimi

Kare analizlerinde kullanılan karenin büyüklüğü kare analizi ile erişilen sonuçlara etki eder. Analizin detayını arttırmak üzere karenin alanının daraltılması analizler esnasında birçok karenin boş olmasına (0 sonuç üretmesine) sebebiyet verir.

Basit olmasına rağmen kare analizlerinin dezavantajları da mevcuttur: Kare analizleri esnasında nokta konum verisinin bir alana atanmış olması detayların yitirilmesine ve veriye gürültü eklenmesine sebebiyet verir. Daha önce de bahsedildiği üzere bu işlem geri dönüşü olmayan bir süreçtir.

2.6.2. Çekirdek yoğunluk tahmini

Çekirdek yoğunluk tahmini; mekansal sınıfın dağılım üzerindeki etkisini araştırmak veya gözlemler arasındaki benzerlikleri tespit etmek üzere kullanılır. Çekirdek yoğunluk tahmininde örneklemin herhangi bir ögesi için nokta sürecine bağlı olarak bir noktanın oluşabilme ihtimali değerlendirilir. Çekirdek tahmini veri kümesinde yer alan her bir noktanın komşuluğunun incelenmesine dayanır.

Çekirdek analizlerinde kare analizlerinden farklı olarak belirli bir noktada örnekleme konu olan olayın gerçekleşme şiddeti¹ ele alınır [27].

Oysa kare analizleri gözlemlenen nokta adedinin örnekleme konu alana oranlanmasına dayanır. Kare analizlerinde oran alan üzerinden yapılırken çekirdek tahminlerinde ise oranlama bant genişliği olarak adlandırılan mesafeye göre yapılmaktadır.

Her bir nokta için; belirlenmiş olan bant genişliği (daire yarıçapı) içinde kalan nokta sayısı sayılarak öngörülen nokta oluşum şiddeti daha önceden tespit edilmiş ağırlıklandırma fonksiyonuna yerleştirilerek hesaplanır.

¹ Literatür taramasında birçok çalışmada yoğunluk (density) ile oluşma şiddeti (intensity) arasında bir fark olmadan bu iki kavram birbirleri yerine kullanılmıştır. Ancak kare analizi ile çekirdek tahmin metodunun kıyaslanarak birlikte ele alınması söz konusu olunca bu kavramların farklılıklarını muhakkak ortaya koymak yerinde olacaktır. Bu çalışmada nokta yoğunluğu ile nokta oluşum şiddeti farklı kavramlar olarak ele alınmıştır.

Örneklemden herhangi bir noktanın nokta oluşum şiddeti Eş. 2.15'te belirtildiği şekilde bulunabilir [27]:

$$\tilde{f}(s) = \frac{1}{Nb} \sum_{i=1}^N \frac{1}{b^2} \text{kern} \left(\frac{(s - s_i)}{b} \right) \quad (2.15)$$

Eş. 2.15'te; $\tilde{f}(s)$ noktasının b bant genişliğinde çekirdek yoğunluğunu, s dairenin merkezinde yer alan nokta oluşum şiddetini, s_i yarıçap içindeki her bir örnekleme, $\text{kern}(\)$ ise, örneklem içinde yer alan noktaların yakından uzağa doğru azalan bir şekilde ağırlıklandırılmasını sağlayan fonksiyondur. Çekirdek tahmininin yapılması esnasında; merkezde yer alan s_1 noktası ile dairenin içinde kalan her bir gözlem s_i arasındaki mesafenin ağırlıklandırılması için kullanılacak $\text{kern}(\)$ fonksiyonunun belirlenmesi gerekmektedir.

Waller, matematik ve hesaplama nitelikleri farklılık göstermekle birlikte pozitif değer üreten herhangi bir fonksiyonun $\text{kern}(\)$ için kullanılabileceğini ve bu tercihin sonuç üzerindeki etkisinin çok küçük olduğunu belirtilmektedir [27]. Smith ve arkadaşları tarafından sıklıkla kullanıldığı belirlenen çekirdek fonksiyonları şunlardır [28]:

$$\frac{1}{\sqrt{2\pi}b} \exp \left[-\frac{1}{2}(u)^2 \right] \quad (2.16)$$

$$\frac{15}{16b} [1 - (u)^2]^2 I(|u| \leq 1) \quad (2.17)$$

$$\frac{1}{b(2 - b)} (1 - |u|) I(|u| \leq 1) \quad (2.18)$$

$$\frac{1}{2b} (|u| \leq 1) \quad (2.19)$$

Eş. 2.16 ile normal dağılımı ifade eden çekirdek fonksiyonuna, Eş. 2.17 ile küresel dağılımı ifade eden çekirdek fonksiyonuna, Eş. 2.18 ile konik dağılımı ifade eden çekirdek fonksiyonuna, Eş. 2.19 ile birörnek dağılımı ifade eden çekirdek fonksiyonuna erişilir.

Kullanılacak olan fonksiyonun belirlenmesinden sonra her nokta için (dairenin merkezinde yer alan s_1 noktası ile) dairenin içinde kalan her bir gözlem s_i arasındaki mesafe ölçülür. Merkeze daha yakın değerlerin ağırlığı uzak olanlardan daha fazladır. Değerler küçükten büyüğe yakından uzağa doğru ağırlıklandırılır.

Çekirdek tahminin gerçekleştirilmesinde sonraki işlem, bant genişliğinin belirlenmesidir. Seçilen bant genişliğinin, ortalama nokta oluşum şiddeti ile gözlemlenen nokta oluşum şiddeti arasındaki standart sapmayı en aza indirmesi beklenir. Hesaplamalar esnasında bant genişliğinin küçük tutulması aralıklı bir haritanın elde edilmesi ile sonuçlanır. Bant genişliği yüksek tutulur ise noktalar arası geçiş daha yumuşak olur.

Bant genişliğinin dinamik olarak belirlenip, dikkate alınması, noktalar arası mesafenin az olduğu yerlerde detayı yitirmeden analizlerin yapılmasını sağlar. Bant genişliğinin belirlenmesinde Diggle tarafından belirlenen ve Brunsdon tarafından uygulanan Uyarlamalı Çekirdek Tahmini metodu kullanılabilir [26].

2.6.3. En yakın komşu indeksi

Örneklemin mekansal dağılımının; kümelenme, rastgelelik veya düzenlilik gösterip göstermediği ikincil etkinin varlığı kontrol edilerek belirlenir. Birincil etki örneklemin konu edindiği tüm alana dair çıkarımlar yapmayı sağlarken ikincil etki sonuçları sınırlı bir alan veya komşu noktalar için geçerlidir.

En yakın komşu indeksi örnekleme yer alan noktaların en yakın komşuları ile aralarındaki mesafenin rastlantısal olup olmadığını kontrol edilmesi ile gerçekleştirilir.

Her bir nokta için en yakın diğer nokta bulunur ve aralarındaki mesafe tüm noktalar için toplanarak toplam nokta sayısına bölünür. Elde edilen değer rastlantısal değere olan oranı en yakın nokta indeksini verir.

$$R = \frac{r_o}{r_e} \quad (2.20)$$

Eş. 2.20’de, R en yakın komşu nokta indeksini, r_o gözlemlenen en yakın komşu indeksi ortalamasını, r_e beklenen en yakın komşuluk indeksi ortalamasını ifade eder.

Beklenen en yakın komşuluk değeri ise Eş. 2.21 ile hesaplanır:

$$r_e = 0,5 \sqrt{\frac{A}{N}} \quad (2.21)$$

Formülde; A gözlemlenen alanı, N örneklem sayısını ifade eder.

Donnelly’in rastlantısallığın gözlem alanının sınırlarında değişimine yönelik düzenlemesi ile Eş. 2.21 şu şekilde değişir [29].

$$r_e = 0,5 \sqrt{\frac{A}{N}} + \left(0,0514 + \frac{0,041}{\sqrt{N}} \right) * \frac{B}{N} \quad (2.22)$$

Eş. 2.22’de; A gözlemlenen alanı, N örneklem sayısını, B ise çalışma alanının çevresini ifade eder. R=0 ise kümelenmiş dağılım, R=1 ise rastlantısal dağılım, R=2,15 ise düzenli dağılım olduğu kabul edilir.

Clark ve Evans; gözlemlenen en yakın komşu uzaklığı ortalamasının, rastgele ortalama uzaklıktan farkının Z testi ile kontrol edilebileceğini ortaya koymuştur [30]:

$$Z = \frac{r_o - r_e}{SE_{d(e)}} \quad (2.23)$$

Eş. 2.23'te kullanılan noktalar arası ortalama mesafeye ait standart sapma ise Eş. 2.24 ile şu şekilde hesaplanır [30]:

$$SE_{d(e)} = \frac{0,26136}{\sqrt{\frac{N^2}{A}}} \quad (2.24)$$

2.6.4. K fonksiyonu

İkincil etkinin araştırılmasında en çok kabul gören ve sıklıkla başvuru alan metot Ripley tarafından 1981 yılında yayınlanmıştır [31].

Ripley'in çalışmasında Clark ve Evans'ın çalışmasından farklı olarak noktalar arası mesafe ölçümlerinin değişik aralıklarla ele alınması, noktalar için ikincil etkinin tanımlanabilmesine imkan sağlamıştır.

Aynı zamanda noktaların dağılımlarına yönelik korelasyonun incelenebilmesi ve mekansal yapı hakkında nicel değerlere ulaşılabilmesi K fonksiyonunun popülerliğini arttırmıştır.

Yamada'nın yapmış olduğu çalışmalar K fonksiyonunun; hastalıkların yayılımında, trafik kazası sonucu meydana gelen ölümlerin modellenmesinde ve bitki örtüsünün desen analizlerinde kullanılabileceğini ortaya koymuştur [10].

Ripley'in K olarak adlandırdığı ikincil moment ölçüsü şu şekilde ifade edilebilir[31]:

$$K(r) = \frac{N(r)}{\lambda} \quad (2.25)$$

Eş. 2.25'te; $K(r)$ K fonksiyonunu, r fonksiyonda ele alınacak mesafeyi, λ r mesafesi içinde kalan alandaki yoğunluğu ifade etmektedir.

K fonksiyonu rastlantısal dağılıma sahip bir örnekleme uygulandığında; r mesafesinde yer alacak nokta sayısını yoğunluk ile alanın çarpımına eşittir.

$$N(r) = \lambda \pi r^2 \quad (2.26)$$

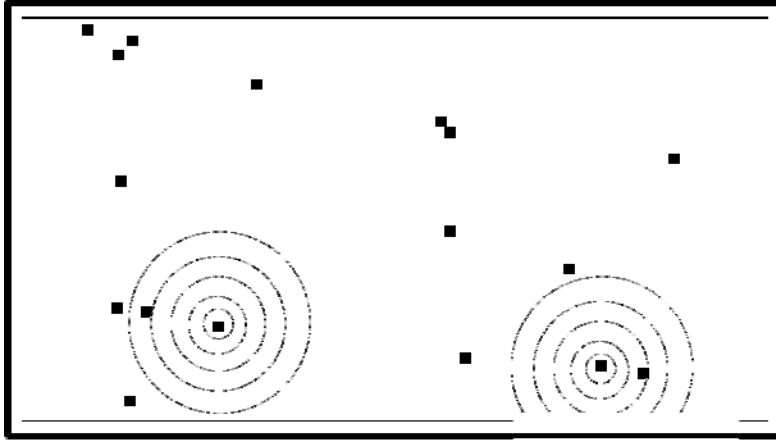
Eş. 2.26'de, $N(r)$ r yarıçapında bulunan nokta sayısını; λ , r mesafesi içinde kalan alandaki nokta yoğunluğunu, r fonksiyonda ele alınacak mesafeyi ifade etmektedir.

Kümelenme gösteren bir dağılımda ise; gözlemlenen örneklem yoğunluğu, öngörülen yoğunluktan fazla olacağı için $K(r) > \pi r^2$ olacaktır. Düzenli bir dağılıma sahip örnekleme ise $K(r) < \pi r^2$ dir [33].

K değerinin hesaplanması, örnekleme yer alan her bir noktanın etrafında belirlenen sıklıkta çizilen dairelerde yer alan noktaların sayılması ile bulunur.

$$K(d) = \frac{1}{\lambda^2 A} \sum_{i \neq j} I_d(d_{ij}) \quad (2.27)$$

Eş. 2.27'de, A incelenen alan, $I_d(d_{ij})$ ise d_{ij} nin d den küçük olması durumunda 1 aksi halde 0 değeri almasını sağlayan çarpanı ifade etmektedir [33].



Şekil 2.6. K fonksiyonunun uygulanişı

Şekil 2-3'te görüleceğı üzere inceleme alanının sınır ve köşelerinde bulunan noktalar için yapılan hesaplamalarda örneklem dışındaki noktalar dikkate alınmadığı için hatalar oluşacaktır.

Nokta sayımında kullanılan dairenin alan ölçüsünün r yarıçaplı ait dairenin alanına oranlanması, örneklemin köşelerdeki dağılımında karşılaşılan hatanın kontrol altında tutulmasını sağlar.

$$\hat{K}(r) = \frac{1}{\lambda^2 A} \sum_{i \neq j} \frac{I_r(d_{ij})}{w_{ij}} \quad (2.28)$$

Dixon'un eş. 2.28 ile; $\hat{K}(r)$ i ve j noktaları için r mesafesindeki K değerini, λ nokta yoğunluğunu, A çalışma alanı ölçüsünü, $I_d(d_{ij})$ ise d_{ij} nin d den küçük olması durumunda 1 aksi halde 0 değeri almasını sağlayan çarpanı, w_{ij} ise örneklemin köşeye gelen değerleri için düzeltme çarpanını ifade eder [35].

Yoğunluğu bilinmeyen durumlarda formül; $\lambda = n/A$ olması nedeni ile şu şekilde düzenlenir [35]:

$$\hat{K}(r) = \frac{A}{n^2} \sum_{i \neq j} \frac{I_r(d_{ij})}{w_{ij}} \quad (2.29)$$

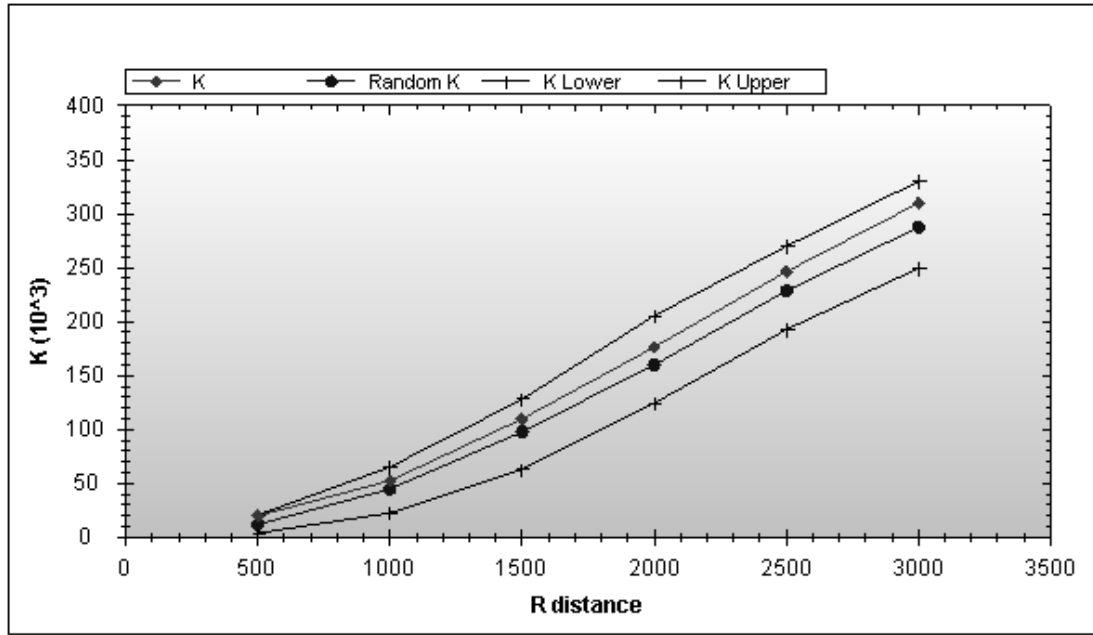
Eş. 2.29’da, $\hat{K}(r)$ i ve j noktaları için r mesafesindeki K değerini, n gözlemlenen örneklem sayısını, A çalışma alanı ölçüsünü, $I_d(d_{ij})$ ise d_{ij} nin d den küçük olması durumunda 1 aksi halde 0 değeri almasını sağlayan çarpanı, w_{ij} ise örneklemin köşeye gelen değerleri için düzeltme çarpanını ifade eder.

$\hat{K}(r)$ değeri örneklemdaki her bir noktada K fonksiyonun hesaplanması ile bulunur. İncelenmekte olan noktanın etrafında belirli aralıklarla daireler çizilir. Her aralık içinde kalan örneklem sayısı sayılır. Tüm noktalarda bu işlem tekrar edilir. Aynı mesafe aralığındaki toplamlar birleştirilir. İncelenen alanın toplamlara oranı $\hat{K}(r)$ değerini verir.

Her bir mesafe için hesaplanan $\hat{K}(r)$ değeri, örneklemin dağılımını diğer dağılımlarla karşılaştırmak üzere grafik üzerine işlenebilir veya matematiksel olarak değerlendirilir.

En yakın komşu analizinde olduğu gibi örnekleme ait sonuçlar, rastlantısal dağılıma ait sonuçlar ile kıyaslanarak değerlendirilir.

Grafik üzerinde de benzeri bir kıyaslama yapılır. Örneklem sayısı kadar rastgele nokta üretilir ve bu noktaların her birinde $\hat{K}(r)$ değeri hesaplanır. Grafik üzerinde $\hat{K}(r)$ değerinin güven aralığına göre maksimum ve minimum aralıkları işaretlenir. Test örnekleminin grafiğe yansıtılmasını takiben; grafik üzerinde incelemekte olduğumuz örneklem işaretlenir. Maksimum değerlerin üzerindeki aralıklar için kümelenme, minimum değerlerinin altındaki aralıklar için ise düzenli dağılımın olduğu söylenebilir.



Şekil 2.7. K fonksiyonu ve rastlantısal sonuçlar ile kıyaslanması

Besag gösterimin düzenlenmesi ve sonuçların 1 ile -1 arasında gösterilmesi için $\hat{L}$ değerinin kullanılmasını tavsiye etmiştir [32].

Bu durumda $\hat{L}$ değeri şu şekilde hesaplanır [32]:

$$\hat{L}(r) = \sqrt{\frac{K(r)}{\pi}} - r \quad (2.30)$$

Eş. 2.30'da, $K(r)$, r mesafesindeki K değerini, r sayıma esas mesafeyi ifade eder.

Goreaud ve Pélissier ise, Poisson dağılımının incelenen tüm aralıklarda $\hat{L}(d)$ 'ın 0 değerini üretmesi nedeni ile $\hat{K}(d)$ 'dan daha kolay yorumlanabilir olduğunu belirtmektedir [33].

$\hat{L}(r)$ değerinin pozitif olması örneklemede kümelenmeyi, negatif olması ise düzenli bir noktasal dağılım olduğunu gösterir.

İki ve çok değişkenli veri kümelerinde de nokta desen analizlerinin gerçekleştirilmesinde de $\widehat{K}_{(d)}$ fonksiyonu kullanılabilir. Tek değişkene (konum bilgisine) sahip örneklemin incelenmesinde olduğu gibi iki ve çok değişkenli veri kümelerinde nokta dağılımlarının; rastlantısallık, düzen veya kümelenme gösterip göstermediği kontrol edilir.

Bu tanım şu formül ile ifade edilir:

$$K_{ij}(r) = \frac{N_{ij}(r)}{\lambda_j} \quad (2.31)$$

Eş. 2.31’de; $K_{ij}(r)$ i noktasının j örneklemine için K değerini, $N_{ij}(r)$ i noktasının r mesafesinde yer alan j örneklemindeki nokta sayısını, λ_j j noktası yoğunluğunu ifade eder.

Dixon $\widehat{K}_{ij}(r)$ değerinin hesaplanması için ise Lotwick tarafından 1982 yılında yayınlanan [34] şu formülün etkin bir değer vereceğini belirtmektedir[35]:

$$\widehat{K}_{ij}(r) = \frac{1}{\lambda_i \lambda_j R} \sum_k \sum_l \frac{I(d_{i_k, j_l})}{w(i_k, j_l)} \quad (2.32)$$

Eş. 2.32’de; $\widehat{K}_{ij}(r)$, i ve j noktaları için r mesafesindeki K değerini, λ_i i nokta yoğunluğunu, λ_j j nokta yoğunluğunu, R çalışma alanı ölçüsünü, $I(d_{i_k, j_l})$ k sıra numaralı i noktası ile l sıra numaralı j noktaları arasında kalan mesafenin d den küçük olması durumunda 1 aksi halde 0 değeri almasını sağlayan çarpanı, $w(i_k, j_l)$ ise örneklemin köşeye gelen değerleri için düzeltme çarpanını ifade eder.

Goreaud ve Pélissier’a göre düzeltme çarpanı, olayların öngörülme yoğunluğunun örneklemdaki yarıçap ile oranlanması ile elde edilir [33].

2.7. Nokta Desenlerinin Modellenmesi

Mekansal veriler esas alınarak mekansal dağılımlarının modellenmesi, olayların oluşumuna etki eden faktörleri belirlemede kullanılan bir yöntemdir.

Olayların mekansal dağılımına yönelik erişilen sonuçlar bilinen nokta dağılımları (kümelenme, düzenlilik, rastgele) ile karşılaştırılır. Bu sayede nokta desenlerinin genel yapısını ve oluşumunu açıklayan hipotezler değerlendirilebilir.

Baddeley [36] nokta süreçlerinin modellenmesinin ile

- Mekansal eğilimin açıklanabileceğini,
- Örneklem değişkenleri arasındaki eş değişimin belirlenebileceğini,
- Örneklemdeki noktalar arası etkileşimin izlenebileceğini,
- Noktaların dağılımında diğer faktörlerinin etkisinin gözlemlenebileceğini belirtmiştir.

Nokta süreçlerinin modellenmesinde modeller arasındaki en keskin farklılık nokta dağılımın rastlantısallığı ile ilgilidir. Dağılımın rastlantısal olması; ele alınan değişkenin örneklemin oluşumunda mekansal herhangi bir etkisini olmadığı anlamına gelmektedir.

Rastlantısal dağılımda; örnekleme nokta ile temsil edilen öge uzayının herhangi bir yerinde de aynı ihtimal ile meydana gelebilir. Rastlantısal bir dağılıma sahip örnekleme; bir ögenin mekansal tanımı ile bir diğer ögenin mekansal tanımı arasında herhangi bir ilişkinin bulunmadığı kabul edilir.

Noktanın örneklem uzayının herhangi bir yerinde aynı ihtimal ile meydana geliyor olması birincil etki ile açıklanır iken, bir noktanın herhangi bir yerde oluşunun bir başka noktaya ait oluşumu üzerindeki etkisi ya da etkisizliği ise ikincil etki ile açıklanmaktadır.

Noktaların birbirlerinin oluşumu üzerindeki olumlu ya da olumsuz etkileri, noktalar arasında, rastlantısal dağılıma göre daha büyük boşlukların ya da daha küçük boşlukların oluşmasına neden olur. Noktalar arası boşlukların rastlantısal dağılımlara göre daha fazla olması durumunda kümelenmenin, boşlukların rastlantısal dağılıma göre daha az olması durumunda ise düzenliliğin varlığı söz konusudur.

Nokta desenlerinde dağılımın yanında ele alınan diğer bir kavram ise sabitliktir¹. Örneklemin yoğunluğunun örneklemin mekansal özelliklerinden bağımsız olması durumunda yoğunluğun değişmezliği söz konusudur. Örneklemin yoğunluğu, incelenen alanın herhangi bir parçasığında değişiklik göstermez.

Örneklemin yönden bağımsız olması nokta desenlerinde gözlemlenen bir diğer özelliktir. Yönden bağımsız dağılım gösteren örneklem izotropik (eş yönlü) olarak adlandırılırlar.

Poisson, nokta dağılımını izlemek üzere en sık kullanılan süreçtir. Poisson dağılımı ile tanımlanmış sınırlı bir alan içerisinde önceden tahmin edilemeyen ancak pozitif değere sahip nokta yoğunluğu (λ) belirlenebilir. Nokta sayısı bir diğer oluşumdan bağımsızdır ve tanımlanan alan içerisinde eşit nokta oluşma ihtimali mevcuttur. Bu tanıma göre Poisson süreci rastlantısal bir dağılım gösterir. Ancak tanımın birinci hükmü ile toplam nokta sayısı nokta yoğunluğuna bağlı olarak değişiklik gösterir.

Homojen poisson dağılımında toplam nokta sayısı ile nokta yoğunluğu arasındaki ilişki Eş. 2.33'te şu şekilde belirtildiği üzeredir:

$$\hat{\lambda} = N/|A| \quad (2.33)$$

Poisson dağılımının ikinci özelliği olan, nokta sayısının diğer etkenlerden bağımsız olması, bize tamamen rastlantısal dağılımı yansıtan bir nokta deseni sağlar. Sıfır

¹ Kavram İngilizce'de stationary olarak adlandırılmaktadır.

hipotezi rastlantısallığın test edilmesi ile kontrol edilebilir. Örneklemenin temsil ettiği nokta deseninin düzenliliği veya rastlantısallığı veya kümelenmesi ile kurulan hipotez kabul veya reddedilir.

Rastlantısal dağılımlara Monte Carlo metodu ile üretilen örneklemlemlerle ulaşılabilir. Hipotez testinin güvenilirliğini arttırmak üzere oluşturulan test örneklemleri sayısı yüksek tutulur. Artan bilgisayar işlemci gücü sayesinde örneklem adedi 1000 örnekleme erişebilmektedir. Ulaşılan P değeri histogram veya harita üzerinde gösterilerek incelenir.

Çevredeki örneklemeleri açıklamakta homojen nokta desenleri genellikle yetersiz kalırlar. Nokta desenlerinin nadiren homojen dağılım gösterdikleri görülür.

Greibe tarafından 2001 yılında gerçekleştirilen çalışmada, sadece trafik yoğunluğunu yansıtan yoldan geçen günlük ortalama araç sayısı değişkeninin kendi başına şehirlerarası trafik kazalarının %70'ni açıklayabildiği belirlenmiştir. Model kavşaklara uygulanınca kazaların %90'nı açıklanabilmektedir [37].

Aynı çalışmada sadece trafik hacmi verisi kullanılarak; çalışmanın başında oldukça önemli olduğu düşünülen yol kesimi uzunluğu, hız limiti, yolun bölünmüş olup olmadığı, şerit sayısı, yol genişliği ve kavşağa ait geometrik özellikleri içeren diğer değişkenler modelden çıkartılmıştır. Bu değişkenlerin kavşaktan kavşağa farklılık göstermekle birlikte modelin açıklayıcı gücünü %1-4 oranında arttırdığı tespit edilmiştir [37].

Trafik kazalarının dağılımını modellenmesinde tüm yol kesimlerinin trafik hacmi değeri aynı olmayacağı için, o yollarda meydana gelen kaza yoğunlukları da farklı olacaktır. Homojen poisson nokta dağılımında kabul gören nokta yoğunluğuna yönelik tanımın göz ardı edilmesi ile heterojen bir dağılıma ulaşılır. Böylece oluşturulan model kümelenme gösterir.

Heterojen poisson dağılımında, inceleme alanının her bir alt parçası Poisson dağılımına uyan rastgele nokta yoğunluğuna sahiptir. İnceleme alanı içindeki örneklem adedi, rastgele örneklem adedindeki noktanın toplamına eşittir.

2.8. Mekansal Veri Madenciliği

Mekansal verilerin çokluğu tanımlayıcı istatistiki analizlerinin gerçekleştirilmesini zorlaştırmaktadır. Haritalar üzerindeki milyonlarca objenin örneklem üzerindeki olası etkisinin araştırılması ve gizli desenlerin açığa çıkarılması ile örneklemin değerlendirilmesi sağlanır. Veri madenciliği neticesinde ulaşılan sonuçlar, istatistiki araçlar ile değerlendirilir.

Mekansal analiz yöntemi ile veri madenciliği arasındaki farklılık; verinin ele alınma şekli ve sonuçların niteliği ile ilgilidir. İstatistiki çalışmalar, hipotezler ile oluşturulan varsayımların doğruluğunu araştırılırken, veri madenciliği hipotez olarak değerlendirilecek niteliğe sahip verilerin açığa çıkarılmasını amaçlamaktadır.

Veri madenciliğinde gözlemlenen veriler üzerinde indirgeme, özetleme, sınıflandırma gibi işlemler yapılarak incelenen veriden anlamlı bilgilere ulaşılması sağlanır.

Shekhar [38] ve Ester'in [39] çalışmalarında mekansal veri madenciliğinin dört farklı uygulamasının olduğu görülür. Mekansal kümelenme ve ilişkilendirme örneklemin tanımlanmasına yönelik verilerin açığa çıkarılması için kullanılırken, sınıflandırma ve regresyon tahmin için kullanılır.

Mekansal kümelenme; taşıdıkları öznitelikler açısından benzerliğe sahip mekansal nesnelerin alt sınıflar halinde gruplanmasıdır. Örneklemin incelemeye alınan herhangi bir noktasında beklenenden daha az veya daha fazla yoğunlukta olayın gözlenmesi ile çıkarımlarda bulunabilir. Örnekleme ait kare analizi, en yakın komşu indeksi ve K değerlerinin üretilmesi ile mekansal dağılımın rastlantısallığı belirlenir.

İstatistiksel tespitin ardından kümelenme algoritmaları ile kümelenmelerin hangi niteliklerle ilişkili olduğu değerlendirilir.

Kümelenme uygulamaları kontrolsüz bir metottur. Uygulama dışarıdan herhangi bir müdahaleye ihtiyaç duymaz. Kümelenmenin ara sonuçları bir sonraki sonuç üzerinde etkiye sahip değildir.

Hiyerarşik kümelenme algoritmaları, istenilen kritere erişilene kadar desenleri birleştirir ve ayırır. Kısmi kümelenme algoritmalarında ise her bir desen ayrı kümedir. Örneklemdaki noktalar öngörülen durma koşuluna erişilene kadar bir kümeden diğerine atanır.

Yoğunluğun dikkate alındığı algoritmalarda ise, örneklemin yoğunlaştığı şartlar belirlenerek kümelenmenin ortaya çıkarılması sağlanır. Daha büyük alanlar ve yüksek kapasiteli verilerin analiz edilmesi için ise inceleme alan üzerinde ızgaralama yapılır. Yoğunluk değerlendirmesi ızgaralar içindeki hücreler esas alınarak gerçekleştirilir.

İlişkilendirme algoritmaları benzer öznelitliklere sahip noktaların hangilerinin birlikte meydana geldiğini ortaya koyar. İlişkilendirme; ilişki kurallarının destek ve güvenilirlik değerlerinin ifade edilmesi ile sağlanır. Destek değeri, kuralı ifade eden şartların toplam örneklem içerisindeki oluşum ihtimalini ifade eder. Güven değeri ise kuralı ifade eden şartların toplamının birincil şarta göreceli oranıdır.

Çizelge 2.2. İlişkilendirme için kullanılacak örnek kaza döküm tablosu

Kaza No	İhlal Edilen Trafik Kuralları
1	Aşırı hız, öncelik, geçiş
2	Aşırı hız, geçiş
3	Aşırı hız, park
4	Öncelik, kırmızı ışık, alkol

Çizelge 2.3. Örnek verilere dayanan ilişkilendirme çalışması sonuç tablosu

Kural	Destek	Güven
Aşırı Hız → Geçiş	%50	%66,67
Geçiş → Aşırı Hız	%50	%100

İlişki kurallarının açığa çıkarılması için kullanılan algoritmalar eşik destek ve güven oranları ile sınırlı sonuçlar üretirler. Böylece minimum destek ve güven oranlarını karşılayan ilişki kuralları tespit edilir. İlişki kurallarının açığa çıkarılmasında apriori algoritması en sık kullanılan algoritmadır.

Apriori algoritması veri kümesinin frekanslarını bularak bunları birbirleri ile ilişkilendirir. Daha sonra bu küme üzerinde destek ve güven sınırlaması uygulanır. Algoritma da yapısal sorgulama dili kullanılabilmesi nedeni ile de sıklıkla tercih edilir.

Sınıflandırma ise örnekleme yer alan her bir olayın mekansal nitelikleri incelenerek aralıklı değişkenler arasındaki ilişkiyi ortaya koyan tanımlamalar belirlenmeye çalışılır. Sınıflandırma için regresyon, karar ağacı algoritması, yapay sinir ağları veya Bayes uygulamaları kullanılabilir. Uygulama neticesinde ortaya çıkan model sınıflandırma kuralları, karar ağacı veya formüller ile sunulabilir. Nihai olarak modelin geçerliliği test edilir.

Sürekliliğe sahip değişken tipleri için kullanılan regresyon analizi, değerlendirilecek olan örnekleme açıklayacak modeli belirler. Doğrusal ya da doğrusal olmayan modeller ile açıklanan veri kümesi haricinde kalan ve gürültüden yalıtılmış verileri ilişkili öznitelikler ile açıklayan bir model oluşturulur. Bu sayede bilinmeyen durumlar için model işletilerek o duruma ait sonuçlara ulaşılır.

Yapay sinir ağları uygulamalarında, eğitim amacı ile ayrılan veriler yapay nöronları temsil eden düğümlerden geçirilerek ağırlıklandırılmış değerler elde edilir. Ağırlıklar ileri düğümlerin ürettikleri geri beslemeler ile birlikte örnekleme verileri yorumlamak üzere kullanılırlar.

Karar ağacı uygulamaları, örneklem içinde yer alan özniteliklere mekansal özellikler ile erişimi tanımlar. Ağaçta yer alan her bir düğüm mekansal niteliği, her bir dal ise öz niteliği temsil eder.

Karar ağacı algoritması en az dal kullanarak örnekleme değerlendirmeyi hedefler. Karar ağacı algoritmasının uygulanması neticesinde sunulan veri bir ağaç çizimi ile gösterilir ve her bir düğüm ile erişilen örneklem oranı ifade edilir. Verimlilik gereksiz saçaklanma gösteren örnekleme açıklamaktan uzak dalların, daha açıklayıcı dallara tercih edilmesi ile sağlanır.

Yapılan literatür taramasında Quinlan tarafından geliştirilen ID3 algoritmasını Ester mekansal sınıflandırma için başarı ile kullanmıştır [39].

3. MEKANSAL ANALİZ ARACI

Trafik kazalarının mekansal analizine yönelik çalışmalarda birincil ve ikincil etki değişik metotlar kullanılarak değerlendirilebilir. Analizlerde kullanılan metotlar ulaşılan sonuçların ele alınışını da etkilemektedir.

Trafik kazalarının analizine yönelik bazı çalışmalar mekansal verilerin gösterimi ile sınırlı kalırken, bazı çalışmalarda hem mekansal verilerin gösterimi hem de mekansal analiz yöntemleri kullanmıştır.

Analiz çalışmalarında birincil etkiyi ele alarak trafik kazalarının yoğunluğunu değerlendirmek üzere kriging ve modelleme yapılması sıklıkla karşılaşılan bir durumdur [13, 15, 16, 37, 40]. Çalışmalarda trafik kazalarının mekansal tanımları diğer mekansal nesneler içerisinde toplanarak, bu mekansal nesnelerin birbirleri ile farklılıkları gözlemlenmiş ve modellenmiştir.

Bir mekansal analiz yöntemi kullanmasa da Eckhardt 2005 yılında, mekansal nitelikleri çok değişkenli modellemede ve lojistik regresyon analizinde kullanarak trafik kazalarının oluşumunu açıklamaya çalışmıştır [13].

2007 yılında Aryaija'da benzer şekilde herhangi bir mekansal analize ihtiyaç duymadan ve kazaların noktasal konumlarını dikkate almadan trafik kazalarını doğrusal modeller ile açıklamaya çalışmıştır [15].

Meliker ise, 2004 yılında çalışmasında trafik kazalarını 2,5 km. X 2,5 km büyüklüğünde 402 hücreye atamış ve hücreler arasındaki oto korelasyonu değerlendirmiştir [16]. Regresyon analizini kullanan Meliker herhangi bir model oluşturmamıştır.

Trafik kazalarını kavşaklar ve yol kesimler için ayrı ayrı geliştirdiği modeller ile tanımlayan Greibe ise, birincil etkiyi konu edinerek doğrusal modellemeyi kullanmıştır [37].

2006 yılında Karaganis trafik kazalarını temsil eden noktaları karayolunun 1'er kilometrelik kesitleri üzerine atamış ve kesitin kaza yoğunluğunu, günüşiği gibi etkenler ile modellemiştir [40]. Karaganis çalışmasının nokta analizi olarak isimlendirse de nokta dağılımını ele almak üzere herhangi bir desen analizi gerçekleştirmemiştir.

2004 yılında Yamada ve Chong ayrı ayrı gerçekleştirdikleri çalışmalarda ise birincil veya ikincil etkinin kazalar ile nedensellik veya ilişkisellikleri değerlendirilmiş ve kaza dağılımındaki rastlantısallık konu edinilmiştir [10, 41].

Yamada çalışmasında, trafik kazalarını yol ağı üzerinde K fonksiyonunu kullanarak incelemiş, ancak modellemeye yönelmemiştir [10]. Yamada'nın bu çalışması gerek birincil gerekse ikincil etkinin değerlendirilmesinin önünü açmıştır.

Chong ise çalışmasında kaza sonuçlarını bağımlı değişken olarak ele almış ve karar ağaçları, yapay sinir ağları ile kazaları sınıflandırarak analiz etmiştir[41].

3.1. Veri Gereksinimleri

Ackoff, insan aklında yer alan bir içeriğin organizasyonel değişiminin verilerden bilgiye; bilgiden, irfana; irfandan bilgeliğe yönelik olduğunu ve bu gelişim esnasında veriler arası ilişkinin insanı önce desenleri daha sonra ise prensipleri kavramaya ulaştırdığını ifade eder [42].

Fenomenler, gözlemler neticesinde elde edilen sayı gibi sembolik değerler ile veri kümeleri şeklinde toplanır. Bilimsel metotlar kullanılarak verilerin organize edilmesi bilgiyi açığa çıkarır. Trafik kaza verilerinden anlamlı bilgiler üretebilmesi için,

verilerin istatistiki metotlar veya veri madenciliği uygulamaları ile incelenmesi gerekmektedir.

Trafik kazalarının mekansal analizinde kullanılacak olan uygulama yazılımı, mekansal veriler (mekansal sınıf tanımları, yol ağı) ile trafik kazalarına ait verileri organize şekilde ele alarak, istatistiki sonuçların görsel formlarda incelenmesini ve veri kümeleri kullanılarak bilgi seviyesinde sonuçların üretilmesini sağlar.

Analiz yazılımının kaza hakkında erişilmesi mümkün tüm verileri ele alabilecek bir veri alt yapısına sahip olması gerekmektedir.

Trafik kazaları hakkında toplanabilecek veriler çeşitlilik gösterir. Kaza oluşum süreci dikkate alındığında trafik kazası verileri aralarındaki muhtemel ilişkiselliği ve nedenselliği de ortaya koyacak şekilde şu şekilde sınıflanabilir:

- Kaza Öncesi Veriler
- Kaza Anı Verileri
- Kaza Sonrası Veriler

3.1.1. Kaza öncesi veriler

Trafik kazası henüz meydana gelmeden önce temin edilen yaya, yolcu, sürücü, araç, yolculuğa ait verilerdir.

Yolculuk, araç ve yol kullanıcılarına (yaya, yolcu ve sürücü) ait verilerin değerlendirilmesi ile kaza öncesinde süregelen etkenlerin kazanın oluşumundaki ve kaza sonucundaki olası etkileri analiz edilebilir.

Bu veriler kazaların oluşumuna yönelik önleyici denetim faaliyetlerinin hedeflerini belirlemekte önem kazanır. Türkiye’de kazanın oluşumu ile ilgili nedensellik Karayolları Trafik Kanunu’nda belirtilen kusurların; yol kullanıcıları, yol ve araç üzerinde paylaştırılması ile gerçekleştirilmektedir.

Kaza öncesi veriler şu sınıflandırma ile incelenebilir:

- Yol kullanıcıları verileri
 - o Yaya
 - Kimlik bilgisi
 - o Yolcu
 - Kimlik bilgisi
 - Araç konumu
 - o Sürücü
 - Kimlik bilgisi
 - Ceza geçmişi
- Araç verileri
 - o Teknik özellikler
- Yolculuk verileri
 - o Güzergah
 - o Süre

3.1.2. Kaza anı verileri

Trafik kazasının meydana gelişini mekan, zaman ve kusur tanımları ile açıklayan verilerdir.

Kaza anı verileri; yol üzerindeki kontrolün yitilmesiyle sonuçlanan aktif veya pasif eylemin gerçekleştirilmesi ile yol kullanıcılarının kontrolü ele alması veya hareketsiz kalması arasında geçen sürede oluşur.

Kaza anı verileri zaman ve mekan tanımlamalarının yanında, kusurun yol kullanıcıları, yol ve araç arasında paylaşılmasını da içerir.

Kaza verileri şu başlıklar altında toplanır:

- Mekan
 - o Yol
 - o Çevre
- Zaman
 - o Tarih
 - o Olay akışı
- Kusur
 - o Yol kullanıcısı
 - Yaya
 - Yolcu
 - Sürücü
 - o Yol
 - o Araç

3.1.3. Kaza sonrası veriler

Trafik kazası neticesinde yol kullanıcılarında ve araçlarda, kaza öncesine kıyasla durumlarında meydana gelen değişiklikleri belirten verilerdir. Kazazedelerin ve araçların kaza sonrası durumları ve meydana gelen hasarlara ait veriler örnek olarak verilebilir.

Kaza sonrası veriler şu başlıklarda sınıflanabilir:

- Sonuç
 - o Ölü sayısı
 - o Yaralı sayısı
 - o Maddi hasar

Klasik kaza analiz çalışmaları; kaza oluşumu, kaza yoğunluğu veya kaza sonuçları ile kaza verileri arasındaki ilişkiyi açıklamaya dayanır. Bu ilişkisellik regresyon analizleri ile ortaya konulabilmektedir.

Mekansal analizlerde ilişkisellik, herhangi bir mekansal sınıfın kazaların mekansal dağılımı üzerindeki etkisi ile sorgulanır. Sorgulanan bu etkinin kazaların düzlem veya yol ağı üzerindeki rastlantısal dağılımlarına etkiyerek kazaların kümelenme göstermesine sebep olup olmadığı araştırılır.

Mekansal analiz sürecinin geleneksel analiz sürecinden farklı veri ihtiyaçları mevcuttur. Bu ihtiyaçlar doğrultusunda kazaların analizi sürecinde tanımlı olan veri grupları şunlardır:

3.1.4. Analiz öncesi veriler

Kazaların mekansal dağılımının ve dağılımla ilişkili mekansal öğelerin analiz edilmesi için ihtiyaç duyulan ve kullanıcı tarafından gözlem neticesinde elde edilerek uygulamaya aktarılması gereken verilerdir.

Çizelge 3.1. Mekansal analiz fonksiyonlarının analiz öncesi veri ihtiyaç tablosu

	Tanımlayıcı İstatistikler	En Yakın Komşu İndeksi	Global K	Çapraz K	Yerel K
Kaza Koordinatları	Gerekli	Gerekli	Gerekli	Gerekli	Gerekli
Kaza Öncesi Veriler	Seçimli	Seçimli	Seçimli	Seçimli	Seçimli
Kaza Verileri	Seçimli	Seçimli	Seçimli	Seçimli	Seçimli
Kaza Sonrası Veriler	Seçimli	Seçimli	Seçimli	Seçimli	Seçimli
Mekansal Sınıf Koordinatları	Gereksiz	Gereksiz	Gereksiz	Gerekli	Gerekli
Yol Ağı Tanımı	Seçimli	Seçimli	Seçimli	Seçimli	Gerekli
Çalışma Alanı Tanımı	Gereksiz	Gereksiz	Gerekli	Seçimli	Gereksiz

3.1.5. Analiz verileri

Kazaların mekansal dağılımlarının analiz edilmesi için işlem görececek olan verilerle birlikte, işlemlerde parametre olarak kullanılan verilerdir:

- *En büyük R değeri:* K değerinin üretilmesi esnasında kaza sayımlarında dikkate alınacak en büyük çapı ifade eder. R değerinin büyüklüğü sonuçlar üzerinde doğrudan etkilidir. Baddeley R değerinin çalışma alanının en kısa kenarının $\frac{1}{4}$ 'nü aşmaması gerektiğini ifade etmiştir [45]. Bu çalışmada üretilen uygulama yazılımında bu değer kullanıcıdan gizlenerek kaynak kodu içerisinde tutulur.
- *R sayım adeti:* 0 ile en büyük R değeri aralığında gerçekleştirilecek kaza sayımı adetidir. En büyük R değeri b sayıya bölünerek sayımlar arası R aralığı hesaplanır.
- *Rastgele örneklem sayısı:* K değerlerinin rastlantısal sonuçlarla karşılaştırılması için üretilen rastlantısal veri seti sayısıdır.
- *Güven aralığı:* K değerlerinin rastlantısal sonuçlarla yorumlanmasında yüzde üzerinden normal dağılımı temsil edecek olan aralıktır.
- *Referans noktaları arası mesafe:* Yerel Ripley K değerinin hesaplanmasında yol ağı üzerinde üretilen referans noktaları arasındaki olası en uzun mesafedir. Yol ağı üzerindeki bağlantıya bağlı olarak referans noktaları arası mesafe bu rakamdan küçük olabilir ancak büyük olamaz. Referans noktaları arasındaki mesafenin belirlenmesinde kullanılan konumlama sisteminin hassasiyeti de dikkate alınmalıdır.

3.1.6. Analiz sonuçları

Uygulama yazılımının analiz neticesinde ham olarak ürettiği ve kullanıcıya sunulmayıp uygulama yazılımının gösterim fonksiyonlarına aktarılan sonuçlardır.

3.1.7. Analiz gösterim verileri

Uygulama yazılımının gösterim fonksiyonlarına gelen ham sonuçların kullanıcıların rahatlıkla inceleyip, değerlendirebileceği şekle getirilmiş son durumlarıdır.

Analiz verisinin bu son hali sıklıkla çıktı olarak adlandırılır.

Bu çalışmada üretilen uygulama yazılımının ürettiği kazaların mekansal dağılımını gösterir çıktılar şunlardır:

- Kazalar arası kuş uçuşu mesafe listesi,
- Kazalar arası yol ağı mesafe listesi,
- Kazalar arası kuş uçuşu mesafe istatistikleri,
- Kazalar arası yol ağı mesafe istatistikleri,
- Kazalar arası kuş uçuşu mesafe histogramı,
- Kazalar arası yol ağı mesafe histogramı,
- Kaza merkezi haritası,
- Kaza noktaları kuş uçuşu en yakın komşu indeksi,
- Kaza noktaları yol ağı en yakın komşu indeksi,
- Kaza noktaları kuş uçuşu global K grafiği,
- Kaza noktaları yol ağı global K grafiği,
- Mekansal sınıfa göre kaza dağılımlarının çapraz K grafiği,
- Referans noktalarına göre kaza dağılımlarının yol ağı üzerinde kümelenmesini gösterir yerel K haritası,
- Referans noktalarına mekansal sınıf öğelerinin en yakın mesafesi,
- Referans noktası K değeri, referans noktası rastlantısal ortalama K değeri,
- Referans noktası rastlantısal yüzdelik K değeri,
- Referans noktasında kümelenme durumunu gösterir veri listesi

3.2. Fonksiyonel Gereksinimler

Trafik kazalarının mekansal analizi; kaza verilerden oluşan örneklemin organize edilmesini, tanımlayıcı nokta desen hesaplamalarının yapılmasını, kaza dağılımlarının belirlenmesini, sonuçların görsel formlarda incelenmesini ve veri kümeleri içinde bilgi değerine sahip veri kümelerinin üretilmesini gerektirir.

Bunun yanında son kullanıcı, uygulama yazılımının ürettiği sonuçları istatistiki ve mekansal hesaplamalardan bağımsız ve habersiz bir şekilde inceleyebilmelidir.

Yapılan tanım doğrultusunda trafik kazalarının mekansal analizinde kullanılacak olan uygulama yazılımından;

- Mekansal analiz çalışmasına ait çalışma verilerini saklaması,
- Kaza veri tiplerindeki (nümerik, nominal) farklılığı da dikkate alarak kaza verilerini saklaması,
- Trafik kazaları verilerinin filtrelenerek alt kümelerinin oluşturulması,
- Yol katmanının analizlerde kullanılabilecek şekilde üretilmesi,
- Trafik kazalarının mekansal dağılımı ve kümelenmeler ile ilgili sonuçlar üretmesi,
- Analiz sonuçlarını kıyaslanabilir arayüzler ile kullanıcıya sunabilmesi beklenmektedir.

3.2.1. Mekansal analiz çalışmasına ait verilerin saklanması

Son kullanıcı daha sonra kullanabilmek üzere mekansal analiz çalışmasında kaza verileri dışında kullandığı diğer verileri saklayabilmelidir.

Mekansal analiz çalışmasında kaza verileri haricinde kullanılan diğer veriler şunlardır:

- *Çalışmanın tanımlayıcı ismi:* Gerçekleştirilen çalışmayı diğer çalışmalardan ayırt etmek üzere kullanıcı tarafından anlamlı bir metin veya sayı dizesi şeklinde verilebilen metin değeridir.
- *Aralık sayısı:* K değerinin belirlenmesinde maksimum R değeri ile 0 sayısı arasında yer alan tam sayıdır. Her bir aralık için K değeri ayrı ayrı hesaplanır.
- *Rastgele örneklem seti sayısı:* Kümelenmenin varlığını belirlemek üzere kaza örnekleme ile kıyaslanmak üzere üretilecek örneklem seti sayısıdır. Rastgele örneklem seti tam sayı ile ifade edilir. Çalışmalar esnasında rastgelelik için en az 100 set oluşturulmuştur. Hassasiyetin daha önemli olduğu çalışmalarda 1000 değeri daha uygundur. Rastgele örneklem seti sayısı ile uygulama yazılımının cevap süresi arasında doğru orantı mevcuttur.
- *Güven aralığı:* Rastgele örnekleme gözlemlenen K değerlerinin dışında kalan kısmı belirlemek üzere kullanılır. Güven aralığının üzerinde kalan değerlerin kümelenme, altında kalanların ise düzenlilik gösterdiği kabul edilir. Güven aralığının içinde olanlar ise rastgele dağılıma sahiptir. Güven aralığı tam sayıdır.
- *Referans aralığı:* Çalışmada ele alınan yol ağının bölümlenmesinde belirlenecek olan en kısa aralığı ifade eder. Kullanılan konumlandırma sisteminden kaynaklanan hatalarda dikkate alınarak referans aralığı tam sayı olarak belirlenebilir.
- *Harita katman verisi:* Veri girişi ve analiz süreçlerinde kullanılan harita arayüzlerinde gösterilmek üzere shapefile “.shp” formatında olan mekansal referans verilerini içerir dosyalar çalışma içerisinde tanımlanabilir. Veri girişi esnasında katman verisinin kullanılmamasının analiz üzerinde herhangi bir etkisi yoktur.

Mekansal analiz çalışmasına ait veriler daha sonra kullanılmak üzere diskte saklanabilir ve üzerinde değişiklik yapılabilir.

3.2.2. Kaza verilerinin saklanması

Son kullanıcı mekansal analiz uygulama yazılımı ile kaza verilerini analizlerde kullanmak üzere tanımlayabilmeli, düzeltebilmeli ve saklayabilmelidir. Saklanan kaza verileri şunlardır:

- *Kaza konum bilgisi:* Kazanın meydana geldiği yerin WGS 84 formatında koordinat karşılığıdır. Kaza dağılımlarının noktasal analizi bu veri ile gerçekleştirilir. Veri dakika saniye formatı ile değil, rasyonel sayı formatında sağlanır ve saklanır.
- *Kazaya ait nümerik veriler:* Kazanın öncesi, oluşumu ve sonrasında elde edilen, üzerinde matematiksel işlemlerin yapılabildiği veri sınıflarıdır. Ölümlü ve yaralamalı kazalara için tutulan Kaza Tespit Tutanağında yer alan nümerik veriler Ek-4 Çizelge 4.1’de verilmiştir.
- *Kazaya ait nominal veriler:* Trafik kazalarının oluşumunu ve sonuçlarını açıklamak üzere kullanılan ve üzerinde sadece mantıksal işlemlerin yapılabildiği veri sınıflarıdır. Ölümlü ve yaralamalı trafik kazaları için üretilen ve saklanan nominal veriler Ek-4 Çizelge 4.2’de verilmiştir.

3.2.3. Kazaların filtrelenmesi

Örneklemin alt kümelerle bölünerek, mekansal dağılımına yönelik kıyaslamalar yapılabilmesi için analiz edilecek kaza verileri kümesinin mantıksal işlemlerle sınırlandırılması gerekmektedir. Uygulama yazılımı barındırdığı filtreleme fonksiyonları ile örneklemin alt kümelerle ayrılmasını sağlamalıdır.

3.2.4. Yol katmanının üretilmesi

Uygulama yazılımı yol ağı üzerinde gerçekleştirilecek olan hesaplamalarda kullanılmak üzere ağ tanımının yapılmasına imkan sağlayacak arayüzlere sahip olmalıdır.

Yol ağı düğümlerden ve bağlantılardan oluşur. Bağlantılar düğümler arasında olası erişimleri mesafe ile birlikte tanımlar ve ağırlıklandırır. Yol ağı bilgisi kaza verileri gibi daha sonra düzeltilmek veya incelenmek üzere saklanır.

3.2.5. Trafik kazalarının mekansal dağılım sonuçlarının üretilmesi ve gösterimi

Trafik kazalarının düzlemde ve yol ağı üzerinde dağılımlarına ait sonuçların üretilmesidir. Kazaların mekansal analizi ile üretilecek bilgiler şunlardır:

- *En Yakın Komşu İndeksi*: Kazaların düzlem ve yol ağı üzerindeki yakınlaşmaları rakamsal bir değer ile ifade edilir.
- *Global K*: Trafik kazalarının düzlem veya yol ağı üzerindeki dağılımlarının kümelenme; rastlantısallık veya düzenlilik gösterip göstermediği grafik üzerinde rastgele üretilen setlere ait dağılım verileri ile kıyaslanarak belirlenmesini sağlar. Global K değeri sadece kazaların kendi içerisindeki ikincil etkisini kontrol etmek üzere kullanılır.
- *Çapraz K*: Mekansal sınıfa göre kazaların dağılımındaki değişikliği belirlemek üzere kullanılır.
- *Yerel K*: Yol ağı üzerinde oluşturulan referans noktalarının her biri üzerinde kazaların mekansal dağılımının hesaplanmasına dayanır.

Mekansal analiz uygulama yazılımının fonksiyonel gereksinimleri şunlardır:

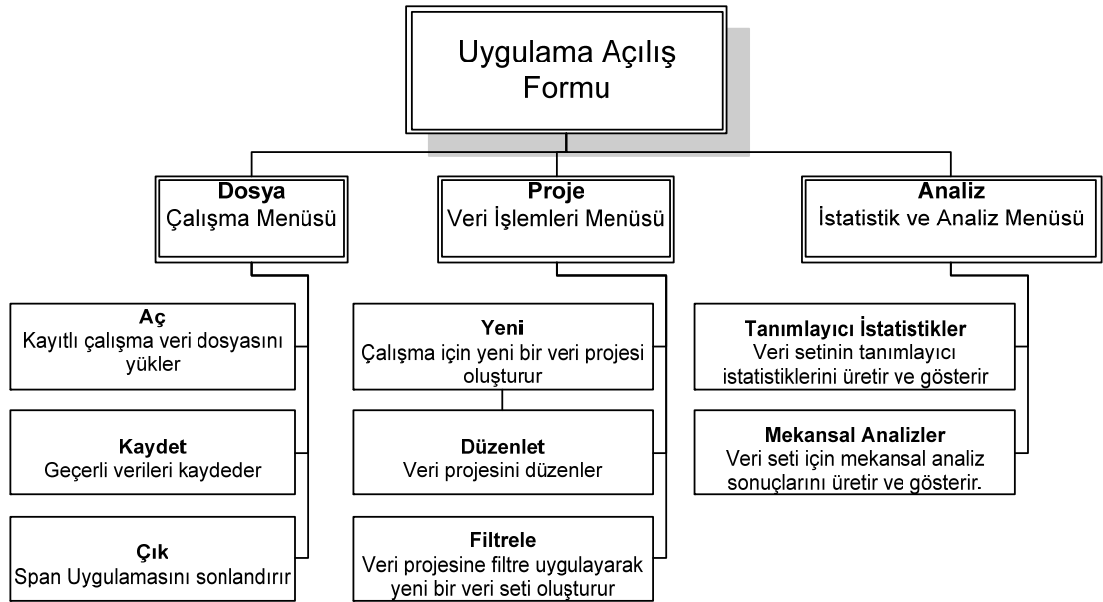
1. Analiz çalışmasına ait verilerin düzenlenmesini ve kaydedilmesini sağlamak,
2. Önceden kaydedilmiş olan verileri tekrar yüklemek,
3. Kullanıcının oluşturduğu “.shp” formatındaki haritaları görüntülemek,
4. Kazalara ait nominal veri tiplerinin tanımlamasını sağlamak,
5. Kazalara ait nümerik veri tiplerinin tanımlamasını sağlamak,
6. Her bir kaza için mekansal konum bilgisinin tanımlanmasını sağlamak,
7. Kaza noktalarını haritada göstermek,
8. Kazalar için nümerik ve nominal veri girişine imkan sağlamak,
9. Kazalara ait; nümerik, nominal ve konum bilgilerinin değiştirilmesine ve nümerik, nominal verilerinin silinmesine imkan sağlamak,

10. Kazalara ait nümerik, nominal ve konum bilgilerini kaydetmek,
11. Çapraz K analizinde kullanılacak mekansal sınıflara ait koordinatların tanımlamasını sağlamak,
12. Mekansal sınıfları harita arayüzü üzerinde göstermek,
13. Mekansal sınıflara ait verilerin kullanıcı tarafından değiştirilmesini ve silinmesini sağlamak,
14. Mekansal sınıfları ve bu sınıflara ait verileri kaydetmek,
15. Yol ağının düğüm ve bağlantılarla harita arayüzü üzerinde tanımlanmasını sağlamak,
16. Yol ağını düzenlemek,
17. Yol ağını kaydetmek,
18. Harita arayüzleri üzerinde yaklaşma ve uzaklaşma yapmak,
19. Kazalar arası mesafeleri düzlem ve yol ağı üzerinde listelemek,
20. Kazalar arası mesafeleri düzlem ve yol ağı için hesaplamak ve minimum, maksimum, ortalama, varyans, standart sapma, basıklık ve çarpıklık verilerini tanımlayıcı nokta desen istatistikleri için üretmek,
21. Kazalararası düzlem ve yol ağı mesafelerini histogram üzerinde göstermek,
22. Kaza merkezini harita üzerinde göstermek,
23. Düzlem ve yol ağı üzerinde kazalar için en yakın komşu indeksini hesaplamak,
24. Düzlem ve yol ağı üzerinde kazalar ve rastgele örneklem için Global Ripley K değerini hesaplamak, grafik üzerinde göstermek,
25. Çapraz K değerini hesaplamak ve sonuçları grafik şeklinde göstermek,
26. Yol ağı referans noktalarında yerel K değerini hesaplamak, harita üzerinde göstermek,
27. Her bir referans noktası için her bir mekansal sınıfın en yakın ögesine olan uzaklığı listelemek.

3.3. Kullanıcı Etkileşimi

Bir masaüstü uygulama yazılımı olan Span kullanıcıya menü tabanlı bir erişim sağlar. Menü veri giriş ve mekansal analiz fonksiyonlarını barındıran iki işlevselliğe göre bölümlenmiştir.

Uygulama yazılımının işlevselliğinin formlarda kullanıcı etkileşimine açılmasından kaçınılarak, basit ve kullanımı kolay bir yazılım üretilme amacı güdülmüştür.



Şekil 3.1. Span uygulaması menü yapısı akış diyagramı

Span uygulama yazılımının barındırdığı form sınıfları şunlardır:

Uygulama Ana Formu: Kullanıcının yüklü olan çalışma hakkında bilgiler almasını sağlayan ve diğer formlarla erişim imkanı veren arayüzdür. Form üzerinde etkileşim menü ile gerçekleştirilir.

The screenshot shows a Windows-style application window titled 'Span - 2006_bahar'. The window has a menu bar with 'Dosya', 'Proje', 'Analiz', and 'Yardım'. Below the menu bar is a form with several input fields and their corresponding values:

Bağımlı Örneklem Adedi:	36
Mekansal Sınıf Adedi:	3
Nominal Veri Tipi Adedi:	0
Nümerik Veri Tipi Adedi:	0
Düğüm Adedi:	62
Bağlantı Adedi:	156

At the bottom of the window, there is a button labeled 'Hazır'.

Şekil 3.2. Span uygulama yazılımı açılış formu

Form üzerinde bulunan metin kutuları ile çalışmada bulunan nokta sayısı, dağılım üzerinde etkisi incelenecek sınıf sayısı, nominal veri tipi sayısı, nümerik veri tipi sayısı, yol ağında bulunan mekansal düğüm ve bağlantı sayısı görülebilir.

Veri İşlem Formu: Kullanıcının yapacağı analiz çalışması ile ilgili verileri gireceği formdur. Verilerin tamamı bu form üzerinden girilir. Değiştirme ve güncelleme işlemleri de bu formdan güncelleme arayüzlerini içerecek şekilde türetilmiş güncelleme formu ile gerçekleştirilir.

Şekil 3.3.Span uygulama yazılımı veri düzenleme ekranı

Form üzerinde konu edinilen veri tipine göre özelleşmiş gösterime sahip şu sekmeler mevcuttur:

- *Genel Tanımlamalar:* Formun bu sekmesinde; proje isminin, K değeri ölçülecek aralık sayısının, rastgele üretilecek veri kümesi sayısının, güvenilirlik değerinin, yol ağı üzerinde oluşturulacak referans noktaları arasındaki mesafenin girileceği metin kutuları mevcuttur. Gösterim ve veri giriş ekranlarında bulunan harita bileşenlerinde gösterilecek harici “.shp” dosyalarının adreslenmesi de bu ekranda gerçekleştirilir.
- *Nominal Tipler:* Nominal tiplerin tanımlaması liste bileşeni üzerinde yapılır. Kullanıcı nominal tipin adını ve alacağı olası değerleri virgül ile ayırarak yazar. Eğer bir nominal tipi silmek istiyor ise bunu ilgili tipi seçip klavyeden silme tuşuna basarak gerçekleştirir.
- *Nümerik Tipler:* Kullanıcı nümerik veri tiplerini isimlerini liste bileşenine bu sekmede ekler. Nümerik tipin silinmesi de nominal tiplerde olduğu gibi

silinmek istenen tipin liste üzerinden seçilip klavyeden silme tuşuna basılması ile gerçekleştirilir.

- *Dağılımı Kontrol Edilecek Veriler:* Nümerik ve nominal tiplerin tanımlanmasını takiben kaza verileri bağımlı veri olarak girilir. Bu ekranda ögenin enlem ve boylam koordinatları ile birlikte nominal ve nümerik veri tiplerinin değerleri de girilir.

Bağımlı Değişken Tanımı

Tanım
No: 10

Koordinatlar
Enlem: 32.73354
Boylam: 39.96647

Nümerik Değerler

No	İsim	Değer
1	Yaralı sayısı	2
2	Ölü sayısı	1

Nominal Değerler

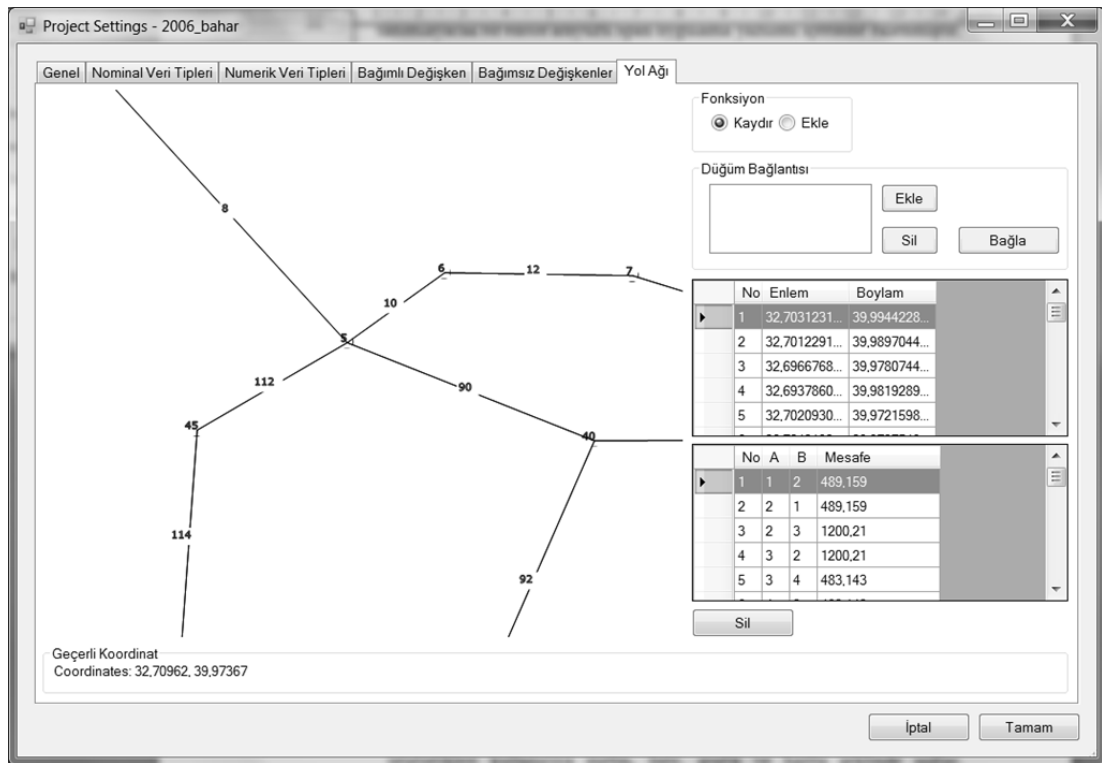
No	İsim	Değer
1	Aydınlatma durumu	Işık var

İptal Edit

Şekil 3.4. Span uygulama yazılımı kaza verisi giriş ekranı

- *Dağılımda Etkisi Araştırılacak Veriler:* Kazaların dağılımı üzerindeki etkisi araştırılacak olan mekansal veri sınıfının gözlemlenen alan içindeki koordinatları bu ekranda tanımlanır.
- *Ağ Tanımları:* Trafik kazalarının yol ağı üzerinde meydana gelmesi nokta dağılımlarının değerlendirilmesinde yol ağının dikkate alınmasını zorunlu kılmaktadır. Yol ağı tanımları her ne kadar harita katmanları içerisinde yer alıyor olsa da, yapılan ön çalışmada temin edilebilen harita dosyalarında yolların bir ağ oluşturacak şekilde tanımlanmadığı görülmüştür. Elde edilen

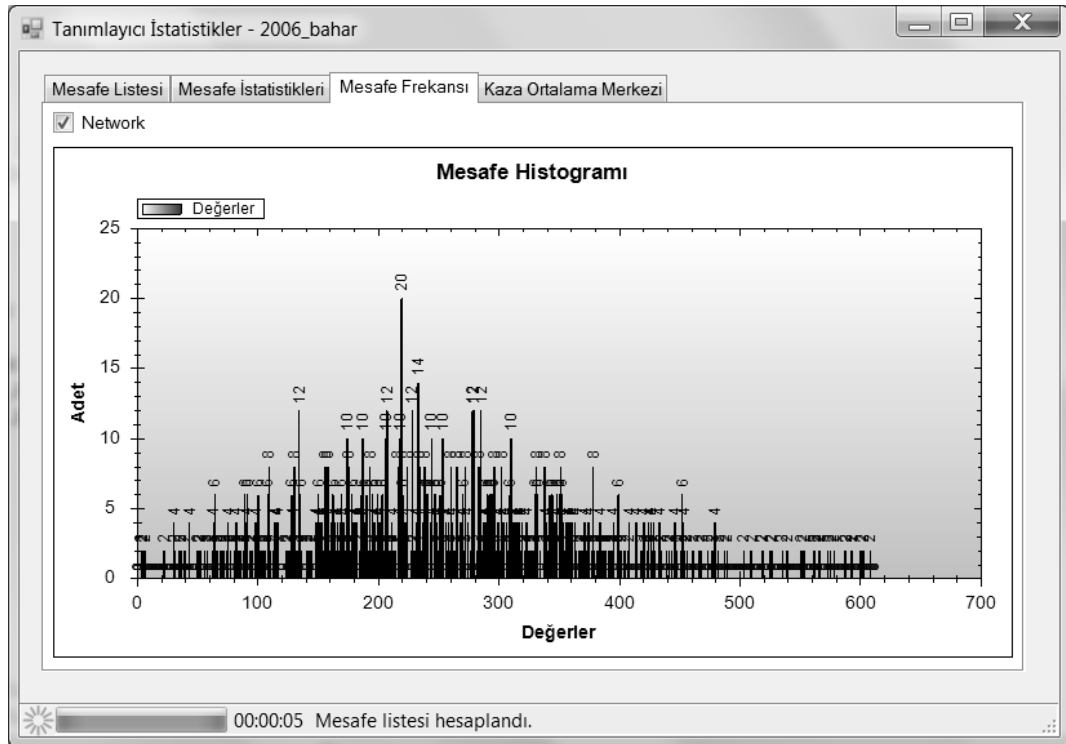
birçok dijital harita da yol bağlantıları ortak düğümler üzerinde sonlandırılmamış, sadece bağlantıların ekran ya da baskı üzerinde görünümü dikkate alınmıştır. Yol katmanı verilerindeki bu nitelik eksikliği nedeni ile bir düğümden bir diğerine ulaşabilecek bağlantıların belirlenmesini mümkün olmamıştır. Yol ağı tanımındaki bu veri eksikliği dikkate alınarak yol ağı tanımlayacak bir editör arayüzü Span uygulama yazılımı içerisine eklenmiştir. Kullanıcı bu sayede düğümleri ve bağlantıları tanımlayarak kendi yol ağı verisini oluşturabilmektedir.



Şekil 3.5. Span uygulama yazılımı kaza verisi giriş ekranı

- *Tanımlayıcı Nokta Desen Analizleri Formu*: Kazalar arası mesafe hakkında tanımlayıcı istatistiklere ulaşılan formdur. Form kazalar arası mesafeye ait istatistikleri kullanıcıya metin, liste, grafik ve harita şeklinde sağlar. Tanımlayıcı nokta desen analizleri formuna ait bağlantının açılış formunda bulunan menüden çağırılması ile birlikte tanımlayıcı istatistikleri form üzerinde gösterebilmek için gerekli hesaplamalar doğrudan yapılır. Bu durum

incelenen verinin çokluğuna bağlı olarak formun gösterilmesi için belirli bir süre beklenmesine neden olur. Hesaplamaların tamamlanması ile birlikte form verilerin gösterimine hazır hale gelir. Bekleme süresince gerçekleştirilen işlemlere ait uyarıcı bilgiler formun sol altında yer alan alanda kullanıcıya sunulur.

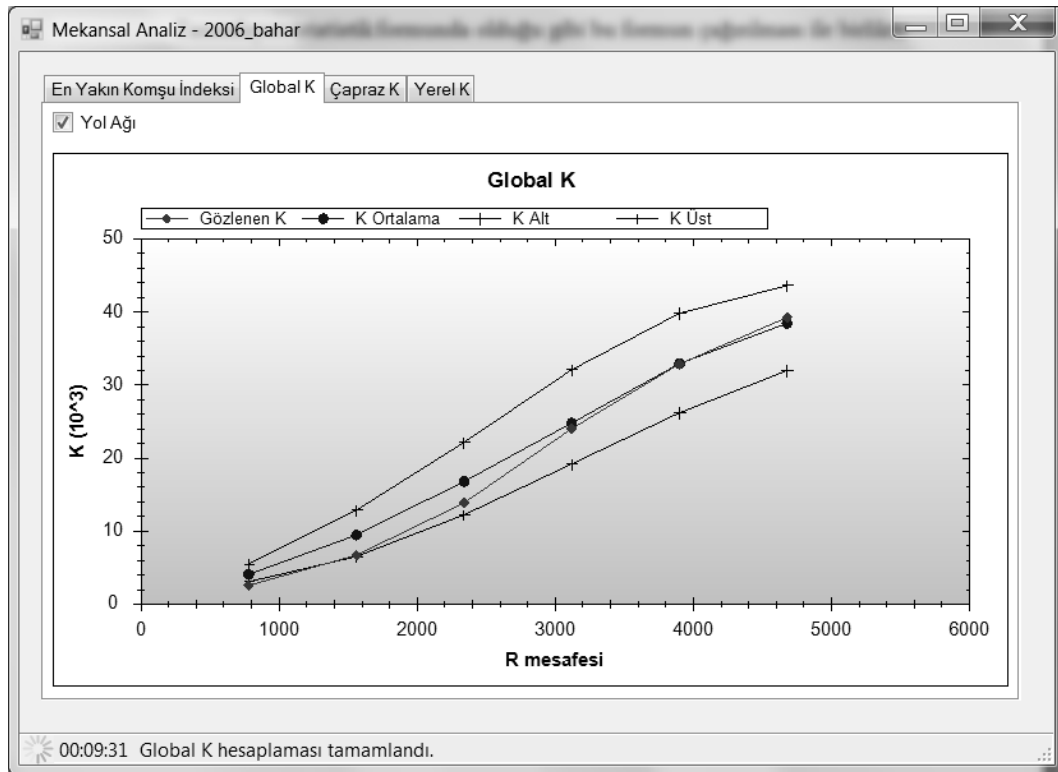


Şekil 3.6. Span uygulama yazılımı tanımlayıcı nokta desen istatistikleri ekranı

Gösterimler düzlem ve yol ağı mesafeleri için düzenlenebilir. Form üzerinde bulunan sekmeler şunlardır:

- *Mesafe Listesi*: Kullanıcı kazalararası düzlem ve yol ağı mesafe listesini görebilir. İstenirse bu bilgiler daha ileri çalışmalarda kullanılmak üzere panoya alınabilir.
- *Mesafe İstatistikleri*: Düzlemsel veya yol ağı üzerindeki mesafe örnekleminin en küçük, en büyük, ortalama değeri, standart sapması, çarpıklık ve basıklığı metin kutuları içerisinde görülebilir.

- *Mesafe Frekansı*: Mesafe örnekleminin histogram üzerinde görüleceği sekmedir. Sekmede olan seçim kutucuğu ile grafik mesafe dağılımı düzlem veya yol ağı verilerini gösterecek şekilde seçilebilir.
- *Ortalama Merkez*: Noktasal dağılımın ortalama merkezinin harita üzerinde gösterildiği sekmedir.

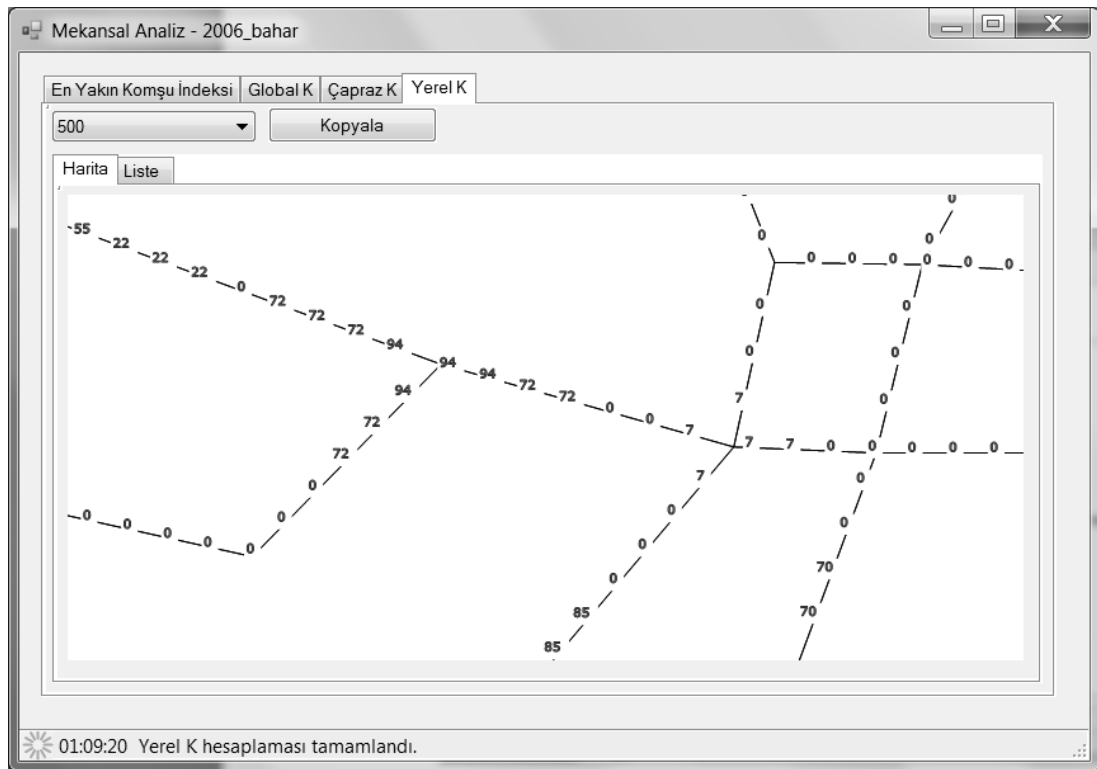


Şekil 3.7. Span uygulama yazılımı mekansal analiz ekranında global K değerlerinin grafik ile gösterimi

- *Mekansal Analiz Formu*: Mekansal analiz sonuçlarının görüleceği ekrandır. Tanımlayıcı istatistik formunda olduğu gibi bu formun çağırılması ile birlikte mekansal analiz hesaplamaları yapılır. Bu nedenle görsel işlevselliğini hesaplamaların tamamlanması ile birlikte kazanır. Verilerden en yakın komşu indeksi metin kutuları içinde, global ve çapraz K sonuçları grafik ile yerel K sonucu ise liste ve harita şeklinde gösterilir.

Formda yer alan şu sekmeler ile mekansal analiz sonuçlarına erişim sağlanır:

- *En Yakın Komşu İndeksi*: Düzlem ve yol ağı üzerinde en yakın komşu indeksinin hesaplanarak kullanıcıya metin kutusu içerisinde sunulduğu sekmedir.
- *Global K*: Kaza dağılımlarının düzlem ve yol ağı üzerinde hesaplanan K değerlerinin rastlantısal dağılımın güven aralığı yüzdelikleri ve ortalaması ile birlikte grafik üzerinde gösterildiği sekmedir.
- *Çapraz K*: Ele alınan mekansal sınıfa göre kaza dağılımlarının düzlem ve yol ağı üzerinde hesaplanan K değerlerinin rastlantısal dağılımın güven aralığı yüzdelikleri ve ortalaması ile birlikte grafik üzerinde gösterildiği sekmedir.



Şekil 3.8. Span uygulama yazılımı mekansal analiz ekranında yerel K değerlerinin harita ile gösterimi

- *Yerel K*: Yol ağı üzerindeki referans noktaları için kaza dağılımlarının yol ağı üzerinde hesaplanan K değerlerinin, rastlantısal sonuçların yüzdeliği içindeki karşılığı ile haritada gösterilmesini sağlayan sekmedir. Yüzdelik sonuç

belirlenen güven aralığının üstünde olması durumunda gösterimler kırmızı renk ile gerçekleştirilir.

3.4. Uygulama Yazılımı Gerçekleştirme Süreci

Uygulama yazılımı doğrusal bir ilerleyişi takip eden çağlayan yazılım geliştirme metodolojisi ile geliştirilmiştir. Mekansal analize yönelik ihtiyaçların tespit edilmesini takiben yazılımının mimarisi belirlenerek, tasarımına geçilmiştir.

Tasarımı takiben kodlama yapılmış, kod parçaları üzerinde birim testleri gerçekleştirilerek formlar üzerindeki etkileşimler içinde kod parçacıkları birleştirilmiş ve ihtiyaçlara cevap veren bir uygulama yazılımı üretilmiştir.

Kodlama aşamasında tasarlanan uygulama mimarisine uygun birbirinden ayrı yazılım projeleri üretilmiştir. Birbirinden ayrı üretilen yazılım kütüphanelerinin işlevselliği; kullanıcı etkileşimini ve formlarını barındıran bir yazılım projesi ile birleştirilmiştir. Bu ayrıma gidilmesi hata ayıklama ve yazılım bakım süreçlerinde karmaşıklığı indirmiştir.

Yazılım geliştirme esnasında kimi fonksiyonların yazılıma katılması diğer fonksiyonlara göre daha fazla önem ve dikkat gerektirmiştir. Bu farklılaşma gereksinim analizi esnasında tespit edilerek risk olarak değerlendirilmiştir. Bu sayede geliştirme esnasında çalışmanın literatür taraması sürecinde üzerinde durulmayan konular yeniden ele alınarak eksiklikler giderilmiştir.

Kodlama esnasında karşılaşılan temel zorluk, geçerli gibi gözüken ve daha önceden üretilmiş bulunan yol ağı haritalarının bu uygulamada kullanılamamasıdır. Bu durum halihazırda var olan yol ağı haritalarının uydu fotoğraflarından üretilirken yol orta çizgisinin, yol bağlantılarını ve düğümlerini nizami bir şekilde tanımlamaması kaynaklıdır.

Sonuçta önceden erişilen “.shp” formatındaki harita verileri kullanılamamıştır. Neticede; analiz için ihtiyaç duyulan harita verisi üretecek coğrafi editör fonksiyonları da uygulama yazılımına eklenmiştir.

Yazılımın geliştirilmesinde kaza noktalarının yol ağı üzerinde meydana geliyor olması, mesafe hesaplamalarında düzlem ve yol ağının ayrı ayrı dikkate alınmasını gerektirmiştir. Düzlem üzerindeki kuş uçuşu ölçümlerde mesafe, nokta ve istikamet açısı hesaplamalarında 1975 yılında Vincenty tarafından yayınlanan ve hassasiyetinin 0,01 milimetre olduğu belirtilen jeodezik mesafe ölçüm metodu benimsenmiştir [46].

Yol ağı üzerinden yapılan mesafe hesaplamalarında en kısa güzergahı bulmak için Dijkstra tarafından 1959 yılında geliştirilmiş olan algoritma kullanılmıştır[47]. Dijkstra’nın algoritmasına göre; başlangıç düğümünün mesafe değeri sıfıra diğer tüm düğümler ise sonsuza eşitlenir, mesafeye göre sıralanan listede en yakın düğümün komşularına mesafe değeri atanır ve bu değer için bir daha hesaplama yapılmaz, tüm düğümler tükenene kadar düğüm için aynı atama işlemi tekrar edilir[47].

Span uygulama yazılımı geliştirilirken üzerinde dikkatle çalışılan diğer bir konu ise yol ağı üzerine rastgele noktaların tanımlanmasıdır. Tanımlama yapılırken Microsoft.Sytem.random sınıfı kullanılmıştır. Atanacak noktanın yol ağı üzerinde olması için öncelikle yol ağını oluşturan linkler uzunluklarına göre ağırlıklandırılmış, ağırlıklandırma esas alınarak link listesinden rastgele bir link seçilmiştir. Daha sonra bu linkin mesafe uzunluğu içinde rastgele bir mesafe seçilerek linkin başlangıç düğümünden bitiş düğümü istikametinde rastgele mesafe uzaklığında bir nokta Vincenty’nin jeodezik metodu ile hesaplanmıştır. Yazılım geliştirme sürecinde 4 farklı yazılım projesinde, 58 Visual C# Kaynak Kod dosyası içinde 5581 satır kod yazılmış, 11 form tasarlanmıştır.

4. BATIKENT’TE MEYDANA GELEN ÖLÜMLÜ ve YARALAMALI TRAFİK KAZALARININ MEKANSAL NİTELİKLERİNDEKİ DÖNEMSEL DEĞİŞİKLİKLER

Geliştirilen uygulama yazılımının etkinliğinin belirlenmesi için gerçek trafik kaza verileri kullanılarak, uygulama yazılımının sunmuş olduğu mekansal analiz fonksiyonları ile kazaların mekansal nitelikleri sorgulanmıştır.

Uygulama yazılımının kullanımında elde edilen sonuçlar neticeler değerlendirme bölümünde ele alınırken kazaların mekansal analizi ile ilgili metodoloji, bulgular ve mekansal analiz çalışmasının değerlendirmesi ise bu bölümde sunulmuştur.

4.1. Materyal

Kazaların mekansal analizi çalışması Batıkent sınırları içerisinde gerçekleştirilmiştir. Bu sınırlandırma esasen Batıkent projesinin Dünya’nın kooperatifçilik sistemi kullanılarak gerçekleştirilen ve başarı ile tamamlanmış en büyük konut üretim pratiği olmasından kaynaklanmaktadır [48].

Ankara’nın batısında yer alan ve tamamı kooperatifçilik sistemi ile üretilmiş evlerden oluşan kent, toplam 10 milyon metrekarelik alanı kapsamaktadır.

Projelendirilerek üretilen bu kentin, taşımış olduğu kimi özelliklerin kazaların mekansal dağılımı üzerindeki olası etkileri açığa çıkaracağı öngörülmüştür.

Kazaların mekansal dağılımına yönelik analiz sonuçlarının trafik kazalarının azaltılmasına yönelik etkileri vurgulayabileceği değerlendirilmiştir.

Çalışmada kullanılan veriler temel olarak kaza ve mekansal tanımlayıcı veriler olarak iki grupta ele alınabilir.

Kullanılan kaza verileri ölümlü ve yaralanmalı trafik kazaları için tutulan trafik kazası tespit tutanağından elde edilen verilerdir. Kaza verilerinin mekansal sınırlaması Batıkent'in bağlı olduğu ilçe tanımı (Yenimahalle) üzerinden yapılmıştır.

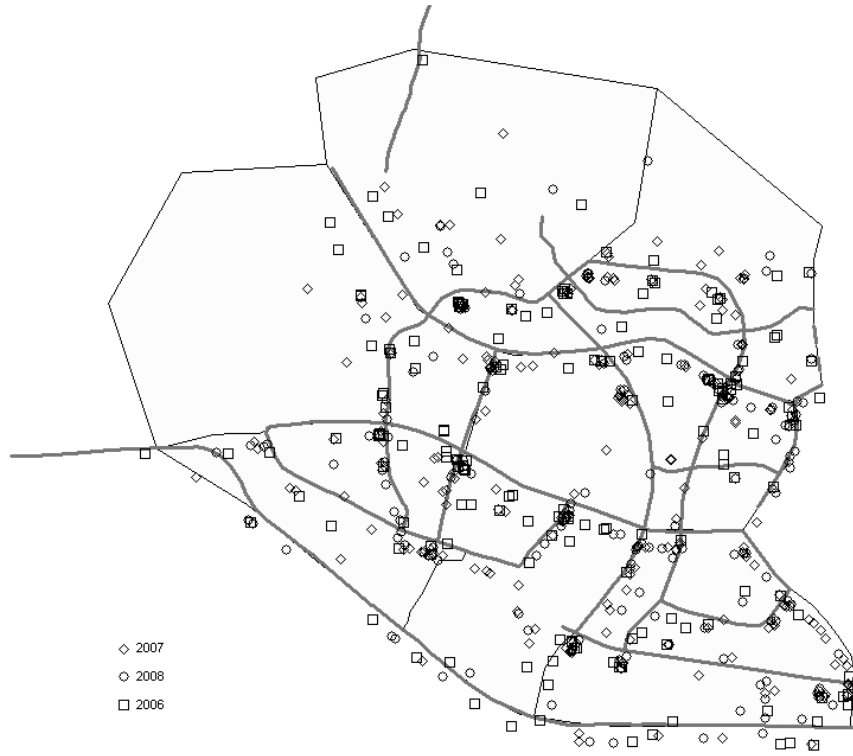
Çalışmada kullanılan kaza verisini tanımlayan diğer bir faktör ise kaza yılları olmuştur.

2003 yılı haziran ayında Trafik Bilgi Sistemi'nin hayata geçmesi ile birlikte ölümlü ve yaralanmalı trafik kazaları için cadde ve sokak tanımlarından oluşan adres verisinin yanında WGS 84 formatında küresel konum verisi de toplanmaya başlamıştır.

Türkiye'de kaza konum verisinin toplanmasına yönelik uygulama 2003 yılı sonuna kadar yaygınlaştırılmış ve 2004 yılı sonuna doğru tutarlı veriler elde edilmeye başlanmıştır.

Çalışmada son 3 yılın (2006, 2007, 2008) ölümlü ve yaralanmalı trafik kazası verileri kullanılmıştır.

Zaman ve mekan sınırlamaları neticesinde toplam 557 adet ölümlü ve yaralanmalı trafik kazasına erişilmiştir. Veriler elektronik ortama aktararak mekansal analiz çalışmasına hazır hale getirilmiştir.



Harita 4.1. Mekansal analiz çalışmasında ele alınan ölümlü ve yaralamalı trafik kazalarının yol ağı üzerinde gösterimi

Çalışmada kullanılan diğer bir veri kümesi ise mekansal sınıfların tanımlandığı verilerdir. Kaza verilerinde olduğu gibi mekansal veriler de Batıkent tanımı ile sınırlanmıştır. Batıkent çevresinde bulunan Ostim iş merkezleri çalışma alanına katılmayarak özellikle çapraz sonuçlara yansıyabilecek olası gürültünün önüne geçilmeye çalışılmıştır.

Mekansal anlamda Batıkent tanımlanırken şu 7 mahallenin sınırları ele alınmıştır:

1. Kent Koop Mahallesi,
2. Kardelen Mahallesi,
3. Batı Sitesi Mahallesi,
4. İnönü Mahallesi,
5. Ergazi Mahallesi,
6. Yeni Batı Mahallesi,
7. İlk Yerleşim Mahallesi



Harita 4.2. Mekansal analiz çalışmasında ele alınan yol ağı ve çalışma alanı

Kazaların mekansal analizinde en çok ihtiyaç duyulan veri tipi yol ağı tanımıdır. Batıkent yol ağı temelde ulaşım ihtiyacını karşılayan ana caddeler ile şekillenmektedir.

Kooperatifçilik sistemi ile oluşturulan sitelerin yoğunlukta olduğu Batıkent'te sokaklar; duvarlarla çevrilmiş olan sitelerin tali çıkışlarını, caddelerle birleştirmektedir.

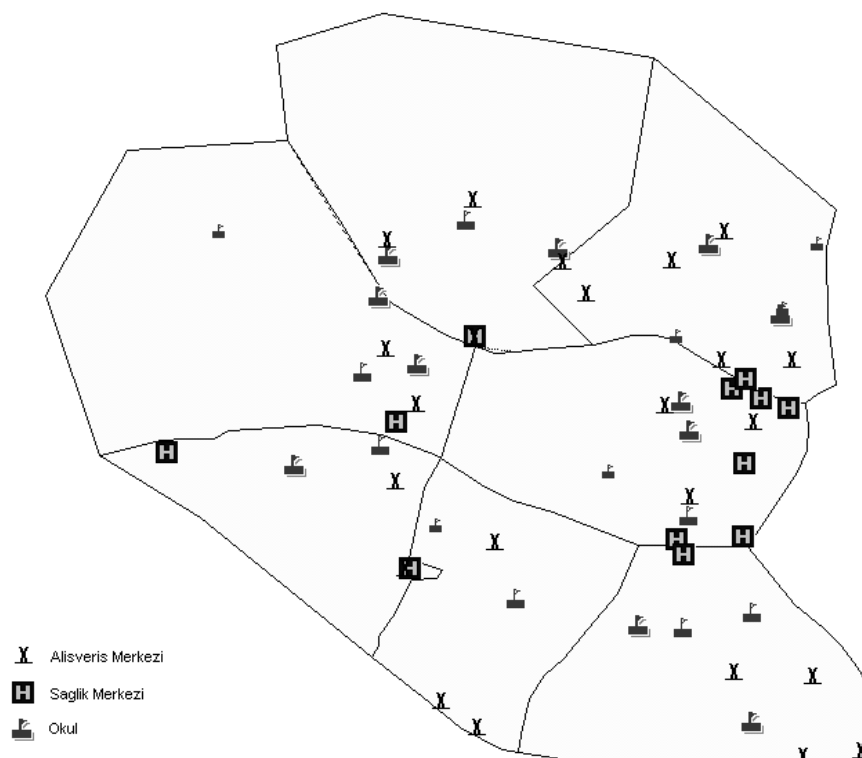
Çalışmada öncelikle sokakların da dahil olduğu, toplam 2194 bağlantıdan oluşan ayrıntılı yol ağı katmanının kullanılması öngörülmüştür.

Ancak Span uygulama yazılımının 233741 metreden oluşan yol ağında 100 metre detay ile lokal K analizlerini normal bir bilgisayarın işlemci gücü ile birkaç haftada tamamlayabileceği belirlenmesi üzerine 78 bağlantıdan oluşan ve Span uygulama

yazılımında 46345 metre ile tanımlanan caddeler ile oluşturulmuş yol ağı kullanılmıştır. Caddelerin belirlenmesinde yol özellikleri veya hizmet seviyesi değil, belediyeninyapmış olduğu isimlendirme dikkate alınmıştır.

Caddelerle tanımlanan ağ üzerinde yıllara ait yerel k analizleri 20-22 saat arasında tamamlanmıştır.

Çalışmada çapraz ilişkilerin sorgulanmasında ele alınan mekansal sınıflar ise sağlık merkezleri, alışveriş merkezleri ve okullar olmuştur. Analiz yazılımının kabiliyetleri diğer bir çok mekansal sınıfın seçimine ve analizine imkan tanıyor olsa da, mekansal sınıfların seçiminde cazibe merkezleri olması göz önünde bulundurulmuştur. Mekansal sınıfların konumları harita ve küresel konumlama cihazı kullanılarak saptanmıştır.



Harita 4.3. Mekansal analiz çalışmasında ele alınan mekansal sınıflar ve çalışma alanı

4.2. Metot

Ele alınan kazaların mekansal analizleri kazalara ait temel tanımlayıcı istatistiklerin belirlenmesine, nokta desen analizlerine ulaşılmasına ve nokta dağılımını gösterir analiz çalışmasının yapılmasına dayanmaktadır.

Öncelikli olarak üretilen kaza verileri Span uygulama yazılımı tarafından erişilebilecek ve okunabilecek Microsoft Excel dosyalarına aktarılmıştır. Kazaların mevsimsel ve yıllık ayırımları Microsoft Excel aracı kullanılarak yapılmıştır.

Sınıflandırılan veriler Span uygulama yazılımı tarafından okunmuş ve yine Span aracı kullanılarak mekansal sınıflara ait veriler çalışma dosyasına eklenmiştir. Yol ağı tanımı Span uygulaması ağ editörü kullanılarak gerçekleştirilmiştir.

Kazaların mekansal dağılımının incelenmesinde en yakın komşu indeksi ve global K değerleri kullanılmıştır. Bu değerlerin üretilmesinde rastgele dağılımı yansıtmak üzere Span uygulama yazılımınca 100 farklı rastgele veri kümesi üretilmesi sağlanmıştır.

Düzlemsel hesaplamalarda bu rastgeleliğin tüm düzlem üzerinde olması sağlarken, yol ağı hesaplamalarında ise rastgele kazaları temsil eden veri kümesi yol ağı üzerinde üretilmiştir.

Benzer şekilde çapraz K hesaplamalarında da her bir mekansal sınıf için ayrı ayrı olmak üzere 100'er farklı kaza noktalarını temsil eden veri kümesi üretilmiştir.

Gözlemlenen K değerinin rastgele örneklemin yüzde 95'lik değerlerinin dışında yer alması durumunda ise ulaşılan değer rastgele dağılımının dışında olduğu kabul edilmiştir.

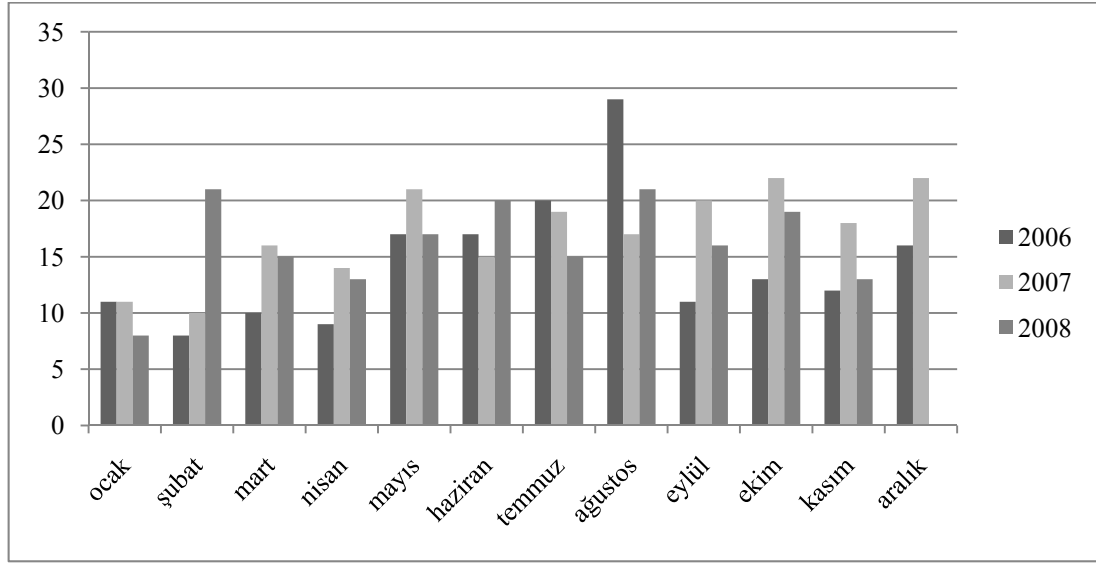
Yerel K hesaplamalarında ise yol ağı üzerinde belirlenen referans noktaları için K değeri hesaplanmıştır. Aynı şekilde rastgele oluşturulan kaza örneklemlerinin sonuçları ile gözlemlenen Ripley K değerinin kıyaslanması ile referans noktasında kaza dağılımları açıklanmıştır.

Analiz çalışmalarında örneklem yıllara ve mevsimlere göre ayrı ayrı alt kümeler oluşturularak incelenmiştir. Mevsim ayrımına gidilirken; aralık, ocak, şubat ayları kış; mart, nisan, mayıs ayları bahar; haziran, temmuz, ağustos ayları yaz; eylül, ekim, kasım ayları güz mevsiminde değerlendirilmiştir.

4.3. Bulgular

Çalışmada kullanılan kaza veri kümesi mekansal dağılıma yönelik tanımlayıcı istatistikler ile kontrol edilmiştir. Bu değerlendirmeler ile gerek ele alınan kümenin büyüklüğü gerek ise kazaların mekansal olmayan dağılımları hakkında bazı sonuçlara erişilmiştir.

2006 yılında Batıkent'te 173, 2007 yılında 205, 2008 yılının 11 ayında ise 178 ölümlü ve yaralamalı trafik kazası meydana gelmiştir. Çalışmada toplam 556 trafik kazası ele alınmıştır.



Şekil 4.1. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının aylara ve yıllara göre dağılımı

Batıkent'te meydana gelen ölümü ve yaralamalı trafik kazalarının aylık dağılımları yıllara göre farklılık göstermektedir.

Kış mevsimi hariç olmak üzere 2007 yılında ölümü ve yaralamalı trafik kazalarında aylık bazda 2006 yılına göre bir artış söz konusudur. 2008 yılında meydana gelen kazalar ise 2007 yılına göre aylık bazda daha azdır.

Çizelge 4.1. Yıllara göre Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının kazaya karışanlara ve kaza sonuçlarına göre dağılımı

Yıllar	Ölü Sürücü Sayısı	Yaralı Sürücü Sayısı	Ölü Yolcu Sayısı	Yaralı Yolcu Sayısı	Ölü Yaya Sayısı	Yaralı Yaya Sayısı
2006	1	102	0	94	0	65
2007	0	107	0	122	3	68
2008	0	101	0	125	0	44
Toplam	1	310	0	341	3	177

Kaza sayısında olduğu gibi kaza sonuçları açısından da 2007 yılı içerisinde kazadan ölüm ya da yaralanma ile etkilenenlerde artış gözlemlenmektedir. 2007 yılında trafik

kazası neticesinde ölen yayalar ile yıllara göre artış gösteren yaralı yolcu sayısı dikkat çekicidir.

Çizelge 4.2. Yıllara göre Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarına karışan araçların sayısı ve yüzdeleri

Yıl	Motosiklet		Otomobil		Minibüs		Otobüs		Jeep		Kamyonet		Kamyon	
	Sayı	%	Sayı	%	Sayı	%	Sayı	%	Sayı	%	Sayı	%	Sayı	%
2006	5	2,89	141	81,5	6	3,47	48	27,75	8	4,62	0	0	12	6,94
2007	13	6,34	165	80,49	9	4,39	49	23,9	7	3,41	1	0,49	17	8,29
2008	7	3,93	150	84,27	7	3,93	44	24,72	11	6,18	0	0	9	5,06

Yıllara göre değışiklik göstermekle birlikte Batıkent'te meydana gelen her 5 kazadan en az dördüne otomobiller karışmaktadır. 2008 yılında sayısal olarak kazaya karışan otomobilde azalma gözlenmekle birlikte yüzde olarak neredeyse 4 puanlık bir artış söz konusudur.

Dikkat çekici bir diğer durum ise 2007 yılında motosikletlerin karıştığı ölümlü ve yaralamalı kazaların bir önceki yıla göre 1,5 kat daha artmış olmasıdır.

Kaza sayımları ve dağılımları üzerinde yapılan istatistiki incelemelerde, yol ağından bağımsız hareket kabiliyetine sahip yaya ve motosikletlilerin kazaya karışmaları dikkate alındığında; yapılacak mekansal analiz çalışmasında kazalar arası mesafenin düzlemde ve yol ağı üzerinde ayrı ayrı değeriendirilmesi gerektiği görülmüştür.

Çizelge 4.3. Yıllara göre Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazaları arasındaki mesafeler

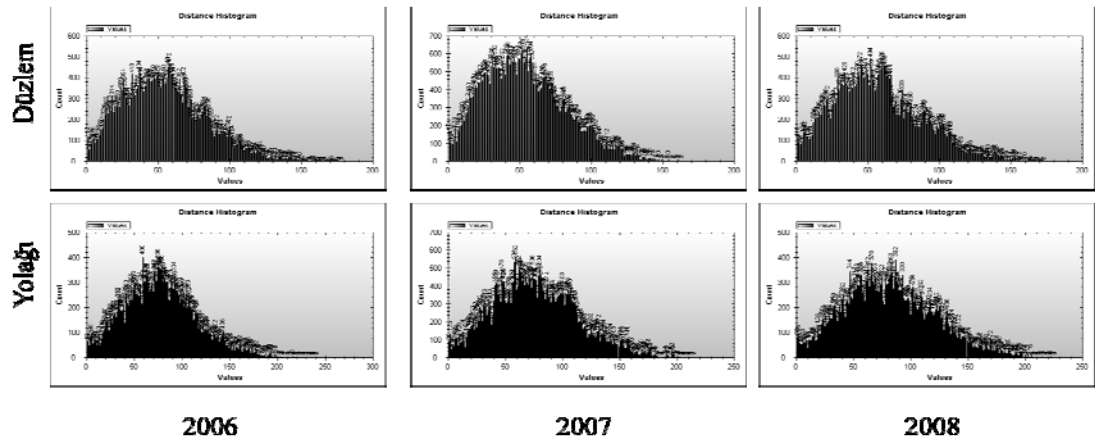
Dönem	Minimum Mesafe (metre)		Maksimum Mesafe (metre)		Ortalama Mesafe (metre)		Standart Sapma	
	Düzlem	Yol	Düzlem	Yol	Düzlem	Yol	Düzlem	Yol
2006 Bahar	34,42	40,6	6805,36	7675,92	2190,64	3244,80	1161,21	1487,05
2006 Yaz	0	39,92	5742,61	9560,15	2105,62	3127,58	1102,38	1605,80
2006 Güz	15,19	64,69	5769,57	7232,57	2109,25	3134,47	1135,43	1521,10
2007 Kış	0	47,12	6289,14	7954,58	2206,79	3130,95	1239,48	1545,16
2007 Bahar	0	63,19	5290,49	7861,83	2070,75	3036,72	1077,87	1441,31
2007 Yaz	29,32	53,25	5707,23	8094,35	2111,10	3024,98	1073,00	1391,49
2007 Güz	0	79,14	5800,14	8476,74	2156,62	3170,51	1166,33	1592,49
2008 Kış	0	65,58	4872,24	7005,49	1898,92	2683,77	970,68	1253,84
2008 Bahar	7,35	46,17	5139,34	7094,31	2225,55	3314,26	1053,44	1497,53
2008 Yaz	0	53,13	5668,80	8148,77	2098,49	2955,61	1146,85	1488,86

Çizelge 4.3'ün incelenmesinden de görüleceği gibi yol ağı ve düzlem kullanılarak gerçekleştirilen mesafe hesaplamaları farklılık göstermektedir. Kazalar arası ortalama mesafe dikkate alındığında bu farklılığın düzlem üzerinden yapılan hesaplamalara göre 1,5 katına çıktığı görülmektedir.

Batıkent bir yerleşim yeri olsa da, Batıkent'te meydana gelen trafik kazalarının mekansal analizlerinde sadece kuş uçuşu hesaplamaların yeterli olamayacağı açıktır.

Bunun yanında yol ağı üzerindeki kazalar arası en uzun mesafeye 2006 yaz döneminde rastlanılmıştır. Yol ağı üzerinde kazalar arası en uzun mesafenin 9560,15 metre olması da dikkate alınarak yol ağı üzerindeki mesafe hesaplamalarında maksimum R değerinin bu değer yaklaşık $\frac{1}{4}$ 'ne karşılık gelen 2500 metre olmasının uygun olacağı değerlendirilmiştir.

Span uygulama yazılımı esas alınarak Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının incelenmesinde histogramlarda kazalar arası mesafe dağılımlarında pozitif bir çarpıklık olduğu gözlemlenmektedir.



Şekil 4.2. Batıkent’te meydana gelen ölümlü ve yaralı trafik kazalarının düzlem ve yolağı üzerindeki mesafe dağılımlarını gösterir histogramlar

Mesafe dağılımlarındaki basıklıklar incelendiğinde dağılımın çoğunlukla normal dağılıma uygunluk gösterdiği görülmektedir.

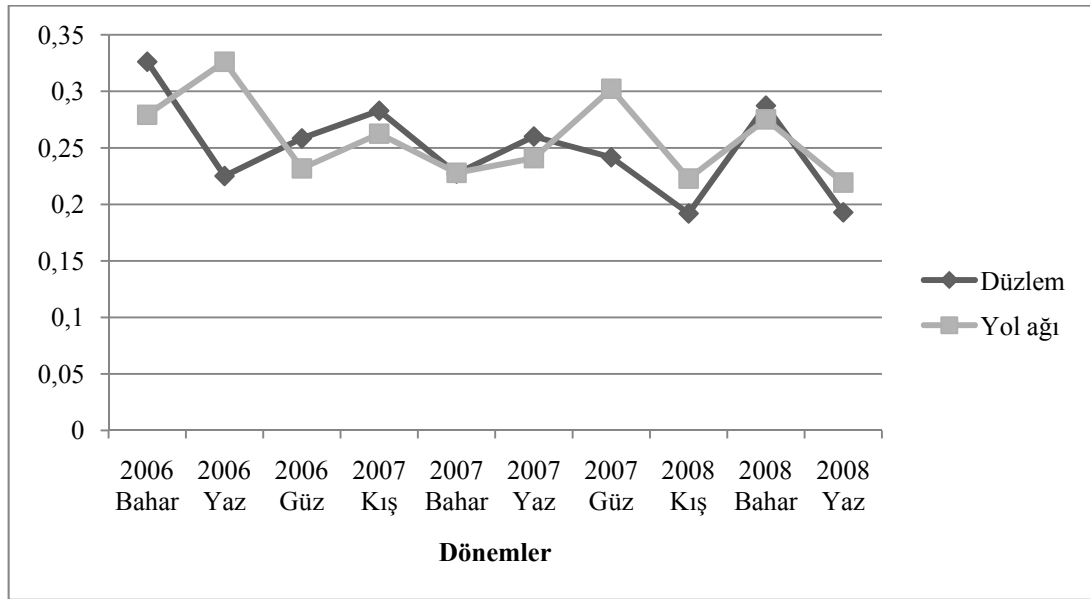
İstisnai olarak 2008 kış ve 2008 bahar döneminde kazalar arası mesafe dağılımları yükseltilmiş kosinüs dağılımına uygunluk göstermektedir. 2008 kış ve 2008 bahar döneminde değişim daha sonra belirtilecek olan K analizlerinde belirlenen kümelenme ile de paralel sonuçlar göstermektedir.

Bu aşamada bulunan K analizleri ile mesafe dağılımları arasındaki örtüşme bu çalışmanın önemli bulguları arasında yer almaktadır.

Çizelge 4.4. Yıllara göre Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazaları arasındaki mesafe dağılımının basıklık değerleri ve dağılım tipi

Dönem	Düzlem Üzerinde Basıklık		Yolağı Üzerinde Basıklık	
	Değer	Dağılım Tipi	Değer	Dağılım Tipi
2006 Bahar	0,633092	Normal dağılım	0,165465	Normal dağılım
2006 Yaz	-0,10231	Normal dağılım	0,417043	Normal dağılım
2006 Güz	0,015129	Normal dağılım	-0,2259	Normal dağılım
2007 Kış	0,10717	Normal dağılım	0,06265	Normal dağılım
2007 Bahar	-0,48257	Normal dağılım	-0,35306	Normal dağılım
2007 Yaz	-0,09624	Normal dağılım	0,026598	Normal dağılım
2007 Güz	-0,2389	Normal dağılım	-0,06517	Normal dağılım
2008 Kış	-0,60655	Yük. Kosinüs dağı.	-0,19595	Normal dağılım
2008 Bahar	-0,5059	Yük. Kosinüs dağı.	-0,53158	Yük. Kosinüs dağı.
2008 Yaz	-0,15085	Normal dağılım	-0,2115	Normal dağılım
2006	0,133046	Normal dağılım	0,182774	Normal dağılım
2007	-0,1038	Normal dağılım	-0,09806	Normal dağılım
2008	0,00063	Normal dağılım	-0,17294	Normal dağılım

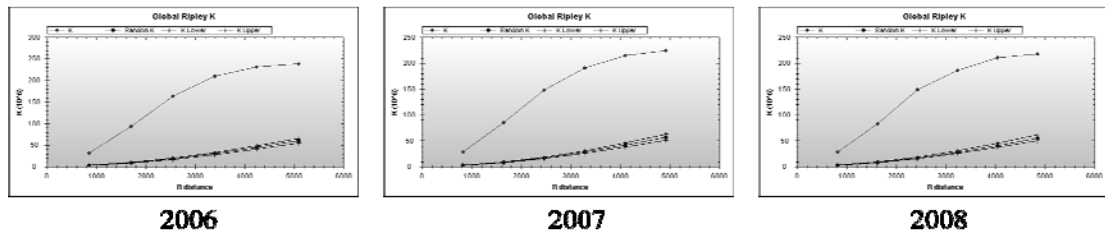
Kaza dağılımları en yakın komşu indeksleri ile incelendiğinde ise; düzlem ve yol ağı üzerinde gerçekleştirilen tüm hesaplamaların 1'in altında olduğu görülmektedir. Kümelenmeyi ifade eden bu durum gerek düzlem gerek ise yol ağı için beklenen bir sonuç olarak değerlendirilmiştir.



Şekil 4.3. Batıkent'te meydana gelen ölümlü ve yaralı trafik kazalarının düzlem ve yolağı üzerindeki en yakın indeksleri

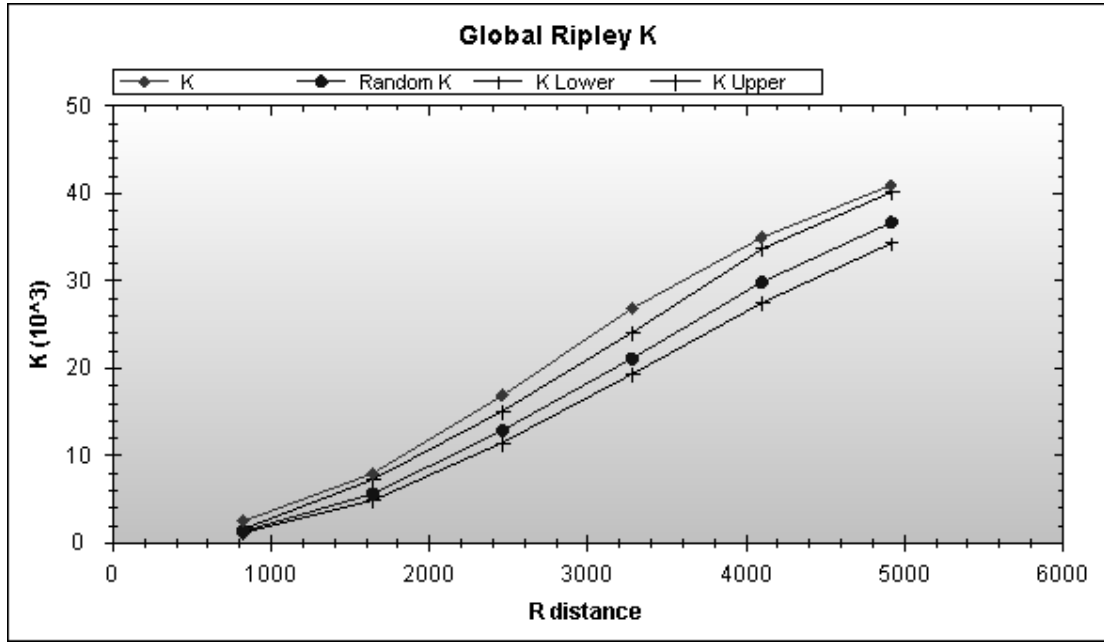
En yakın komşu indeksi değerleri dönemlere göre değişiklik göstermektedir. Bunun yanında düzlem ve yol ağı değerlerinde bu değişikliğin eş yönlü olduğu görülmüştür. Bu bulgunun tersi ancak 2006 yaz dönemi için geçerlidir. Diğer dönemlerde indeks her iki hesaplama şekli içinde önceki döneme göre aynı şekilde değişim göstermiştir.

Düzlem üzerinde global K değerleri incelendiğinde kazaların mekansal kümelenme gösterdiği açıktır. Ulaşılan değerler rastlantısal değerlerin üstündedir. Tüm R değerleri için tüm dönem ve yıllarda düzlem üzerinde kazaların kümelenmesi mevcuttur.



Şekil 4.4. Batıkent'te meydana gelen ölümlü ve yaralı trafik kazalarının yıllara göre düzlemde Global K değerlerini gösterir grafik

Ancak global K hesaplaması yol ağı üzerinde tekrar edildiğinde 2006 ve 2008 yıllarında trafik kazalarının kümelenme göstermediği, kümelenmenin sadece 2007 yılında olduğu belirlenmiştir.



Şekil 4.5. Batıkent'te 2007 yılında meydana gelen ölümlü ve yaralı trafik kazalarının yol ağı üzerinde Global K değerlerini gösterir grafik

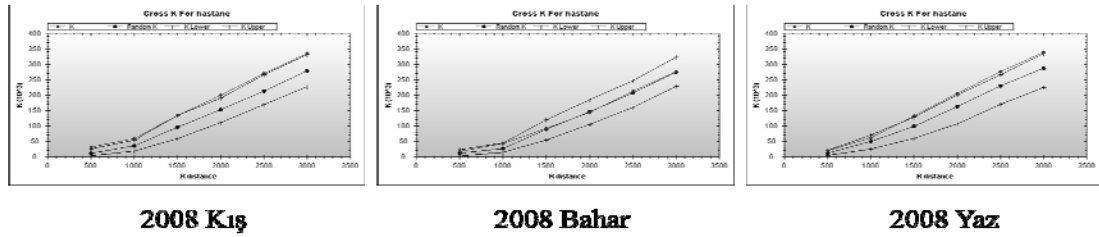
Şekil 4.5.'in incelenmesi ile de görüleceği gibi K değeri normal dağılımın yüzdelik değerlerinin üstündedir. 2007 yılında meydana gelen ölümlü ve yaralı trafik kazalarının ölçüm yapılan tüm R aralıkları için kümelendiği görülmektedir.

Çalışmada alışveriş merkezleri, sağlık merkezleri ve okul çevresinde trafik kazalarının kümelenmesini araştırmak üzere Span uygulama yazılımı ile yol ağı üzerinde çapraz Ripley K değerleri hesaplanmıştır.

12 sağlık merkezi dikkate alınarak gerçekleştirilen hesaplamalar dönemsel ve yıllık seviye de ayrı ayrı ele alınmıştır.

Dönemsel hesaplamalara göre 2008 kış, 2008 bahar ve 2008 yaz aylarında sağlık merkezleri çevresinde özellikle ilk 500 metre içerisinde trafik kazalarının kümелendiği tespit edilmiştir.

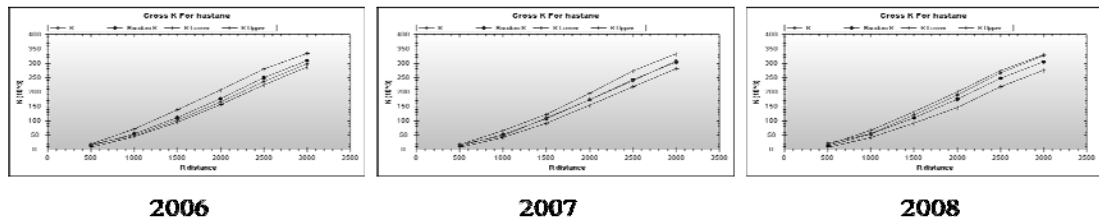
2006 bahar, yaz, güz, 2007 kış, bahar, yaz ve güz dönemlerinde ise sağlık merkezleri çevresinde kaza dağılımları normal dağılımın içerisinde kalmıştır.



Şekil 4.6. Batıkent'te meydana gelen ölümlü ve yaralı trafik kazalarının sağlık merkezleri çevresinde kümelenme tespit edilen dönemlere ait Çapraz Ripley K değerlerini gösterir grafik

Şekil 4.6.'da da görüleceği gibi dönemlerin ardışıklığı dikkate alındığında zamanla sağlık merkezleri çevresinde kazaların normal dağılımdan kümelenmeye doğru geçişi söz konusudur.

Dönemsel hesaplamalar yerine sağlık merkezleri için yıllara göre çapraz K değerleri hesaplandığında ise, kazaların sağlık merkezleri etrafında kümelenmesinde bir yükselme olmakla birlikte yıl içinde değerlerin normal dağılım içinde kaldığı belirlenmiştir.

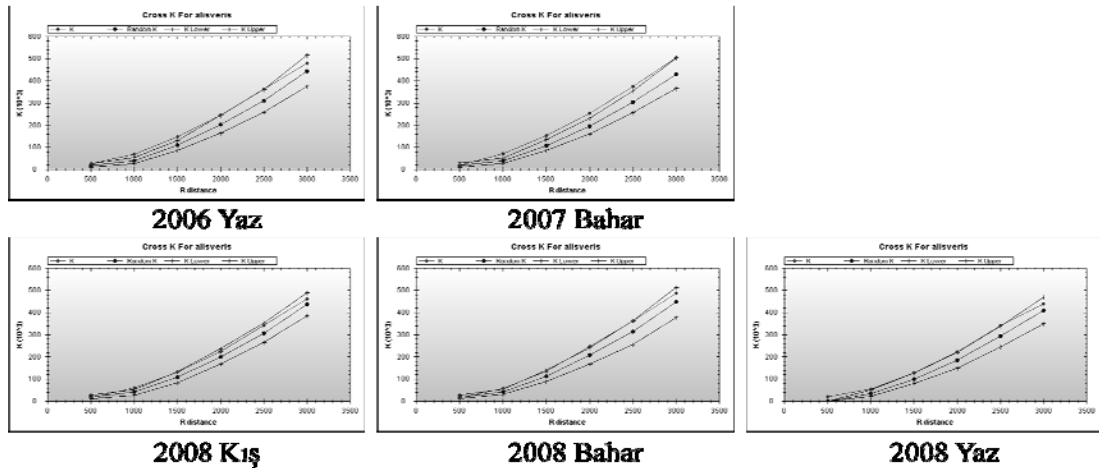


Şekil 4.7. Batıkent'te meydana gelen ölümlü ve yaralı trafik kazalarının yıllara göre sağlık merkezleri esas alınarak hesaplanan Çapraz K değerlerini gösterir grafik

Dönemsel hesaplamalarla yıl hesaplamaları sonucunda erişilen bu farklılık veri kümelerinin büyüklüğüne dayanmaktadır. Veri kümesinin küçüklüğü elde edilen sonuçların hassasiyetini azaltmaktadır.

Span uygulama yazılımı kullanılarak gerçekleştirilecek kazaların dağılımına yönelik analizlerde incelenen olaya ait mümkün olduğunda büyük bir veri setine sahip olmak gerekmektedir.

Alışveriş merkezlerine yönelik çapraz K hesaplamaları da nerede ise sağlık merkezi için ulaşılan sonuçlarla örtüşmektedir. Span uygulama yazılımı ile Batıkent'te bulunan 23 alışveriş merkezi ve çarşı çevresinde trafik kazalarının kümelenmesine yönelik çapraz K değerleri hesaplanmıştır. Hesaplanan değerler rastlantısal noktalar esas alınarak üretilen sonuçlar ile grafik üzerinde yapılan kıyaslanmıştır.

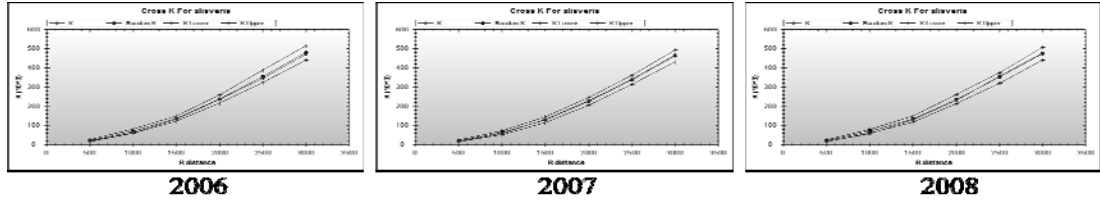


Şekil 4.8. Batıkent'te meydana gelen ölümlü ve yaralı trafik kazalarının alışveriş merkezleri çevresinde kümelenme tespit edilen dönemlere ait Çapraz K değerlerini gösterir grafik

Şekil 4.8.'in incelenmesi ile görüleceği üzere sağlık merkezlerinde olduğu gibi alışveriş merkezlerinde de kazalar 2008 yılı içerisinde kümelenme göstermektedir.

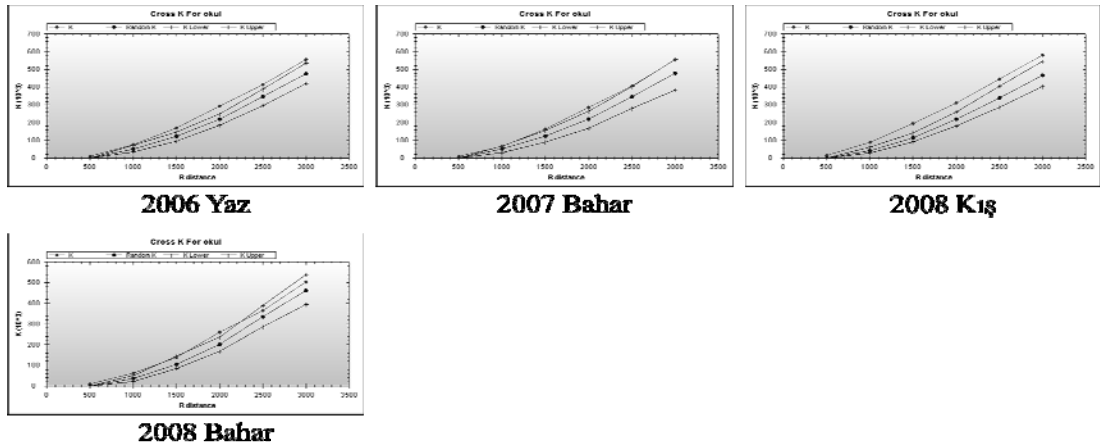
Aynı zamanda 2006 yaz ve 2007 bahar dönemlerinde de alışveriş merkezleri çevresinde yaralı ve ölümlü kazaların kümelenmesi söz konusudur.

Hesaplamalar yıllar üzerinden yapıldığında ise alışveriş merkezleri çevresinde kazaların kümelenmesi söz konusu değildir. Yıllara göre ayrılmış veri setleri esas alındığında alışveriş merkezleri çevresinde trafik kazaları normal dağılım göstermektedir.



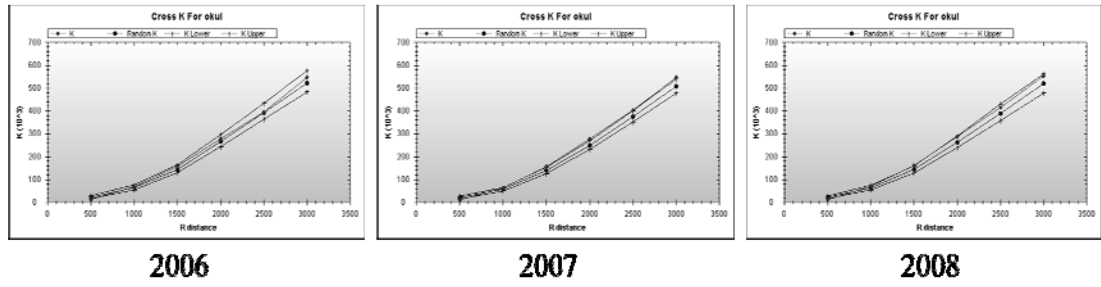
Şekil 4.9. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının yıllara göre alışveriş merkezleri esas alınarak hesaplanan çapraz K değerlerini gösterir grafik

Çalışmada çapraz K hesaplamaları Batıkent'te bulunan 25 okul için tekrar edilmiştir. Alışveriş merkezleri için erişilen sonuçlardan farklı olarak 2008 yılı yaz döneminde okullar çevresinde kaza kümelenmesi tespit edilememiştir.



Şekil 4.10. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının okullar çevresinde kümelenme tespit edilen dönemlere ait Çapraz K değerlerini gösterir grafik

Şekil 4.10.'un incelenmesi ile de görüleceği gibi eğitim öğretim dönemi ile kazaların kümelenmesi arasındaki ilişkiyi ortaya koyacak bir sonuç bulunmamaktadır.



Şekil 4.11. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının yıllara göre okullar esas alınarak hesaplanan çapraz K değerlerini gösterir grafik

Okullar için yıllara göre çapraz Ripley K sonuçları incelendiğinde, özellikle ilk 500 metre için yapılan hesaplamalarda kaza dağılımlarının düzenlilik gösterdiği gözlemlenmektedir.

Okullar çevresinde görülen kaza dağılımı düzenliliğinin kazalardaki yoğunluk artış veya azalışından çok kaza dağılımlarını forma sokan etkilere kaynaklandığı değerlendirilmektedir. Detaylı incelemelerle görüleceği gibi mesafe 1000 metreye çıktığında dağılım, normal dağılımı yakalamakta, 1500 metre için ise kazalar da tekrardan kümelenme başlamaktadır.



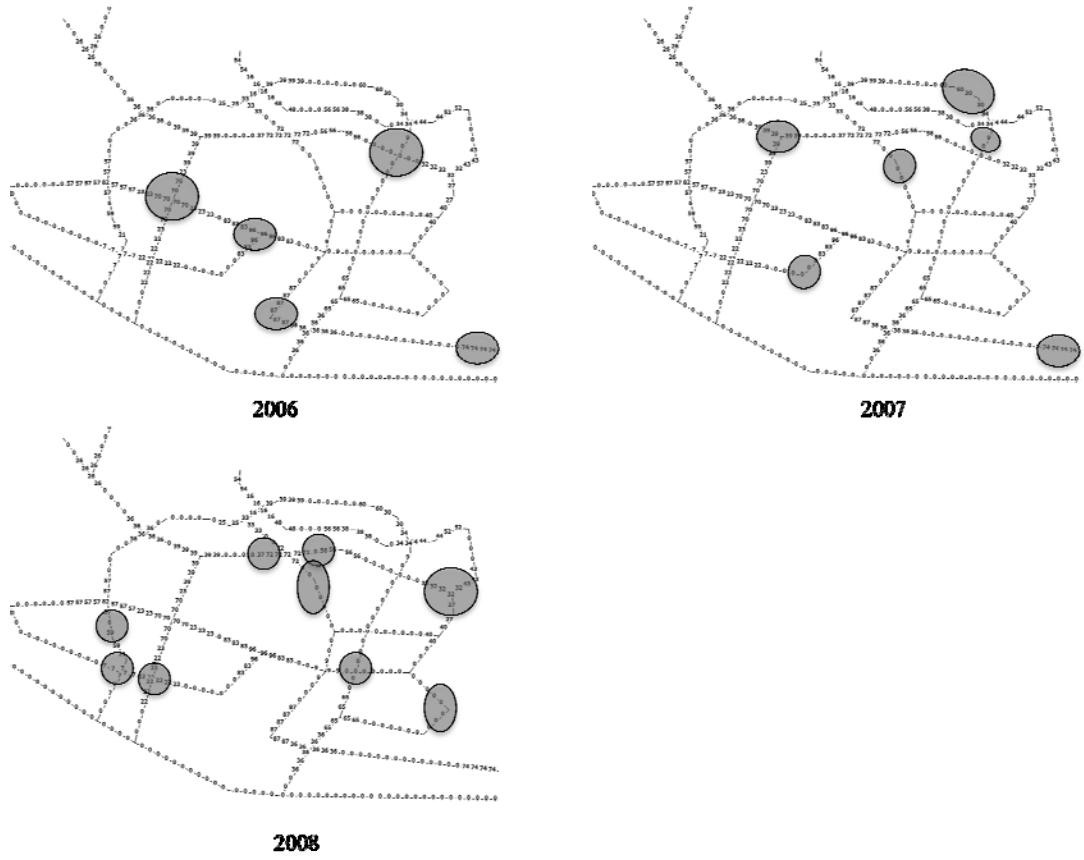
Harita 4.4. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının yıllara göre kaza merkezlerini gösterir harita

Harita 4.4.'ün incelenmesi ile Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının merkezlerinin yıllara göre farklılık gösterdiği görülebilir.

Merkez güney doğu doğrultusunda 2007 yılında 220, 2008 yılında ise 150 metre kayma göstermiştir. Batıkent'in kuzeyinde yeni sitelerin yerleşime açılmasına rağmen kaza merkezlerinin güney doğuya doğru kayması dikkat çekicidir.

Ulaşılan bulgular arasında yol ağı üzerinde oluşturulan referans noktaları için hesaplanan lokal K değerleri en önemli sonuçları yansıtmaktadır.

K değerleri esas alındığında gerek dönemsel sonuçlar gerek ise yıllara ait sonuçlar dikkate alındığında tüm hesaplamalarda yol ağının çeşitli noktalarında kaza kümelenmelerinin olduğu tespit edilmiştir.



Harita 4.5. Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının yıllara göre yol ağı üzerindeki kümelenmelerini gösterir haritalar

Kazalara ait kümelenme Yol ağı üzerinde neredeyse sadece kavşaklarda gözlemlenmektedir. Her ne kadar ölümlü ve yaralamalı trafik kazalarının kümelenmesi yıllara göre değişiklik göstermekte ise de, değişim daha çok kümelenmenin başka kavşaklara kayması şeklinde kendini göstermektedir.

Kümelerin kayması kimi zaman komşu kavşaklara sıçrama şeklinde olurken, kimi zaman ise kümenin daha alt kümelere ayrılıp çevre kavşaklara da sıçrama şeklindedir.

2006 yılında; 5. Cadde ile 11. Cadde'nin kesiştiği kavşakta, 5. Cadde ile 241. Sokak'ın giriş kavşağında, Batıkent Bulvarı ile 1. Cadde'nin kesiştiği kavşakta ve

Abdülhalik Renda Caddesi'nin yataya geçen kavşağında kazalar kümelenme göstermektedir.

2007 yılında ise; yatay 1. ve 5. caddelerin dikey 12. Caddeyi kestiği güneybatı da yer alan kavşaklarda, Başkent Bulvarı ile Batıkent Bulvarının kesiştiği kavşakta, 5. Cadde 6. Cadde ve Rajiv Gandhi Caddesi'nin kesiştiği kavşakta, Başkent Bulvarı, Alinteri Bulvarı ve 6. Caddesi kesiştiği ve Batıkent'in doğu girişi olan kavşaklarda kümelenme gözlemlenmiştir.

2008 yılında ise kümelenme 2006 ve 2007 yıllarına göre daha yaygınlaşmıştır. Özellikle 11 Cadde üzerindeki kümelenmeler dikkat çekicidir. Yine Başkent ve Batıkent bulvarlarının kesiştiği kavşağın öncesinde ve sonrasında kümelenmelere rastlamak mümkündür. Fakat bu bulvarların kesiştiği kavşak içerisinde kümelenme yoktur. 2006 ve 2007 yıllarında Başkent Bulvarı, Alinteri Bulvarı ve 6. Caddesinin kesiştiği kavşaktaki kümelenme ise daha doğudaki 100 Yıl Bulvarı ile Başkent Bulvarı arasındaki kavşağa kaymıştır. 2008 yılında gözlemlenen bir diğer kümelenme ise Rajiv Gandhi Caddesi ile 3. Cadde arasındaki kavşaktadır.

5. SONUÇ VE ÖNERİLER

Kazaların sebeplerine yönelik analiz çalışmaları, trafik güvenliğini artırıcı birçok çabanın yönünü belirlemektedir. Geleneksel anlamda sürdürülen bu çalışmalarda kaza öncesi ve kaza anına ait veriler ile kaza ve sonuçları arasında ilişkiler test edilerek, anlamlı bulunan sonuçlar trafik güvenliği politikalarına yansıtılmaktadır.

Ancak kazaların kendi arasındaki ilişkiyi (oto korelasyon), kazaların mekansal sınıflar ile olan mekansal ilişkisini ya da kazaların yol ağı ile olan ilişkisini sorgulayan çalışmalara nadiren rastlanmaktadır.

Trafik kazalarının mekansal analizine yönelik bu eksikliği gidermek üzere düzlem ve yol ağı üzerinde kaza dağılımının analizini yapacak ve sonuçları kullanıcıya uygun arayüzler ile gösterecek kullanılabilir bir uygulama yazılımının geliştirilmesi bu çalışmanın hedefi olmuştur. Bu kapsamda nokta desenlerinin analiz edilmesini konu edinen çalışmalar incelenerek, kazaların mekansal analizinde kullanılacak metot belirlenmiştir. Bu metot ile kaza analizi yapan bir yazılım üretilmiştir.

Geliştirilen yazılım kullanılarak; Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazaları analiz edilmiş ve analiz neticesinde şu sonuçlara ulaşılmıştır:

Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının kazalar arası mesafe dağılımları genellikle normal dağılım göstermektedir. İstisnai olarak dağılımın yükseltilmiş kosinüs dağılımı gösterdiği durumlara nadiren rastlanmıştır ve ayrıca bu dağılımın belirginliği net değildir.

Kaza dağılımlarının en yakın komşu indeksi hesaplamaları ele alındığında kazaların düzlem ve yol ağı üzerinde kümelenme gösterdikleri belirlenmiştir.

Düzlem üzerinde yapılan global K hesaplamaları ile de aynı yönde sonuçlara erişilmiş ve Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazalarının

kümelenme gösterdiği görülmüştür. Düzlem üzerindeki kümelenmenin esas nedeni ele alınan çalışma alanında incelenen kaza örnekleminin yol ağı üzerinde gerçekleşiyor olmasıdır. Yol ağı; araç ve yol kullanıcılarının hareketlerini düzenleyen ve sınırlayan bir çerçeve çizmektedir.

Yol ağı üzerindeki global K hesaplamaları; dönemlere ve yıllara göre farklılık göstermektedir. 2007 yılı hariç olmak üzere Batıkent'te meydana gelen ölümlü ve yaralamalı trafik kazaları yol ağı üzerinde rastgele dağılmıştır. 2007 yılındaki farklılık; aynı yıl başlayan metro ve büyük inşaat çalışmaları nedeni ile kapanan ve yükü tali yollara aktarmak üzere yapılan trafik düzenlemeleri ile ilişkilendirilmektedir.

2008 kış, bahar ve yaz dönemlerinde ardışık olarak yol ağı üzerinde görülen kaza kümelenmeleri ise bir eğilim olarak değerlendirilmektedir. Yerel K hesaplamaları da dikkate alınarak; bu eğilimin başlıca nedeninin yine trafik akışının yol kullanıcılarının etkileşimini arttıracak şekilde düzenlenmesi olduğu düşünülmüştür.

Kazalar ile mekansal sınıflar arasındaki ilişkilerin sorgulandığı Çapraz K analizleri neticesinde her mekansal sınıf için farklı bulgulara erişilmiştir. Batıkent'te genel anlamda dönemsel olarak kış, bahar ve yaz dönemlerinde mekansal sınıfa göre değişen şekilde kümelenme tespit etmek mümkün iken; sonuçlar yıl bazında ele alındığında, sağlık merkezleri ve alışveriş merkezleri için kümelenmenin söz konusu olmadığı görülmektedir.

Yerleşimin bir plan dahilinde başladığı Batıkent'te, alışveriş ve sağlık merkezleri çevresinde kaza dağılımlarında rastlantısallık gözlemlenmektedir. Sonuç olarak planlamanın trafik kazalarının mekansal dağılımı üzerindeki etkisi, rastlantısallık şeklinde görülmektedir.

Okullar çevresinde gerçekleştirilen analizlerde eğitim öğretim sürelerinde kazaların kümelenmesi sorgulanmış ancak bu yönde bir sonuca erişilememiştir. Okullar

evresinde kmelenmenin olmayışı ve nispeten yakın mesafelerde (<1000 metre) kaza dağılımlarında dzenliliğın söz konusu olması, okulların yer seimlerinin, cadde ve sokaklara aılıř şekillerinin, işaretleme ve önlemlerin, altyapı tercihlerinin etkisine dikkat çekmektedir. Neticede okullar uygulanan güvenlik önlemleri ile kaza dağılımları dzenliğı arasında bir ilişki gözlemlenmiştir.

Kaza merkezleri dikkate alındığında ise; Batıkent'in kuzeyinde yeni yerleşim merkezleri açılmasına rağmen kaza merkezlerinin güney doğuya doğru kayması dikkat çekicidir.

Yol ağı üzerinde gerçekleştirilen yerel K hesaplamaları Batıkent yol ağı üzerinde yıllara göre değıřen bir kaza dağılımının söz konusu olduğunu göstermektedir. Kavşakların trafik lambalarının kontrolünde olmasına ve yol işaretlemelerinin bulunmasına rağmen neredeyse kmelenmelerin tamamı kavşaklardır.

Dzenleyici önlemler ile yatay ve dikey düzlemde trafik etkileşiminin artırılması kmelenmeyi arttırmakta ve kmelenmeyi, daha küçük ve görünürlüğün daha az olduğu kavşaklara taşımaktadır.

Analiz çalışması neticesinde trafik düzenlemelerinin kaza kmelenmelerindeki etkisinin ancak yıl gibi uzun periyotlar içinde gözlemlenebildiğı sonucuna da erişilmiştir.

Yerel K hesaplamalarına göre kaza kmelenmelerinin kavşaklarda gözlemlenmesi ise; kavşakların yol kullanıcıları arasındaki etkileşim görevini yüklenirken daha dzenleyici ve aktif bir role sahip olması gerektiğini göstermektedir.

Sonuç olarak; planlamalar neticesinde kazaların mekansal dağılımları rastgele dağılım göstermekte, güvenlik önlemleri ile kaza dağılımları dzenlilik sergilemektedir.

Yazılımın Batıkent kaza analizlerinde kullanımı ile; örneklem sayısının (n) 100'ü aştığı durumlarda kullanılan işlemci modeline göre değişiklik göstermekle birlikte toplam işlem süresi 5 saate yaklaştığı görülmüştür. Benzer bir etki ise yol ağında bulunan bağlantı sayısındaki artış için söz konusudur. Sonuçta; gerçekleştirilecek çalışmalarda örneklem sayısının 100 sayısı ile sınırlanması, yol ağında bulunacak bağlantı sayısının ise 200'ü geçmemesine özen gösterilmesi performans açısından önemlidir. Her halükarda bu rakamların aşılması durumunda programatik bir hata ile karşılaşılması için gerekli önlemler alınarak değişken tanımları uygun genişlikte seçilmiştir.

Batıkent çalışması ve sonuçları değerlendirildiğinde; kazaların mekansal analizi için uygulanan metodun ve üretilen uygulama yazılımının kazaların mekansal analiz çalışmalarını karşılayabilecek nitelikte olduğu sonucuna erişilmiştir.

KAYNAKLAR

1. Peden, M., Scurfield, R., Sleet, D., Mohan, D., Hyder, A. A., Jarawan, E., Mathers, C., "The global impact", World Report on Road Traffic Injury Prevention, **World Health Organization**, Geneva, 34 (2004).
2. İnternet: Emniyet Genel Müdürlüğü "Genel Kaza İstatistikleri" http://www.trafik.gov.tr/istatistikler/10_yil_istatistik.aspx (2011).
3. İnternet: Türkiye İstatistik Kurumu "Genel Kaza İstatistikleri" http://www.tuik.gov.tr/PreIstatistikTablo.do?istab_id=362 (2011).
4. İnternet: Türkiye İstatistik Kurumu "Trafik Kazalarına Neden Olan Unsurlar" http://www.tuik.gov.tr/PreIstatistikTablo.do?istab_id=365 (2011).
5. Tobler, W. R., "A Computer Movie Simulating Urban Growth in the Detroit Region", **Economic Geography**, 46: 234-240 (1970).
6. Melnick, A., "GIS Data Acquisition and Storage", Melnick Introduction to Geographic Information Systems in Public Health, **Aspen Publishers**, United States of America, 9 (2002).
7. Black, W. R., "Highway Accidents: A Spatial and Temporal Analysis." **Transportation Research Record**, 1318: 75-82 (1991).
8. Levine, N., Kim, K. E., Nitz, L.H., "Spatial Analysis of Honolulu Motor Vehicle Crashes: I. Spatial Patterns", **Accident Analysis and Prevention**, 27(5): 663-674 (1995).
9. Jones, A.P., Langford, I.H., Bentham, G., "The Application of K-function Analysis to the Geographical Distribution of Road Traffic Accident Outcomes in Norfolk, England" **Social Science and Medicine**, 42(6): 879-885 (1996).
10. Yamada, I., Thill, J., "Comparison of planar and network K-functions in traffic accident analysis", **Journal of Transport Geography**, 12: 144-158 (2004).
11. Miaou, S., Song, J.J., Mallick, B.K., "Roadway Traffic Crash Mapping: A Space-Time Modeling Approach", **Journal of Transportation and Statistics**, 6(1): 33-58 (2003).
12. Noland, R.B., Quddus, M.A., "A spatially disaggregate analysis of road casualties in England", **Accident Analysis and Prevention**, 36(6): 973-984 (2004).
13. Eckhardt, N., Thomas, I., "Spatial Nested Scales For Road Accidents In The Periphery Of Brussels", **IATSS Research**, 29(1): 66-78 (2005).

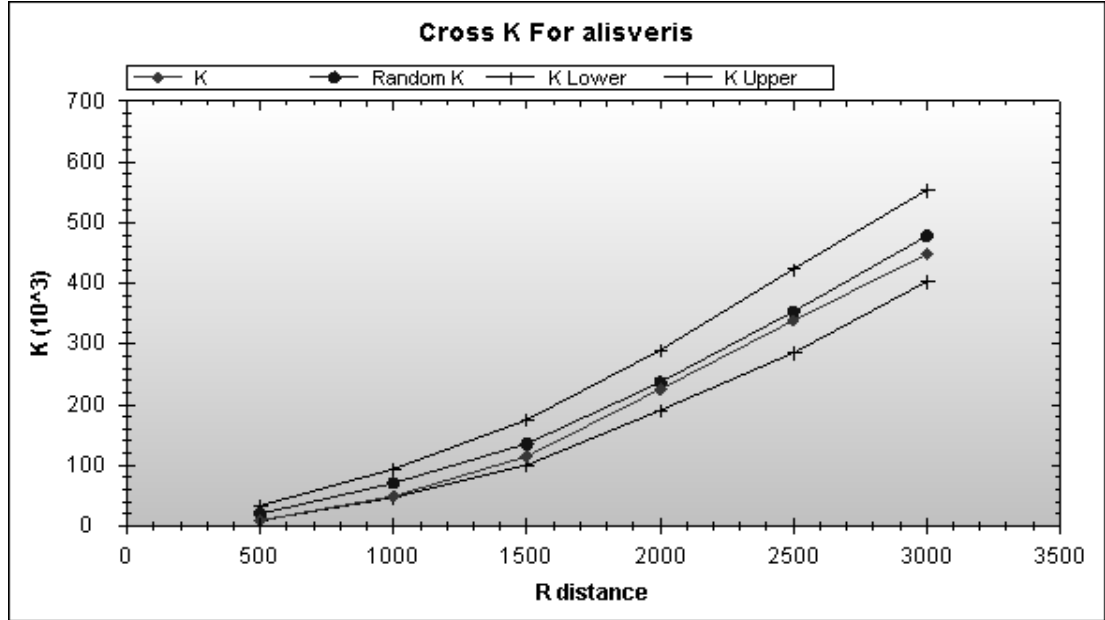
14. Agüero, J., Jovanis, P., "Spatial analysis of fatal and injury crashes in Pennsylvania", *Accident Analysis and Prevention*, 38(3): 618-625 (2006).
15. Aryaija, J., Dissanayake, D., Wedagama, D.M., "Modelling the relationship between child pedestrian accidents and land use", *Traffic Safety Center Newsletters*, 4(2): 6-8 (2007).
16. Meliker, J., Maio, R., Zimmerman, M., "Spatial analysis of alcohol-related motor vehicle crash injuries in southeastern Michigan", *Accident Analysis and Prevention*, 36: 1129-1135 (2004).
17. Geyik, A., "Kaza kara noktalarının tespiti ve değeriendirilmesine yönelik yeni bir yöntemin oluşturulması", Yüksek Lisans Tezi, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 1-2 (2001).
18. Bektaş, S., "Aksaray ili çevre karayollarında meydana gelen trafik kazalarının çok yönlü analizi ve kaza tahmini modeli", Doktora Tezi, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 1-5 (2002).
19. Özkan, K. , Çubuk, K., Özkan, M., "Turizmin Trafik Kazaları Üzerindeki Mekansal Etkisi", *3. Uluslar arası Trafik ve Karayolu Güvenliği Kongresi*, Ankara, 627 (2005).
20. Turnbull, L., A., Coomes, D. A., Hector, A., Rees, M. "Seed mass and the competition/colonization trade-off: competitive interactions and spatial patterns in a guild of annual plants", *Journal of Ecology*, 92(1): 97 (2004).
21. Besag, J., Newell, J., "The detection of clusters in rare diseases", *Journal of the Royal Statistical Society*, 154(1): 143 (1991).
22. Güvenal, B., Çabuk, A., Yavuz, M., "Trafik Kazaları Verilerine Bağlı Olarak Cbs Destekli Ulaşım Planlaması: Eskişehir Kenti Örneği", *2. Mühendislik Ölçmeleri Sempozyumu*, İstanbul, 424-436 (2005).
23. Cliff, A.D., Ord, J.K., "Spatial Processes: Models & Applications", *Pion Ltd.*, London, 5 (1981).
24. Anselin, L., "Spatial Econometrics: Methods and Models", *Kluwer Academic Publishers*, London, 2 (1988).
25. Openshaw, S., "Developing Appropriate Spatial Analysis Methods for GIS", *Geographical Information Systems*, 1, Maguire, D.J., Goodchild, M.F. and Rhind, R.W., *Longman Scientific and Technical*, 390 (1991).

26. Gatrell, C. A., Bailey T. C., Diggle P. J., "Spatial point pattern analysis and its application in geographic epidemiology", *Transactions of the Institute of British Geographers*, 21(1): 260-263 (1996).
27. Waller, L.A., Gotway, C.A., "Applied Spatial Statistics for Public Health Data", *John Wiley & Sons, Inc.*, New Jersey, 130 (2004).
28. Smith, M., Goodchild, M., Longley, P., "Geospatial Analysis - A Comprehensive Guide to Principles, Techniques and Software Tools", *Troubador Publishing Ltd*, Leicester, 135 (2007).
29. Donnelly, K., "Simulations To Determine The Variance And Edge-Effect Of Total Nearest Neighbor Distance", *Simulation Methods in Archeology*, *Cambridge University Press*, London, 91-94 (1978).
30. Clark, P.J., Evans, F.C., "On Some Aspects of Spatial Pattern in Biological Populations", *Science*, 121(3142): 397 (1955).
31. Ripley, B.D., "Spatial statistics", *John Wiley & Sons, Inc.*, New Jersey, 133 (1981).
32. Besag, J., "Contribution to the discussion of Dr Ripley's paper", *Journal of The Royal Statistical Society - Series D*, 39: 193-195 (1977).
33. Goreaud, F., Pélissier, R., "*Spatial Structure Analysis of Heterogeneous Point Patterns: Examples of Application to Forest Stands*", ADS in ADE-4 Topic Documentation, 8(1): 7-8 (2000).
34. Lotwick, H.W., Siverman, B.W., "Methods for analysing spatial process of several types of points", *Journal of The Royal Statistical Society - Series B*, 44: 406-413 (1982).
35. Dixon, P.M., "Ripley's K function", *Encyclopedia of Environmetrics*, *John Wiley & Sons Ltd*, Chichester, 4 (2002).
36. Baddeley, A., Turner, R., "Modelling Spatial Point Patterns in R", *Case Studies in Spatial Point Process Modeling*, 185: 28 (2006).
37. Greibe, P., "Accident Prediction Models For Urban Roads", *Accident Analysis and Prevention*, 35(2): 273-285 (2003).
38. Shekhar, S., Zhang, P., Huang, Y., Vatsavai, R.R., "Trends in Spatial Data Mining, Data Mining: Next Generation Challenges and Future Directions", *Amer Assn for Artificial*, 1-24 (2003).

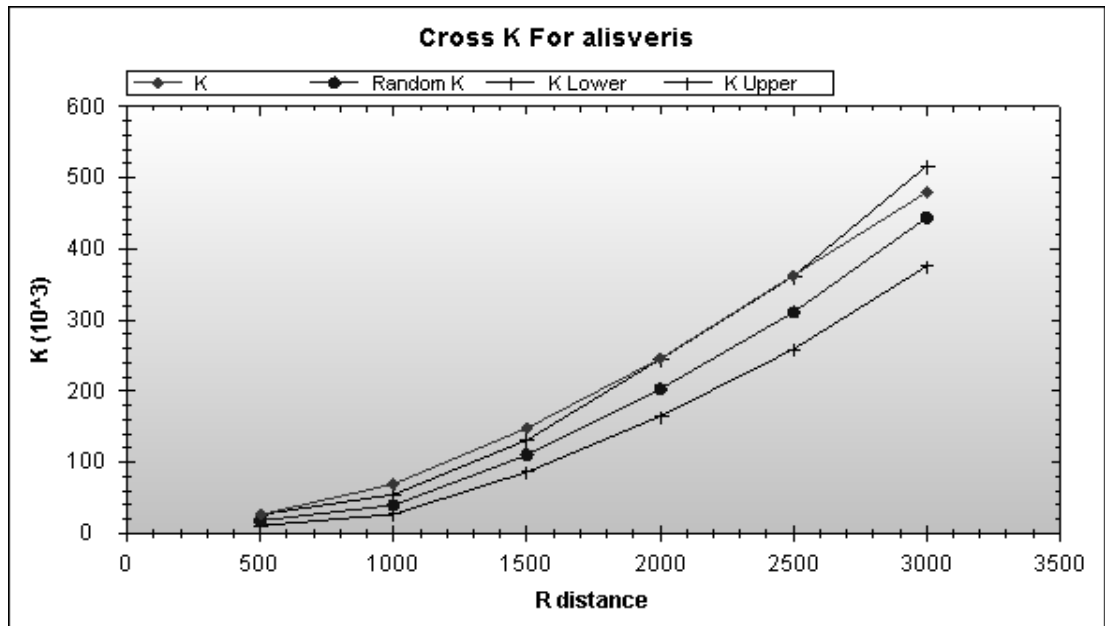
39. Ester, M., Kriegel, H.P., Sander, J., "Algorithms and applications for spatial data mining", *Geographic Data Mining and Knowledge Discovery*, Miller, H., Han, J., **CRC Press**, New York, 160-187 (2001).
40. Karaganis, N., Mimis, A., "A Spatial Process for Estimating the Probability of Occurance of a Traffic Accident", **46th Congress of The European Regional Science Association**, Volos, 640 (2006).
41. Chong, M., Abraham, A., Paprzycki, M., "Traffic Accident Analysis Using Machine Learning Paradigms", **Fourth International Conference on Intelligent Systems Design and Applications (ISDA'04)**, Hungary, 415-420 (2004).
42. Ackoff, R.L. "From Data to Wisdom", *Journal of Applied System Analysis*, 16: 3-9 (1989).
43. Foskett, D.J., "Libraries and Information Systems- A Fruitfull Partnership", **Canadian Association for Information Science Connectedness: Information, Systems, People, Organizations (CAIS 1995)**, Alberta, 15-23 (1995).
44. Scott, D., "On Optimal and Data-Based Histograms", *Biometrika*, 3-66: 605-610 (1979).
45. Baddeley, A.J., "Spatial sampling and censoring", **Stochastic Geometry: Likelihood and Computation**, Chapman and Hall, 37-78 (1998).
46. Vincenty, T., "Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations", *Survey Review*, 22-176 (1975).
47. Dijkstra, E.W., "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik*, 1: 269-271 (1959).
48. Turhal, M., "Real Estate Development Process in Turkey", Yayınlanmamış Master Tezi, Department of Infrastructure, Division of Building and Real, Estate Economics, **Royal Institute of Technology**, Stockholm, 65 (2005).

EKLER

EK-1 Dönemsel çapraz K grafikleri (alışveriş merkezleri)

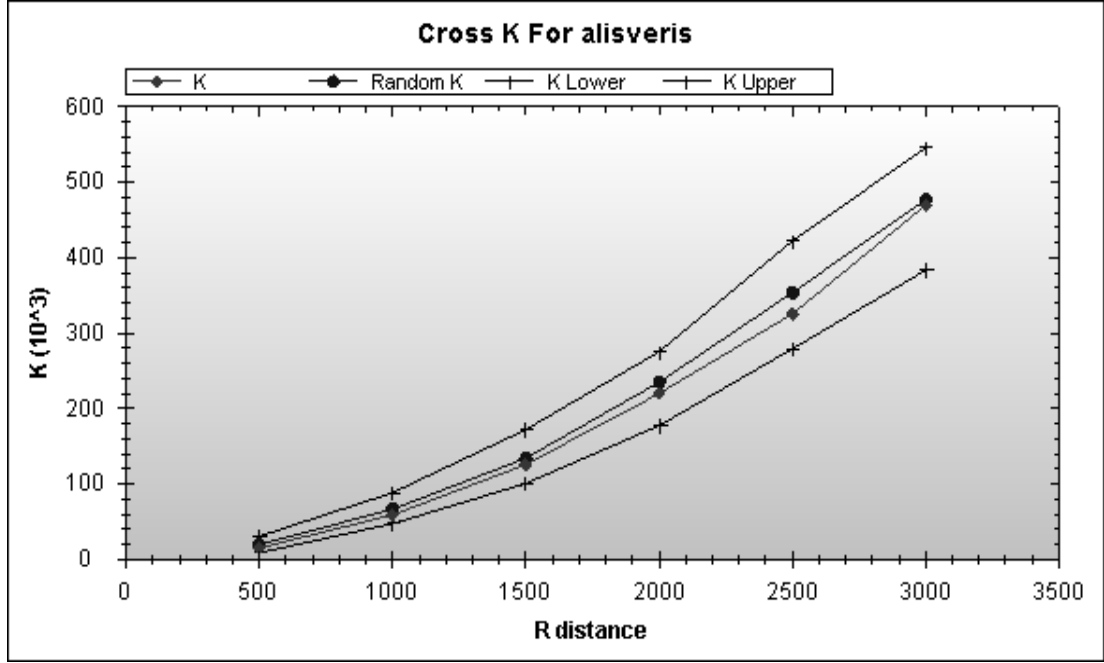


Şekil 1.1. 2006 Bahar dönemi alışveriş merkezleri çapraz K değerlerini gösterir grafik

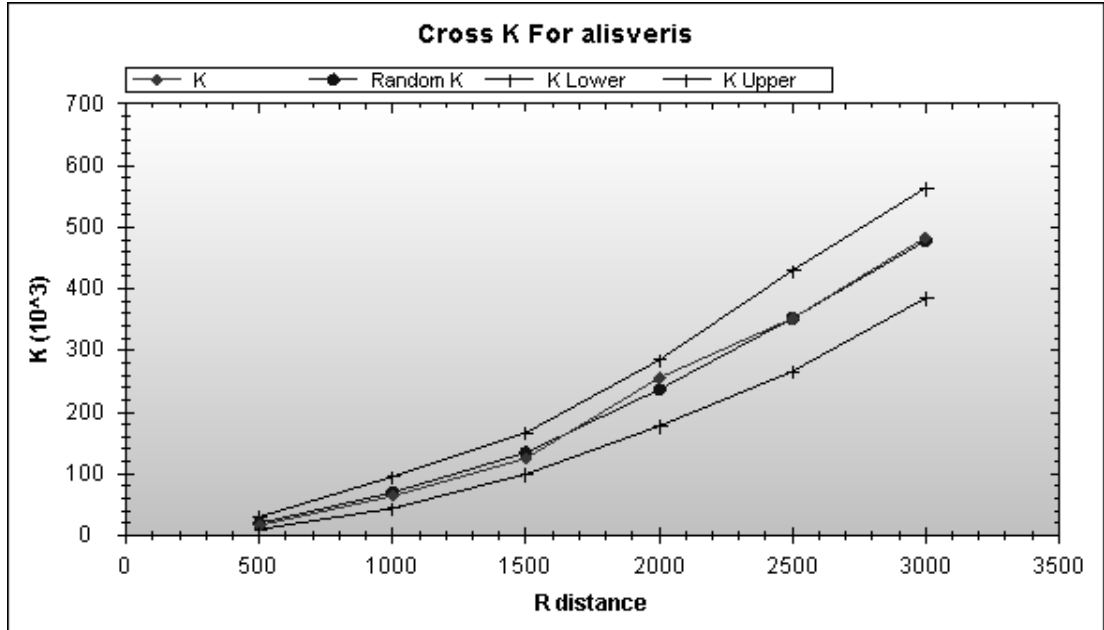


Şekil 1. 2. 2006 Yaz dönemi alışveriş merkezleri çapraz K değerlerini gösterir grafik

EK-1 (Devam) Dönemsel çapraz K grafikleri (alışveriş merkezleri)

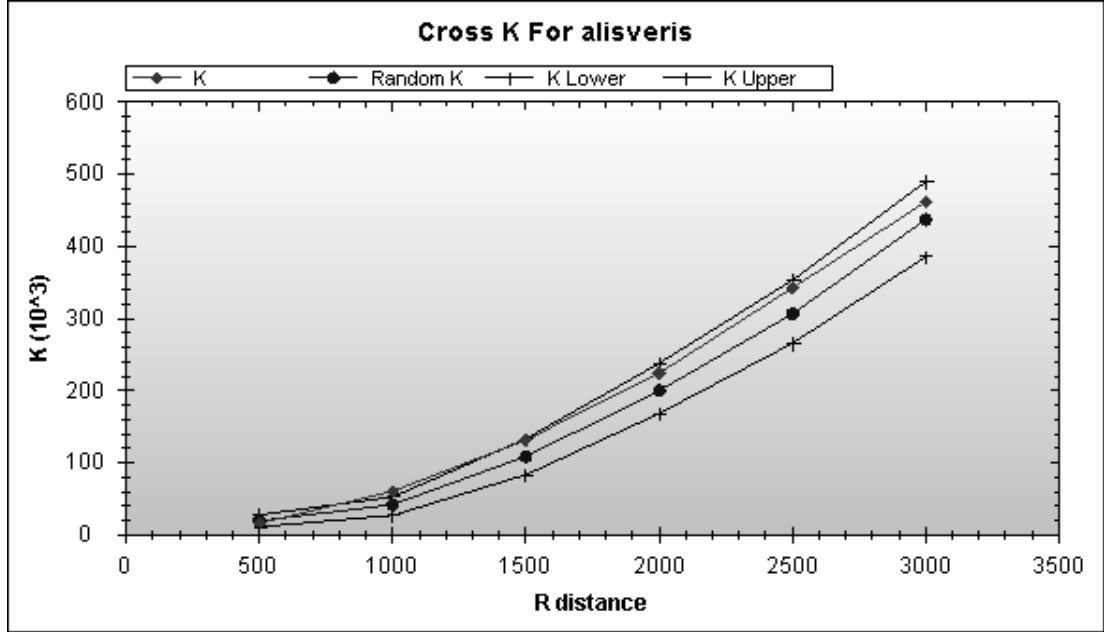


Şekil 1.3. 2006 Güz dönemi alışveriş merkezleri çapraz K değerlerini gösterir grafik

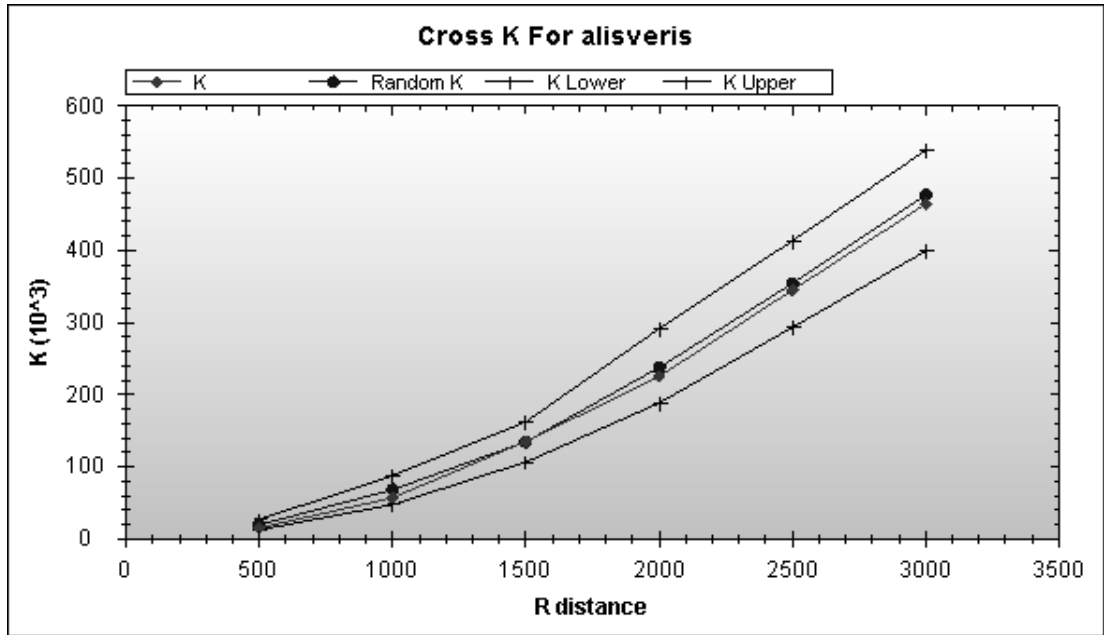


Şekil 1.4. 2007 Kış dönemi alışveriş merkezleri çapraz K değerlerini gösterir grafik

EK-1 (Devam) Dönemsel çapraz K grafikleri (alışveriş merkezleri)

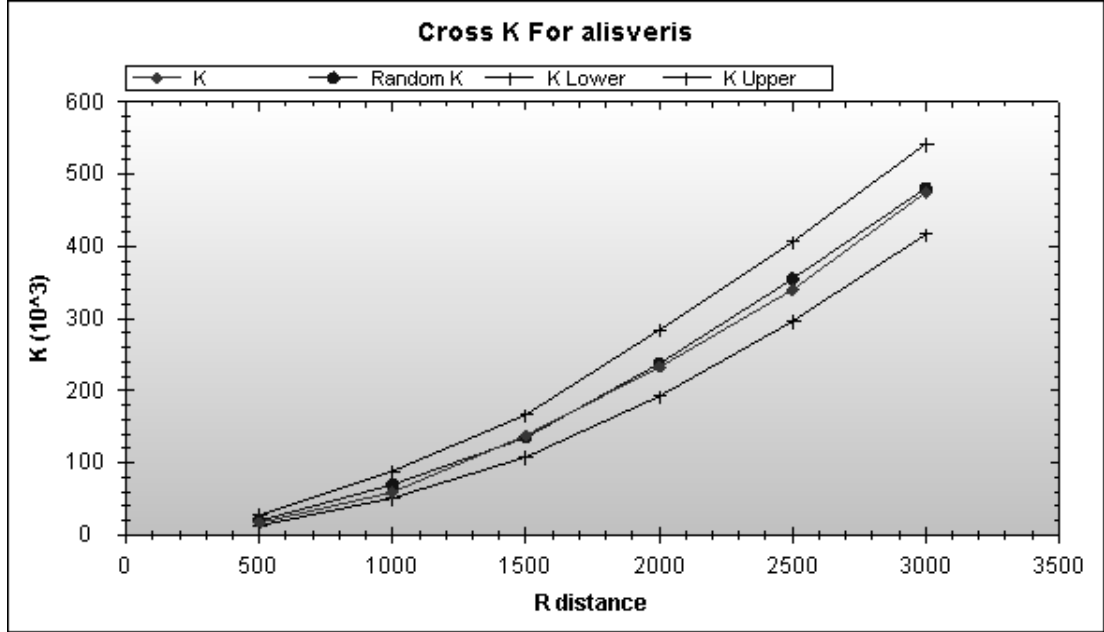


Şekil 1.5. 2007 Bahar dönemi alışveriş merkezleri çapraz K değerlerini gösterir grafik

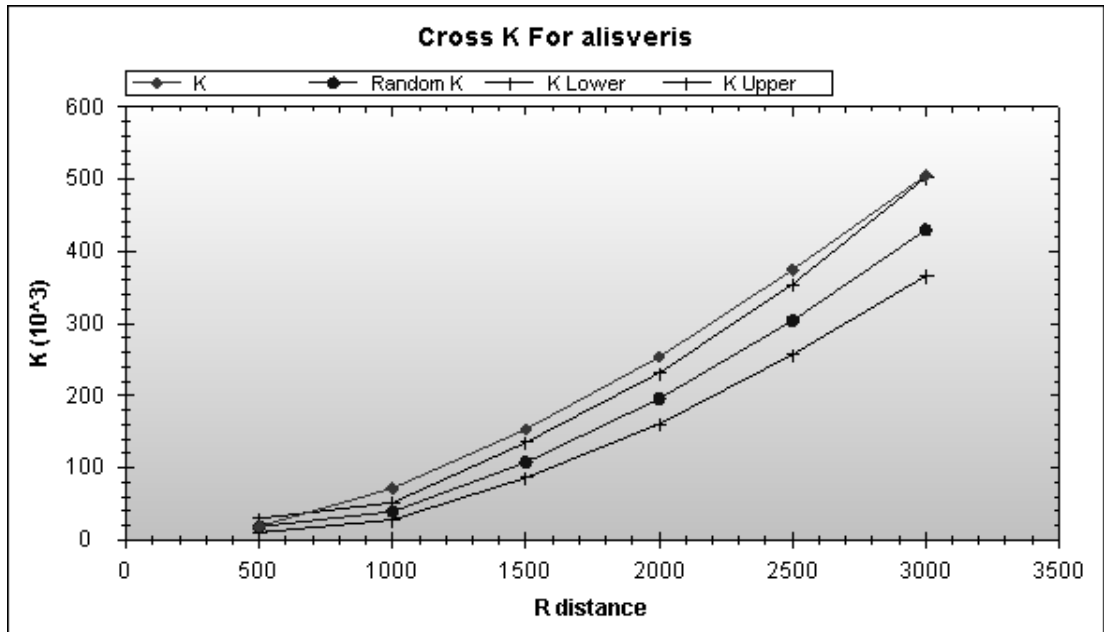


Şekil 1.6. 2007 Yaz dönemi alışveriş merkezleri çapraz K değerlerini gösterir grafik

EK-1 (Devam) Dönemsel çapraz K grafikleri (alışveriş merkezleri)

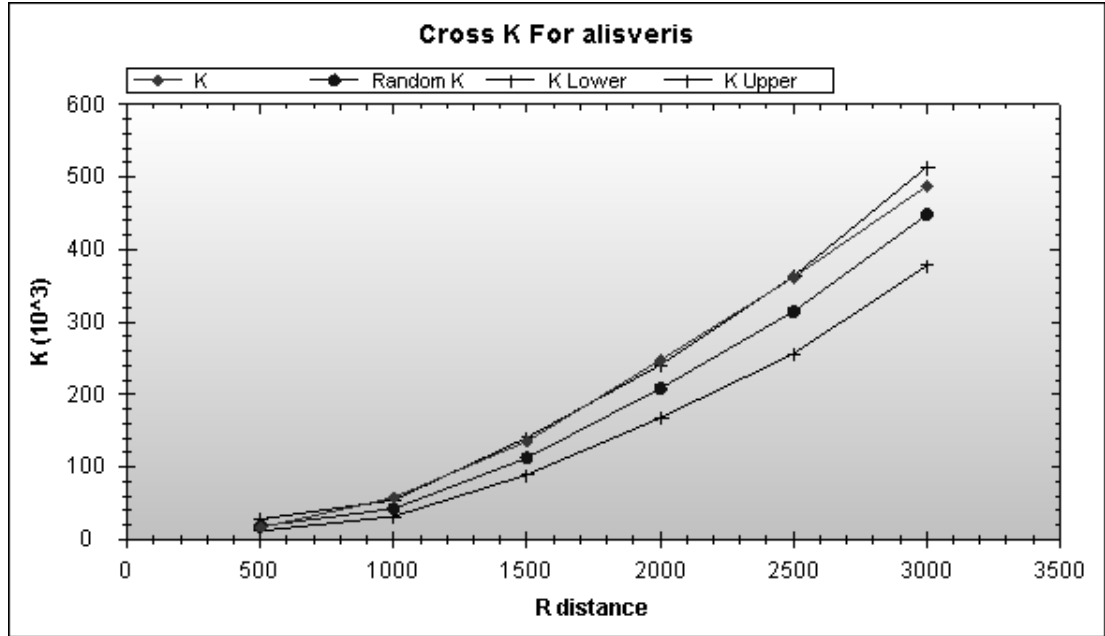


Şekil 1.7. 2007 Güz dönemi alışveriş merkezleri çapraz K değerlerini gösterir grafik

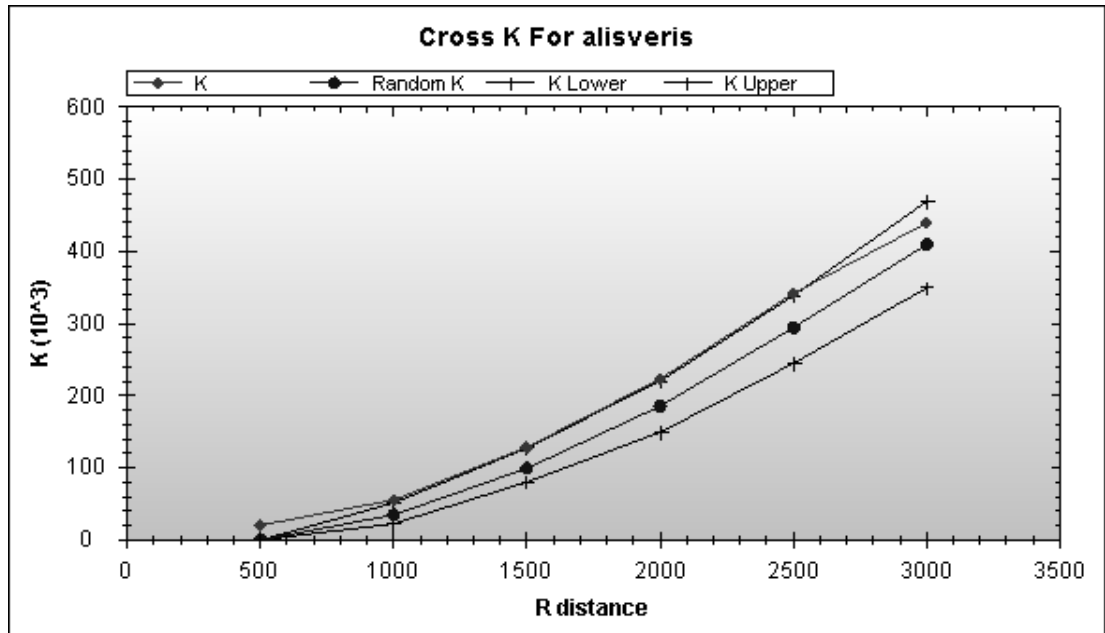


Şekil 1. 8. 2008 Kış dönemi alışveriş merkezleri çapraz K değerlerini gösterir grafik

EK-1 (Devam) Dönemsel çapraz K grafikleri (alışveriş merkezleri)

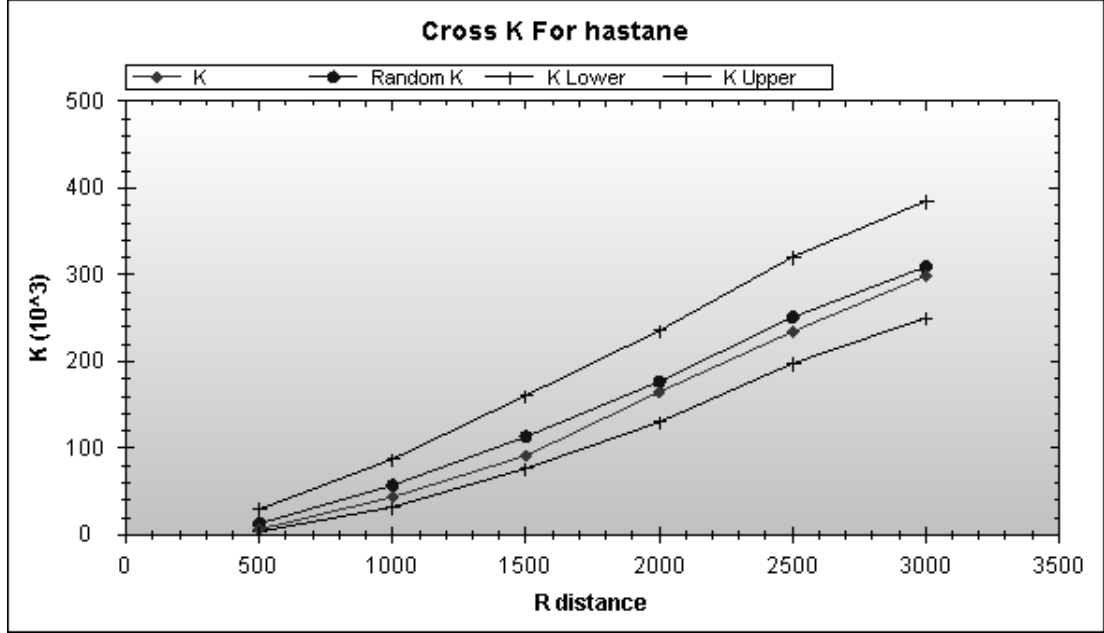


Şekil 1.9. 2008 Bahar dönemi alışveriş merkezleri çapraz K değerlerini gösterir grafik

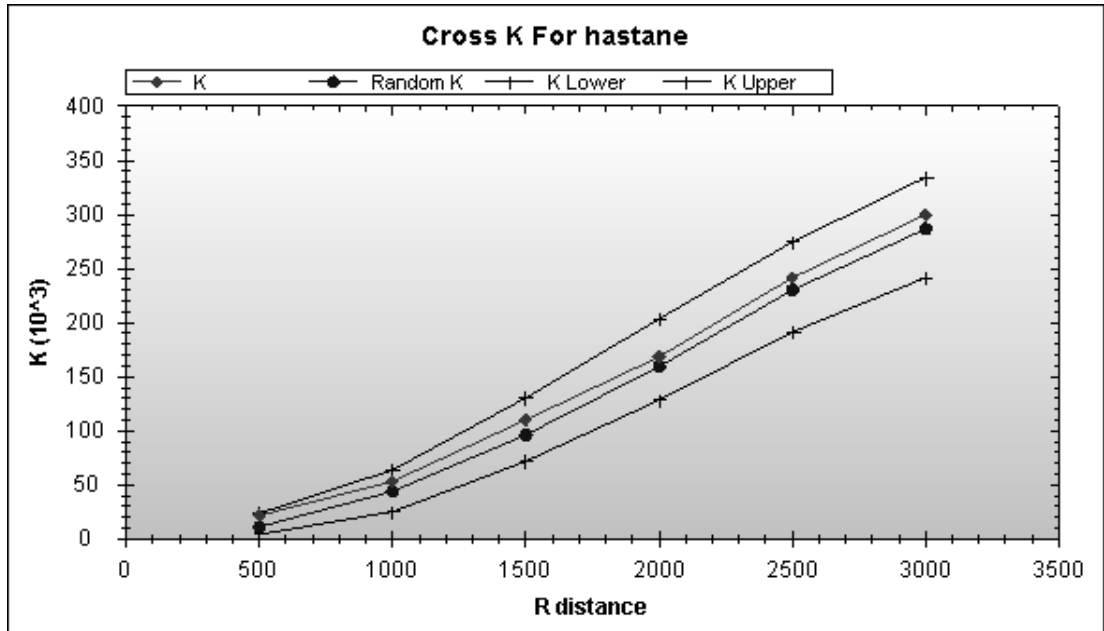


Şekil 1.10. 2008 Yaz dönemi alışveriş merkezleri çapraz K değerlerini gösterir grafik

EK-2 Dönemsel çapraz K grafikleri (sağlık merkezleri)

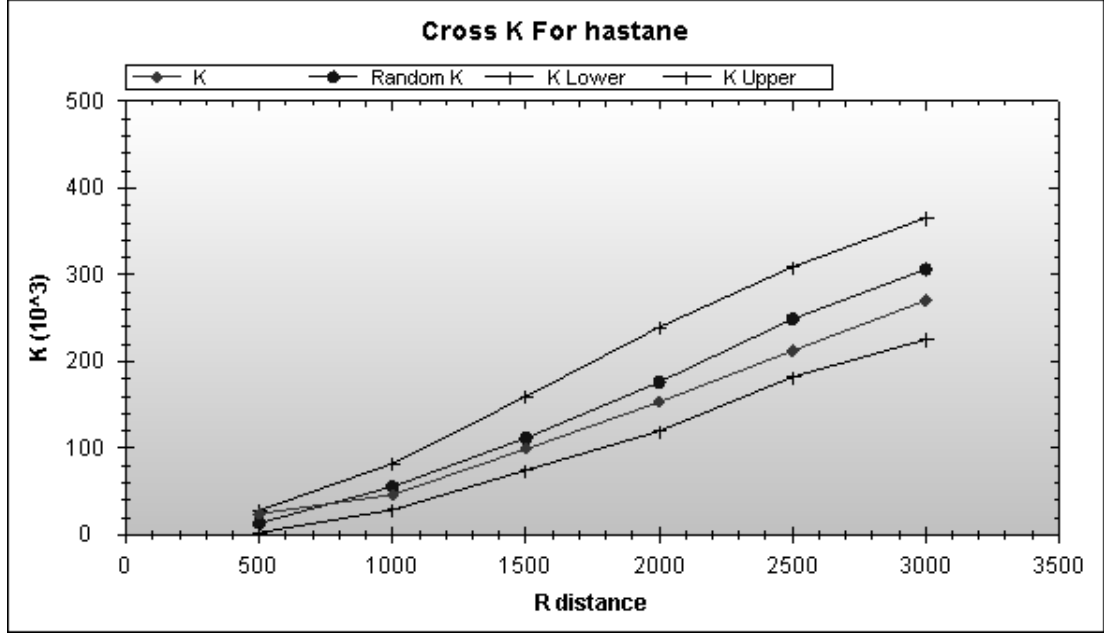


Şekil 2.1. 2006 Bahar dönemi sağlık merkezleri çapraz K değerlerini gösterir grafik

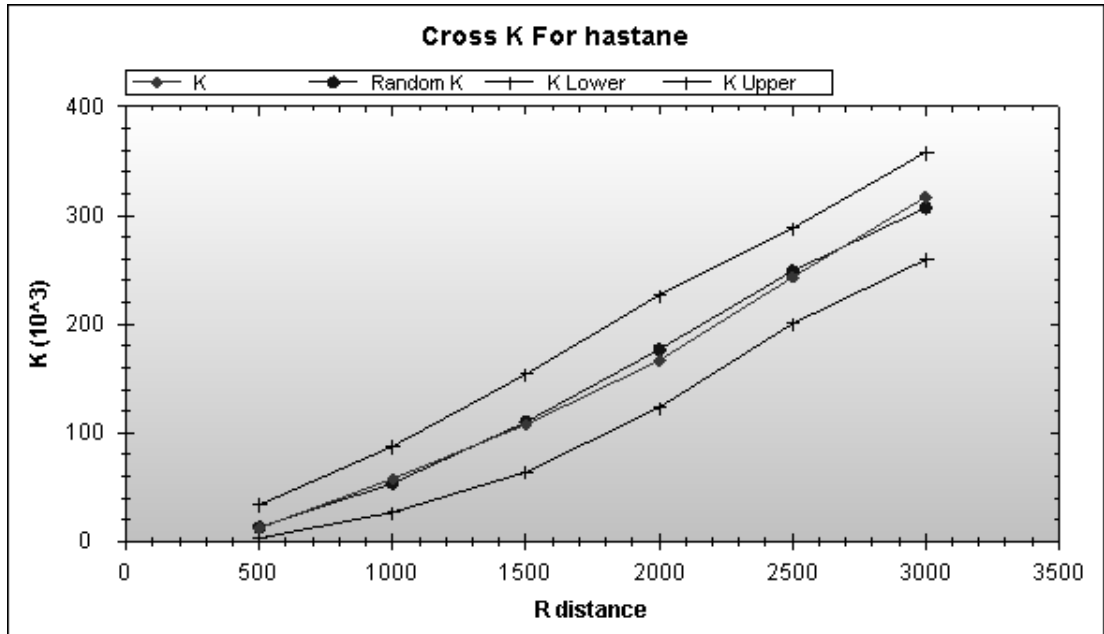


Şekil 2.2. 2006 Yaz dönemi sağlık merkezleri çapraz K değerlerini gösterir grafik

EK-2 (Devam) Dönemsel çapraz K grafikleri (sağlık merkezleri)

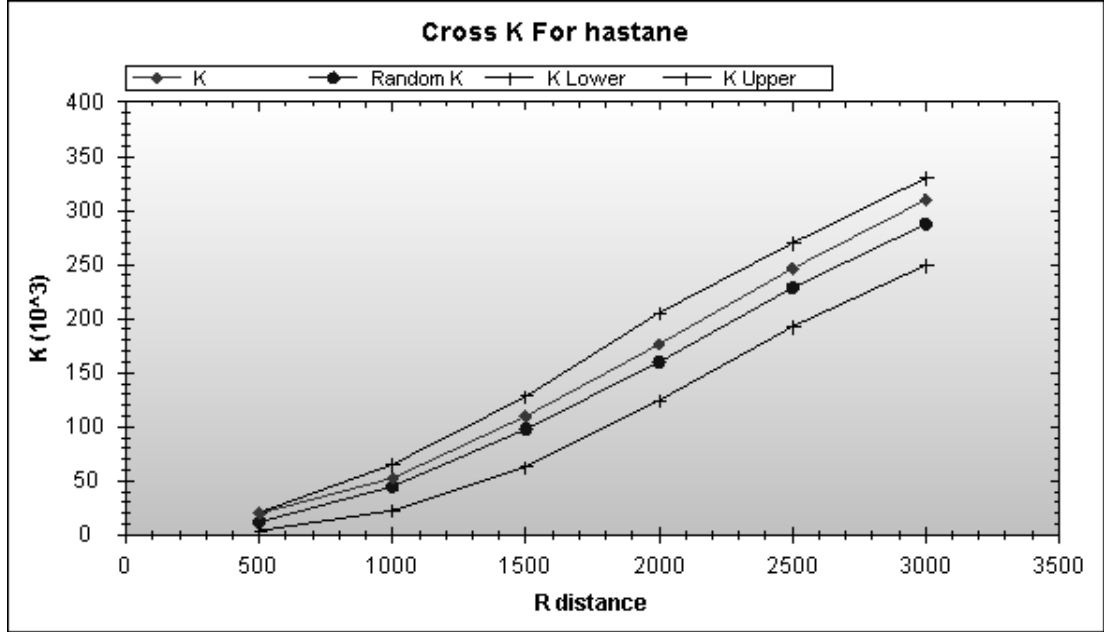


Şekil 2.3. 2006 Güz dönemi sağlık merkezleri çapraz K değerlerini gösterir grafik

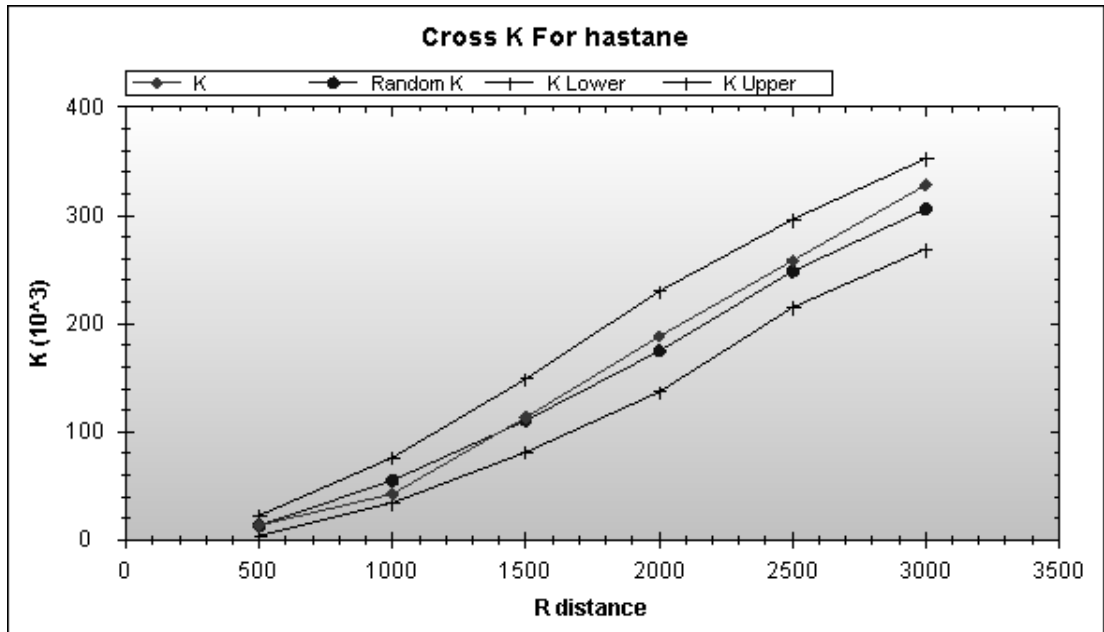


Şekil 2.4. 2007 Kış dönemi sağlık merkezleri çapraz K değerlerini gösterir grafik

EK-2 (Devam) Dönemsel çapraz K grafikleri (sağlık merkezleri)

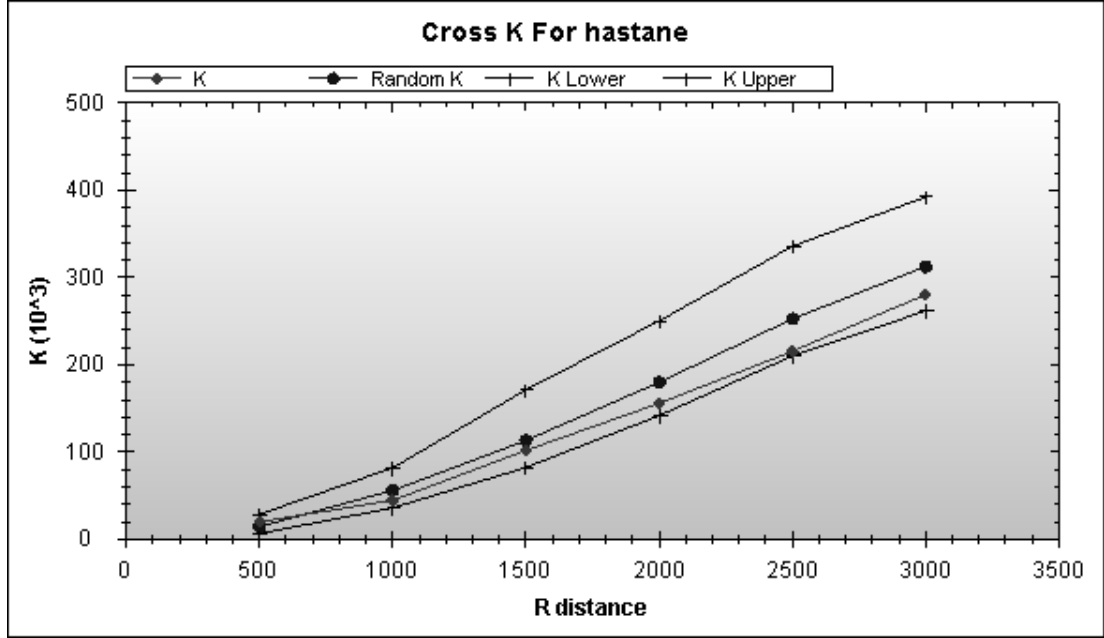


Şekil 2.5. 2007 Bahar dönemi sağlık merkezleri çapraz K değerlerini gösterir grafik

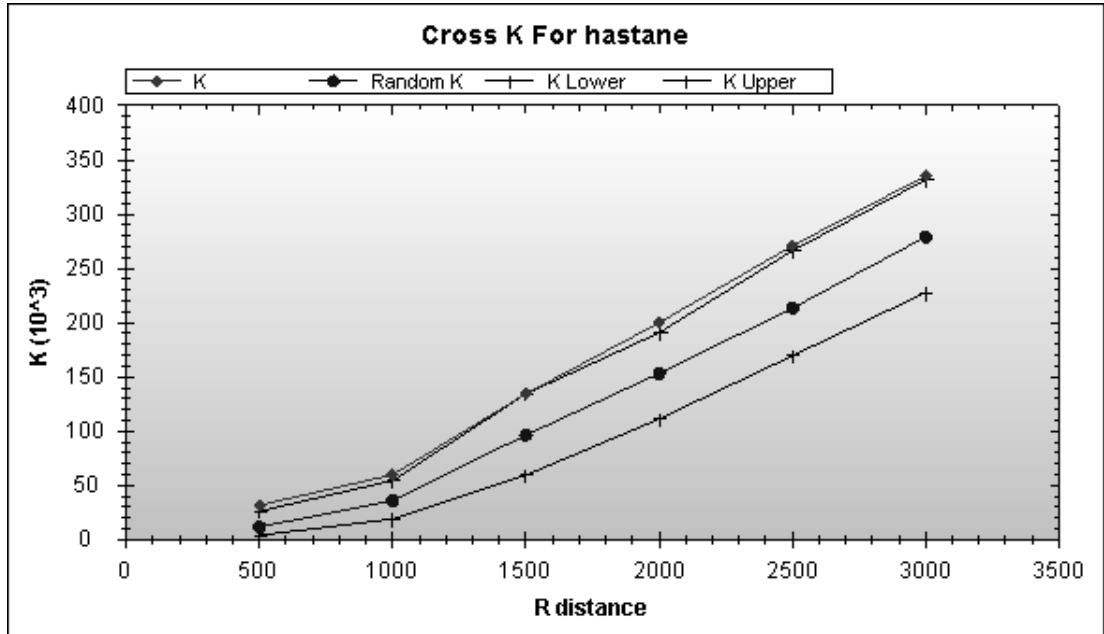


Şekil 2.6. 2007 Yaz dönemi sağlık merkezleri çapraz K değerlerini gösterir grafik

EK-2 (Devam) Dönemsel çapraz K grafikleri (sağlık merkezleri)

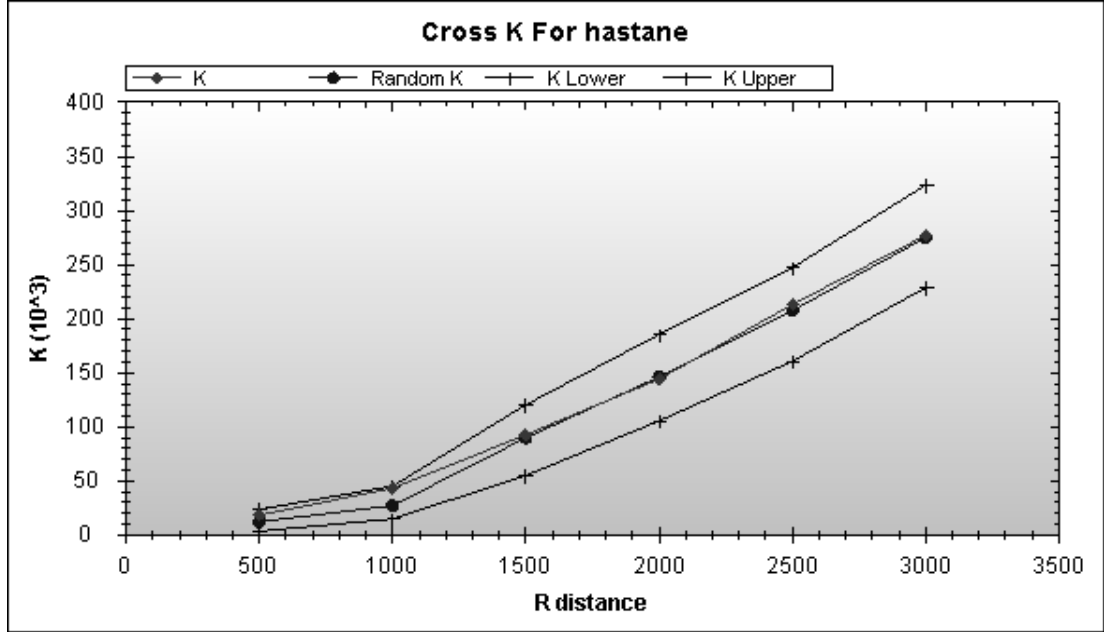


Şekil 2.7. 2007 Güz dönemi sağlık merkezleri çapraz K değerlerini gösterir grafik

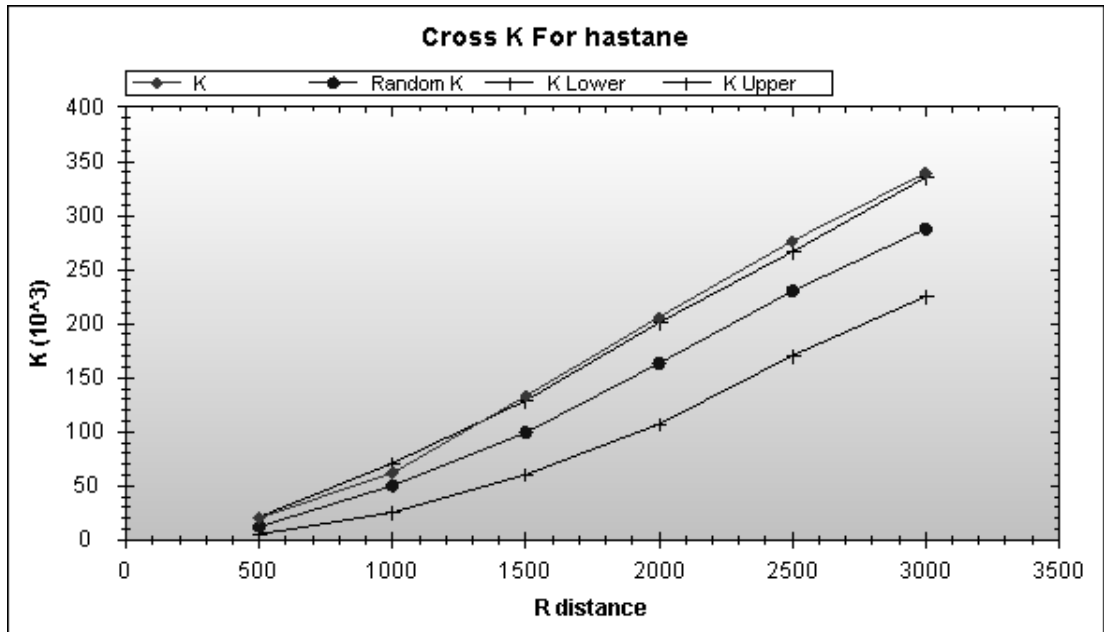


Şekil 2.8. 2008 Kış dönemi sağlık merkezleri çapraz K değerlerini gösterir grafik

EK-2 (Devam) Dönemsel çapraz K grafikleri (sağlık merkezleri)

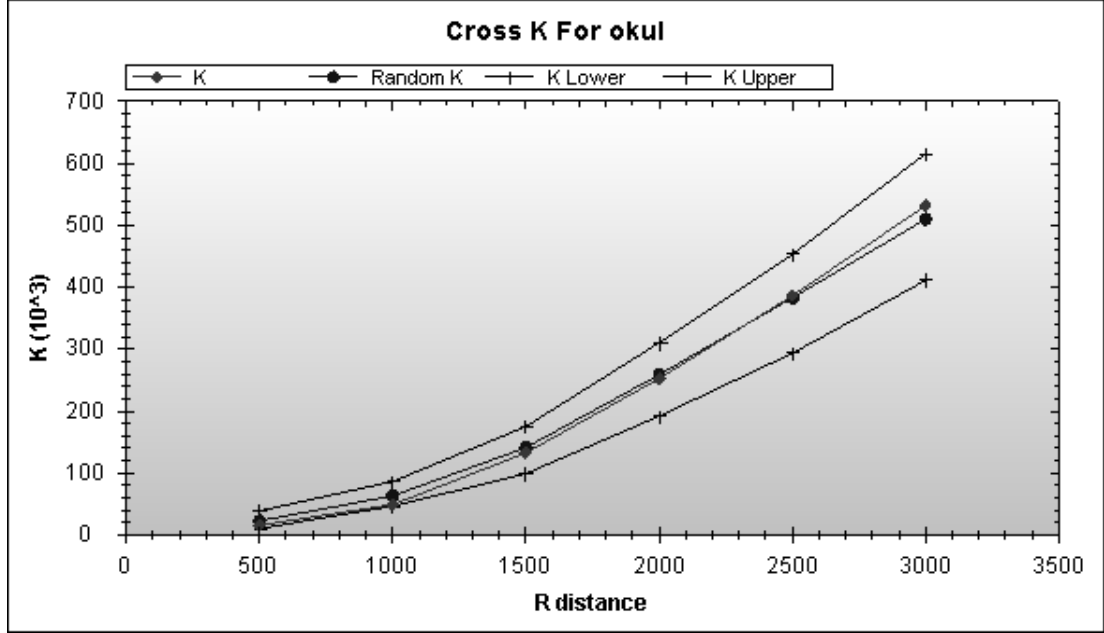


Şekil 2.9. 2008 Bahar dönemi sağlık merkezleri çapraz K değerlerini gösterir grafik

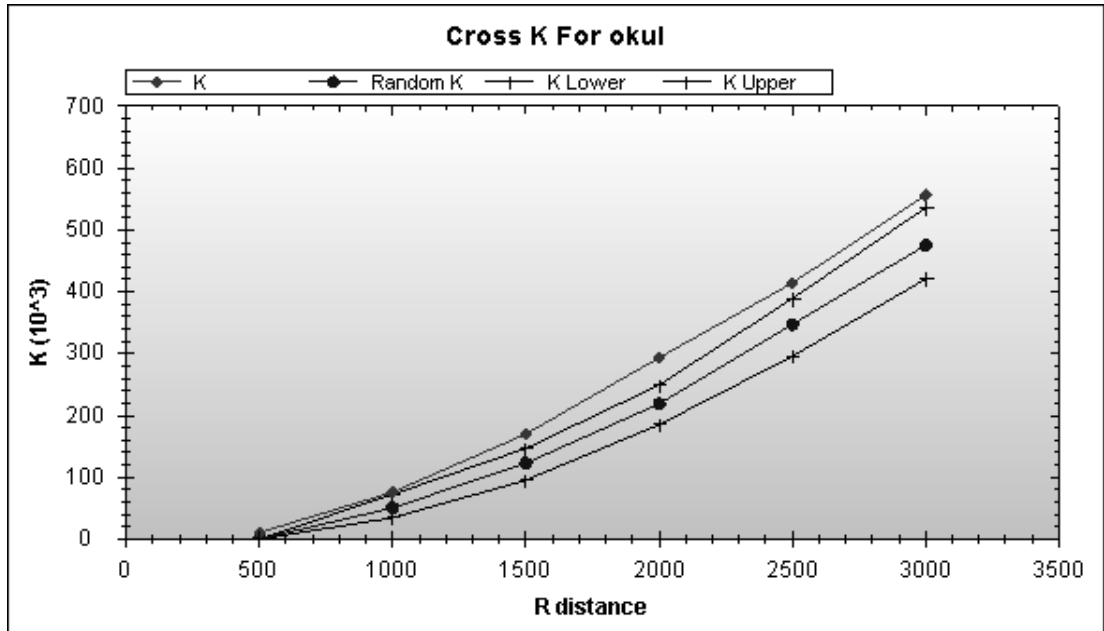


Şekil 2.10. 2008 Yaz dönemi sağlık merkezleri çapraz K değerlerini gösterir grafik

EK-3 Dönemsel çapraz K grafikleri (okullar)

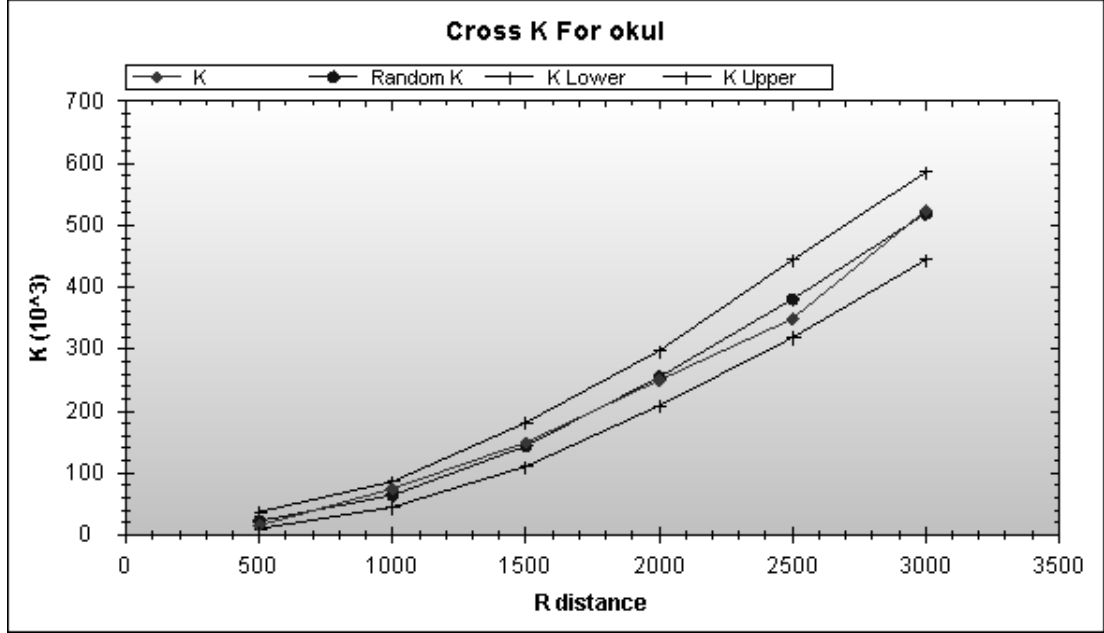


Şekil 3.1. 2006 Bahar dönemi okullar çapraz K değerlerini gösterir grafik

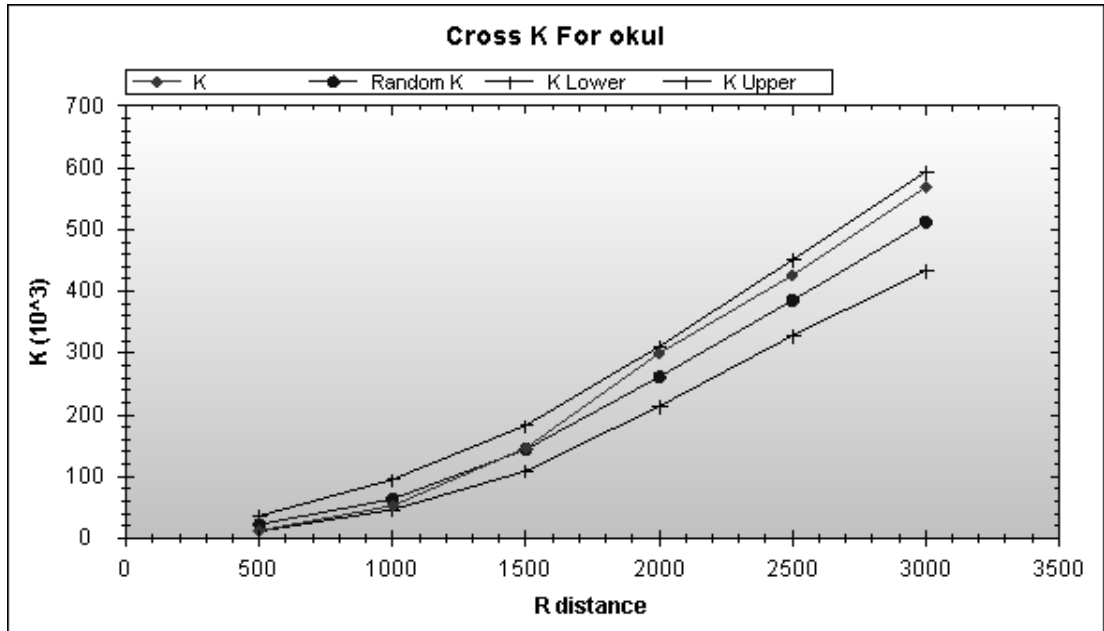


Şekil 3.2. 2006 Yaz dönemi okullar çapraz K değerlerini gösterir grafik

EK-3 (Devam) Dönemsel çapraz K grafikleri (okullar)

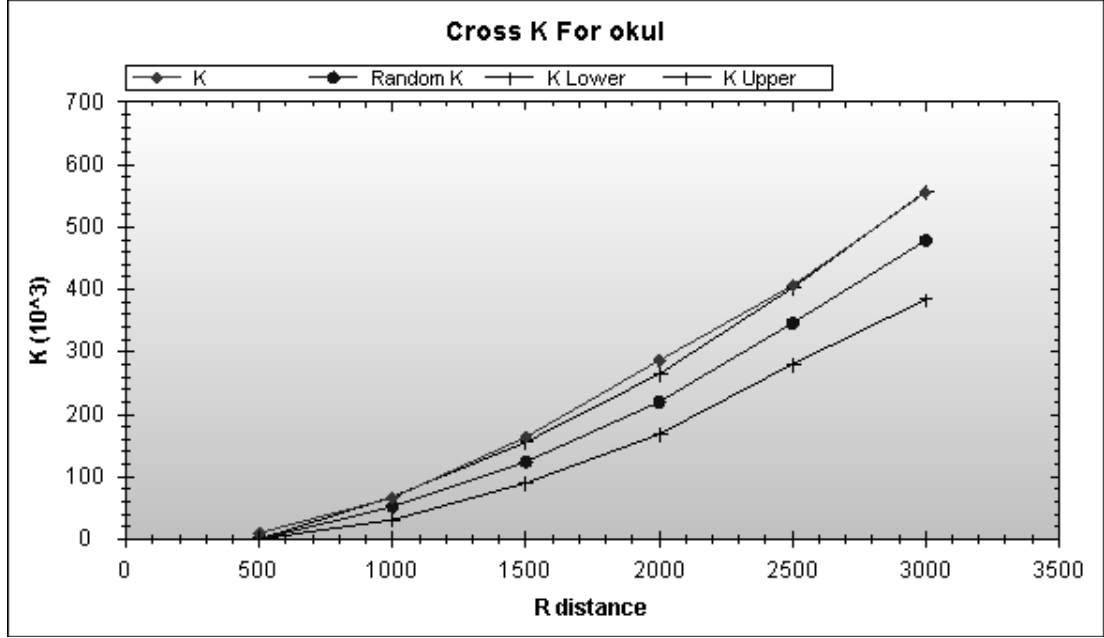


Şekil 3.3. 2006 Güz dönemi okullar çapraz K değerlerini gösterir grafik

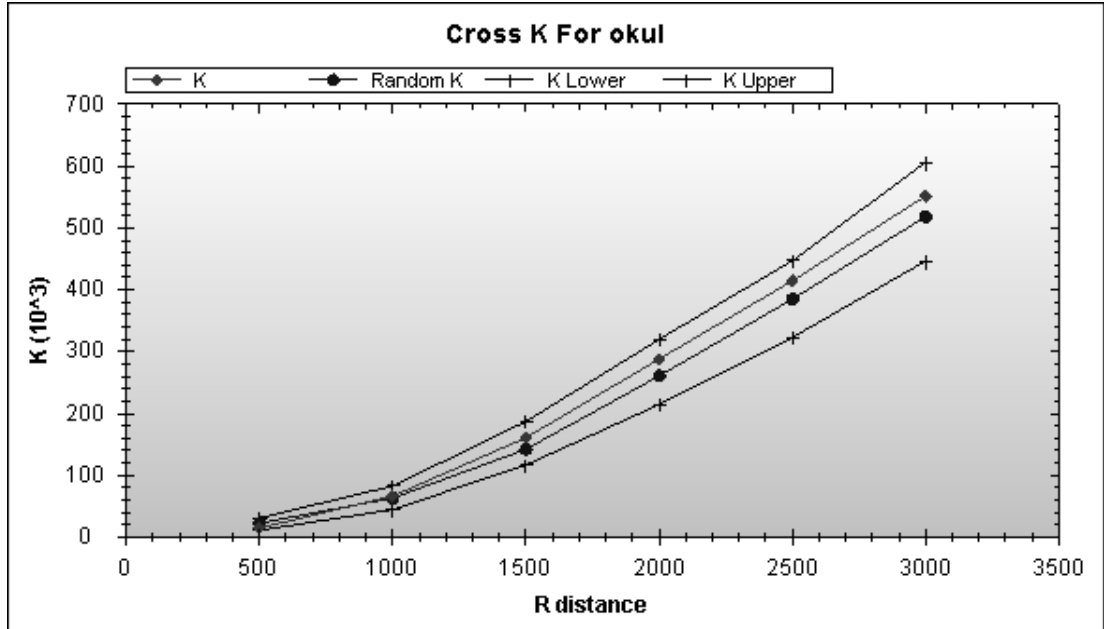


Şekil 3.4. 2007 Kış dönemi okullar çapraz K değerlerini gösterir grafik

EK-3 (Devam) Dönemsel çapraz K grafikleri (okullar)

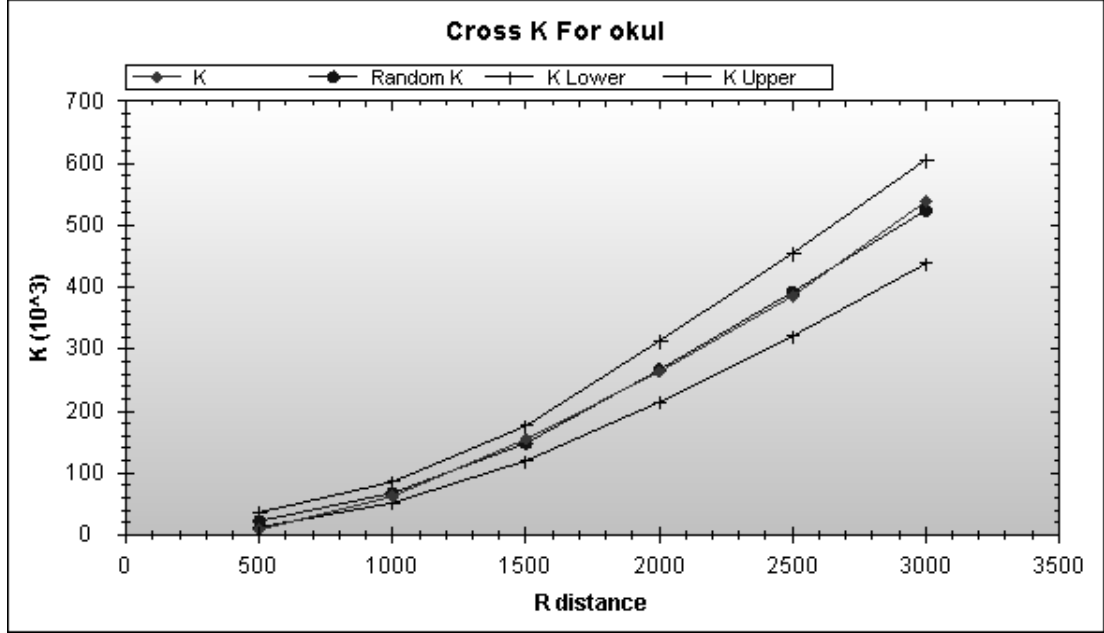


Şekil 3.5. 2007 Bahar dönemi okullar çapraz K değerlerini gösterir grafik

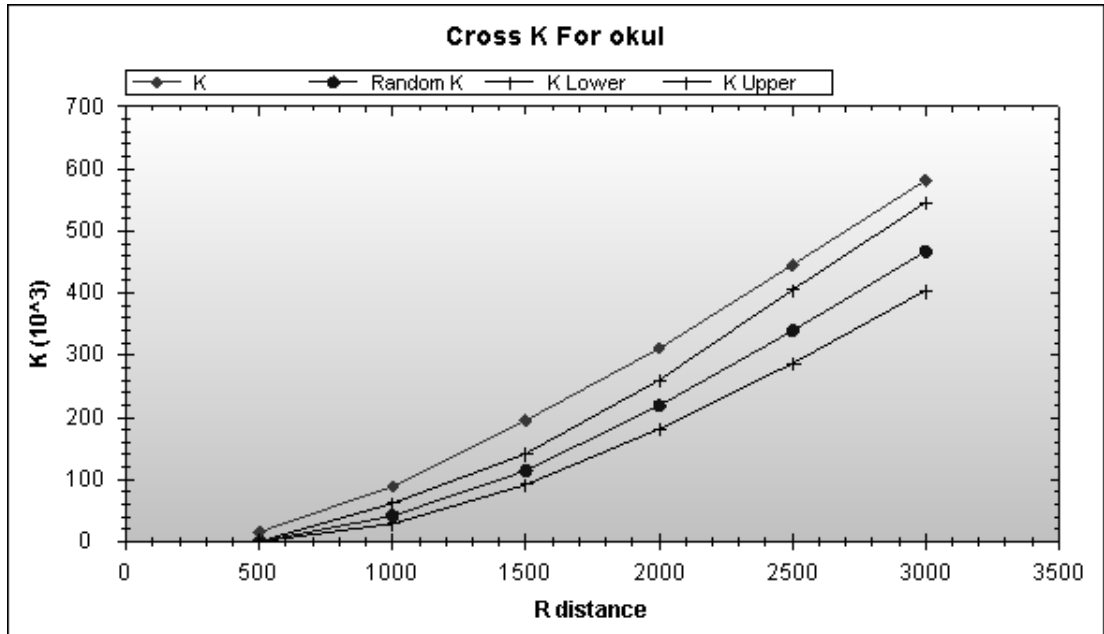


Şekil 3.6. 2007 Yaz dönemi okullar çapraz K değerlerini gösterir grafik

EK-3 (Devam) Dönemsel çapraz K grafikleri (okullar)

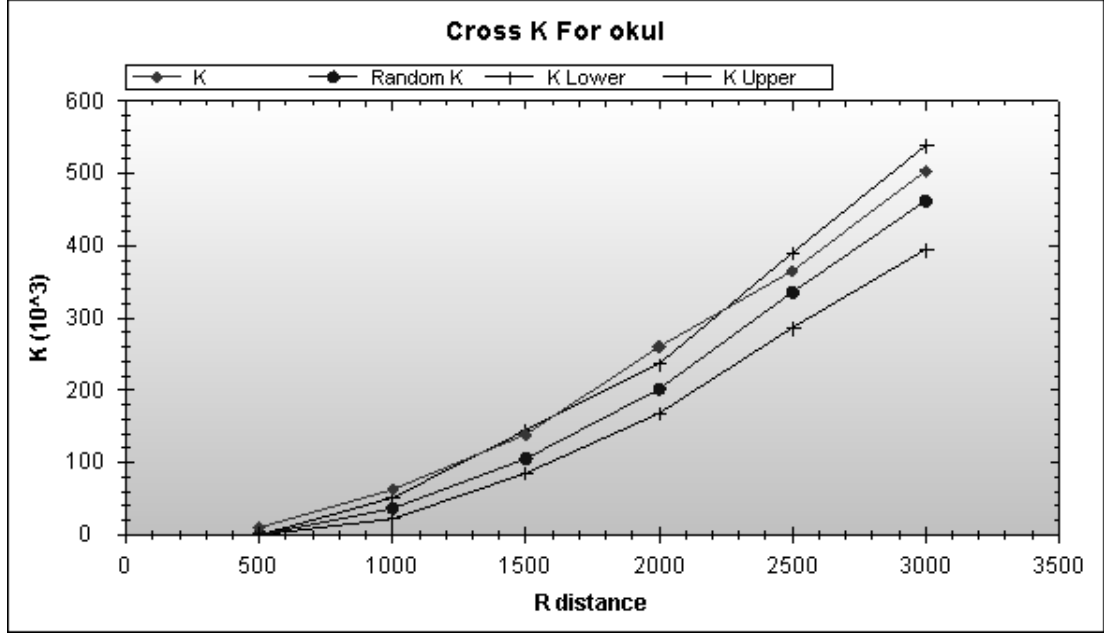


Şekil 3.7. 2007 Güz dönemi okullar çapraz K değerlerini gösterir grafik

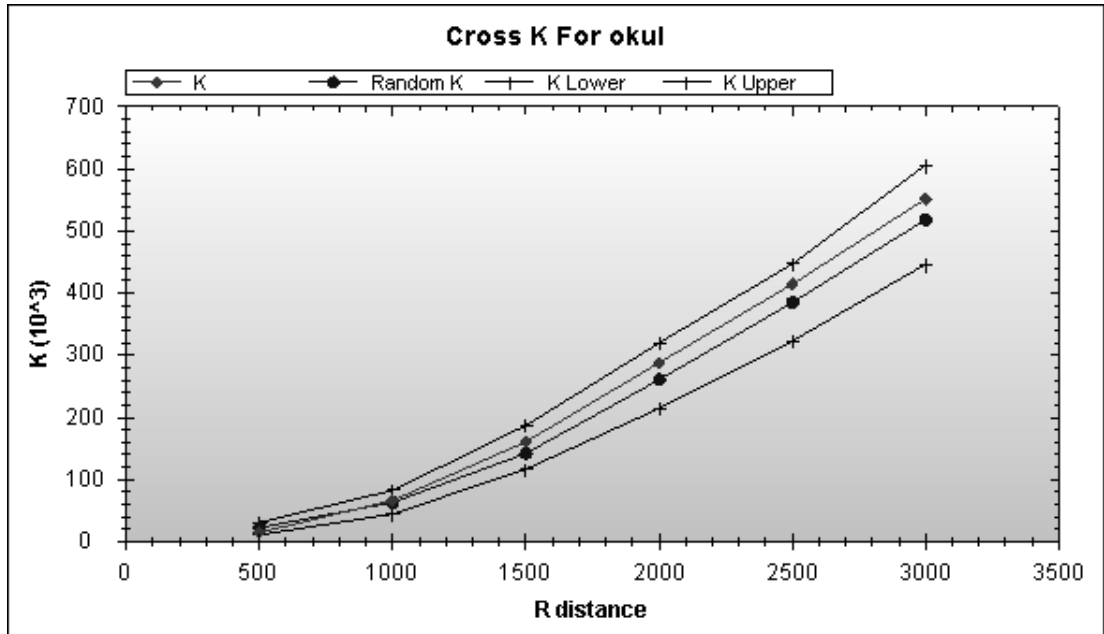


Şekil 3.8. 2008 Kış dönemi okullar çapraz K değerlerini gösterir grafik

EK-3 (Devam) Dönemsel çapraz K grafikleri (okullar)



Şekil 3.9. 2008 Bahar dönemi okullar çapraz K değerlerini gösterir grafik



Şekil 3.10. 2008 Yaz dönemi okullar çapraz K değerlerini gösterir grafik

EK-4 Geliştirilen yazılımında kullanılan veri tipleri

Çizelge 4.1. Türkiye’de ölümlü ve yaralanmalı trafik kazaları için kullanılan nümerik veri sınıfları

Kaza Unsuru	Veri Sınıfı
Araç	Araç hızı
Araç	Aşılan istiap haddi yolcu
Araç	Aşılan istiap haddi yük
Araç	Maddi hasar miktarı
Araç	Yük değeri
Kaza	Ölü sürücü sayısı
Kaza	Ölü yaya sayısı
Kaza	Ölü yolcu sayısı
Kaza	Yaralı sürücü sayısı
Kaza	Yaralı yaya sayısı
Kaza	Yaralı yolcu sayısı
Kaza	Yol mesafesi
Kazazede	Alkol testi sonucu
Kazazede	Yaşı
Sürücü	Alkol testi sonucu
Sürücü	Yaşı
Yol	Araç haricinde hasar gören diğer unsur mesafesi
Yol	Kaplama genişliği

Çizelge 4.2. Türkiye’de ölümlü ve yaralanmalı trafik kazaları için kullanılan nominal veri sınıfları

Kaza Unsuru	Veri Sınıfı	Değer
Araç	Cinsi	Bisiklet, at arabası, motorlu bisiklet, motosiklet, otomobil, minibüs, kamyonet, kamyon, çekici, otobüs, traktör, arazi taşıtı, özel veya diğer amaçlı taşıt, iş makinesi, ambulans, tanker, tren, tramvay
Araç	Firma adı	
Araç	Kask durumu	
Araç	Kazaya etken araç aksamı 1	Fren, rot, makas, şaft, şanzıman – vites, aks, direksiyon, far, diğer ışık, lastik patlaması, kapı, klakson, cam sileceği, diğer

EK-4 (Devam) Geliştirilen yazılımda kullanılan veri tipleri

Çizelge 4.2. (Devam) Türkiye’de ölümlü ve yaralanmalı trafik kazaları için kullanılan nominal veri sınıfları

Araç	Kazaya etken araç aksamı 2	Fren, rot, makas, şaft, şanzıman – vites, aks, direksiyon, far, diğer ışık, lastik patlaması, kapı, klakson, cam sileceği, diğer
Araç	Kullanım amacı	Özel, ticari, emniyet, askeri, diğer kamu kuruluşu, zirai, yabancı
Araç	Marka	
Araç	Model	
Araç	Plaka	
Araç	Poliçe no	
Araç	Sigorta tarihi	
Araç	Sigorta acente kodu	
Araç	Sigorta şirketi	
Araç	Yük cinsi	
Kaza	Adres	
Kaza	Araç sayısına göre kaza türü	Tek araçlı, iki araçlı (aynı yön), iki araçlı (zıt yön), iki araçlı (komşu yön), çok araçlı
Kaza	Dakika	
Kaza	Gün durumu	Gündüz, gece, alacakaranlık
Kaza	Haftanın günü	Pazartesi, salı, çarşamba, perşembe, cuma, cumartesi, pazar
Kaza	Hava durumu	Açık, bulutlu, sisli, yağmurlu, karlı, fırtınalı, tipili
Kaza	İl	
Kaza	İlçe	
Kaza	Kaza türü	Ölümlü, yaralanmalı
Kaza	Mahalle / köy	
Kaza	Oluşumuna göre kaza türü	Karşılıklı çarpışma, arkadan çarpma, yandan çarpma, duran araca çarpma, sabit cisme çarpma, yayaya çarpma, hayvana çarpma, devrilme, yoldan çıkma, araçtan düşen insan, araçtan düşen cisim
Kaza	Saat	
Kaza	Sorumluluk bölgesi	Polis, jandarma
Kaza	Tarih	

EK-4 (Devam) Geliştirilen yazılımda kullanılan veri tipleri

Çizelge 4.2. (Devam) Türkiye’de ölümlü ve yaralanmalı trafik kazaları için kullanılan nominal veri sınıfları

Kaza	Yeri	Cadde, sokak, otoyol, devlet yolu, il yolu, köy yolu, orman yolu, servis yolu, bağlantı yolu, park alanı, tesis önü veya içi
Kaza	Yeri	Yerleşim yeri, yerleşim yeri dışı
Kaza	Yol adı	
Kaza	Yol tipi	Bölünmüş, bölünmemiş
Kazazede	Adı soyadı	
Kazazede	Alkol testi durumu	Kontrol edildi, kontrol edilemedi
Kazazede	Bulunduğu araç	
Kazazede	Cinsiyeti	Erkek, kadın
Kazazede	Emniyet kemeri durumu	Takılı, takılı değil, zorunlu değil, belirsiz
Kazazede	İlk yardım yapan	Sağlık ekibi, polis, jandarma, vatandaş
Kazazede	Kaza sonucu	Ölü, yaralı
Kazazede	Kusur 1.	Yolcu için: Taşıttan sarkmak; habersiz inmek veya binmek; taşıt içinde gayri nizami hareket etmek; taşıt dışında seyahat etmek; hareket halindeki taşıttan yere atlamak; açık yük üzerinde gayri nizami ve tehlikeli biçimde seyahat etmek; diğer; Yaya için: Kırmızı ışıklı trafik işaretinde veya yetkili memurun dur işaretinde geçmek; yola birden bire çıkmak; duran aracın önünden arkasından çıkmak; araçlara ilk geçiş hakkını vermemek; yol içinde koşmak, yürümek, oynamak, oturmak; hareket halindeki taşıta asılmak; otoyola girmek; yol içerisinde hatalı şekilde elle sürülen araç kullanmak; yol içerisinde hatalı şekilde hayvan sev ve idare etmek; görünürlüğü arttırıcı tedbirleri almamak; diğer

EK-4 (Devam) Geliştirilen yazılım kullanılan veri tipleri

Çizelge 4.2. (Devam) Türkiye’de ölümlü ve yaralanmalı trafik kazaları için kullanılan nominal veri sınıfları

Kazazede	Kusur 2.	Yolcu için: Taşıttan sarkmak; habersiz inmek veya binmek; taşıt içinde gayri nizami hareket etmek; taşıt dışında seyahat etmek; hareket halindeki taşıttan yere atlamak; açık yük üzerinde gayri nizami ve tehlikeli biçimde seyahat etmek; diğer; Yaya için: Kırmızı ışıklı trafik işaretinde veya yetkili memurun dur işaretinde geçmek; yola birden bire çıkmak; duran aracın önünden arkasından çıkmak; araçlara ilk geçiş hakkını vermemek; yol içinde koşmak, yürümek, oynamak, oturmak; hareket halindeki taşıta asılmak; otoyola girmek; yol içerisinde hatalı şekilde elle sürülen araç kullanmak; yol içerisinde hatalı şekilde hayvan sev ve idare etmek; görünürlüğü artırıcı tedbirleri almamak; diğer
Kazazede	Tabiiyeti	TC, yabancı, diğer
Kazazede	Yaya ise giysi özelliği	Açık, koyu, reflektif malzeme
Kazazede	Yol kullanım durumu	Yolcu, yaya
Sürücü	Adı soyadı	
Sürücü	Adres	
Sürücü	Alkol testi durumu	Kontrol edildi, kontrol edilemedi

EK-4 (Devam) Geliştirilen yazılım kullanılan veri tipleri

Çizelge 4.2. (Devam) Türkiye’de ölümlü ve yaralanmalı trafik kazaları için kullanılan nominal veri sınıfları

Sürücü	Asli kusur 1	Kırmızı ışıklı trafik işaretine veya yetkili memurun dur işaretinde geçme; “taşıt giremez” trafik işaretinin bulunduğu karayoluna veya bölünmüş karayolunda karşı yönden gelen trafiğin kullandığı şerit, rampa, ve bağlantı yollarına girme; ikiden fazla şeritli taşıt yollarında, karşı yoldan gelen trafiğin kullandığı şerit veya yol bölümüne girme; arkadan çarpma; geçme yasağı olan yerlerde geçme; doğrultu değiştirme manevralarını yanlış yapma; şeride tecavüz etme; kavşaklarda geçiş önceliğine uymama; kaplamanın dar olduğu yerlerde geçiş önceliğine uymama; manevraları düzenleyen genel şartlara uymama; yerleşim birimleri dışındaki karayolunun taşıt yolu üzerinde, zorunlu haller dışında park etme veya duraklama ve her durumda gerekli tedbiri almama; park için ayrılmış yerlerde veya taşıt yolu dışında kurallara uygun olarak park etmiş araçlara çarpma
--------	--------------	---

EK-4 (Devam) Geliştirilen yazılım kullanılan veri tipleri

Çizelge 4.2. (Devam) Türkiye’de ölümlü ve yaralanmalı trafik kazaları için kullanılan nominal veri sınıfları

Sürücü	Asli kusur 2	Kırmızı ışıklı trafik işaretine veya yetkili memurun dur işaretinde geçme; “taşıt giremez” trafik işaretinin bulunduğu karayoluna veya bölünmüş karayolunda karşı yönden gelen trafiğin kullandığı şerit, rampa ve bağlantı yollarına girme; ikiden fazla şeritli taşıt yollarında, karşı yoldan gelen trafiğin kullandığı şerit veya yol bölümüne girme; arkadan çarpma; geçme yasağı olan yerlerde geçme; doğrultu değiştirme manevralarını yanlış yapma; şeride tecavüz etme; kavşaklarda geçiş önceliğine uymama; kaplamanın dar olduğu yerlerde geçiş önceliğine uymama; manevraları düzenleyen genel şartlara uymama; yerleşim birimleri dışındaki karayolunun taşıt yolu üzerinde, zorunlu haller dışında park etme veya duraklama ve her durumda gerekli tedbiri almama; park için ayrılmış yerlerde veya taşıt yolu dışında kurallara uygun olarak park etmiş araçlara çarpma
Sürücü	Cinsiyeti	Erkek, kadın
Sürücü	Diğer kanuni kusur 1	2918 sayılı karayolları trafik kanununda yer alan ve kazaya sebebiyet verdiği belirlenen kural ihlali
Sürücü	Diğer kanuni kusur 2	2918 sayılı karayolları trafik kanununda yer alan ve kazaya sebebiyet verdiği belirlenen kural ihlali
Sürücü	Emniyet kemeri durumu	Takılı, takılı değil, zorunlu değil, belirsiz
Sürücü	İlk yardım yapan	Sağlık ekibi, polis, jandarma, vatandaş

EK-4 (Devam) Geliştirilen yazılım kullanılan veri tipleri

Çizelge 4.2. (Devam) Türkiye’de ölümlü ve yaralanmalı trafik kazaları için kullanılan nominal veri sınıfları

Sürücü	Kaza sonucu	Ölü, yaralı, sağlam
Sürücü	Öğrenim durumu	İlk, orta, ilköğretim, lise, yüksek
Sürücü	Sürücü belge no	
Sürücü	Sürücü belgesi durumu	Var, yok, tespit edilemedi
Sürücü	Sürücü belgesi sınıfı	
Sürücü	Sürücü belgesi veren birim	Emniyet, özel, askeri, yabancı
Sürücü	Sürücü belgesinin verildiği il	
Sürücü	Sürücü belgesinin verildiği ilçe	
Sürücü	Sürücü belgesinin verildiği yıl	
Sürücü	Sürücü okulu kodu	
Sürücü	Tabiiyeti	TC, yabancı, diğer
Sürücü	Telefon	
Sürücü	Uygulanan ceza	
Sürücü	Uygulanan ceza tutanağı seri no	
Sürücü	Uyuşturucu şüphesi	Var, yok
Tutanak	Hazırlayanın adı soyadı	
Tutanak	Hazırlayanın rütbesi	
Tutanak	Hazırlayanın sicil no	
Tutanak	Tutanağı düzenleyen birimin adı	
Tutanak	Tutanağı düzenleyen birimin telefonu	
Tutanak	Tutanak defter sıra no	
Yol	Araç haricinde hasar gören diğer unsur adı	
Yol	Aydınlatma durumu	Var, yok, bozuk
Yol	Banket durumu	Var, yok, bozuk
Yol	Diğer yol özellikleri	Dar yol, dar köprü, köprü üstü, menfez üstü, kasis, tünel içi, hiçbirisi
Yol	Düşey güzergah	Eğimsiz, hafif eğimli, dik eğimli, tepe üstü
Yol	Geçit	Kontrollü demiryolu geçidi, kontrolsüz demiryolu geçidi, okul geçidi, yaya geçidi, geçit yok

EK-4 (Devam) Geliştirilen yazılım kullanılan veri tipleri

Çizelge 4.2. (Devam) Türkiye’de ölümlü ve yaralanmalı trafik kazaları için kullanılan nominal veri sınıfları

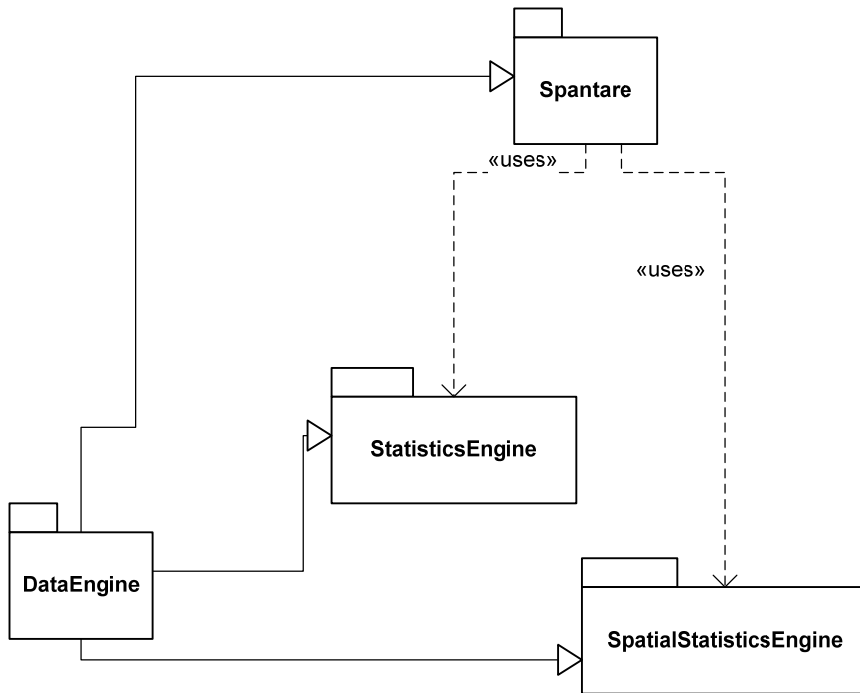
Yol	Görüşe engel cisim durumu	Var, yok, bozuk
Yol	Kaplama cinsi	Beton, asfalt, parke, stabilize, ham yol
Yol	Kavşak	Üç yönlü t, üç yönlü y, dört yönlü, beş veya daha fazla yönlü, dönel, diğer kavşak çeşitleri, kavşak yok
Yol	Trafik görevlisi durumu	Var, yok, bozuk
Yol	Trafik işaret levhası durumu	Var, yok, bozuk
Yol	Trafik lambası durumu	Var, yok, bozuk
Yol	Yatay güzergah	Düz yol, hafif viraj, korkuluklu sert viraj, korkuluksuz sert viraj
Yol	Yaya kaldırımı durumu	Var, yok, bozuk
Yol	Yol şerit çizgisi durumu	Var, yok, bozuk
Yol	Yolda çalışma durumu	Var, yok, bozuk
Yol	Yolda soruna ait uyarıcı işaretleme var mı?	Var, yok
Yol	Yoldan kaynaklanan 1. Sorun	Köprü çökmesi, tekerlek izine oturma, heyelandan dolayı şerit çökmesi, kısmi veya münferit çökme, düşük banket, yol sathında gevşek malzeme, yolda münferit çukur
Yol	Yoldan kaynaklanan 2. Sorun	Köprü çökmesi, tekerlek izine oturma, heyelandan dolayı şerit çökmesi, kısmi veya münferit çökme, düşük banket, yol sathında gevşek malzeme, yolda münferit çukur
Yol	Yön	Tek, iki
Yol	Yüzey durumu	Kuru, ıslak, çamurlu, karlı, buzlu, tozlu, yağ veya akaryakıt, su birikintili

EK-5 Geliştirilen yazılımın mimarisi

Bu çalışma kapsamında üretilen ve trafik kazalarının mekansal analizinde kullanılacak olan uygulama yazılımı Span ismi ile adlandırılmıştır.

Uygulama yazılımı nesne yönelimli bir mimari ile geliştirilmiştir. Span tek katmanlı yazılım mimarisinde ve Microsoft Framework 3.0 C# dili kullanılarak, Windows İşletim sistemleri için nesne yönelimli bir masaüstü uygulaması şeklinde üretilmiştir.

Uygulama yazılımı gereksinimleri modülerliği sağlamak amacı ile dört ayrı paket içerisinde yer alan fonksiyonlar ile karşılanmıştır. İşlevsel fonksiyonların tamamı bu çalışmada üretilmiştir. Gösterimde ve etkileşimde kullanılan harita ve grafik bileşenleri ise açık kaynak kodlu ve uygun lisanslanmış başka bir kod parçasıdır.



Şekil 5.1. Span uygulama yazılımının sistem mimarisi

EK-5 (Devam) Geliştirilen yazılımın mimarisi

Span uygulama yazılımı modüllerinin işlevsel tanımları şunlardır:

DataEngine modülü: Verinin; tanımlama, saklama, yükleme fonksiyonlarını yerine getirir. Uygulama yazılımında kullanıcı tarafından bilinen tüm veri tipleri bu paket tarafından tutulur.

DataEngine modülü içerisinde veri tipleri şu sınıflarla yer alır:

SpatialPoint: Koordinat şeklinde tanımlan ve tekil bir numara ile takip edilebilen mekansal nokta tanımıdır. Düzlem üzerinde bir noktayı belirlemek üzere enlem ve boylam koordinatları kullanılır. SpatialPoint sınıfı bir sıra numarasına, enlem ve boylam tanımlamalarına sahiptir. Nesnenin enlem ve boylam bilgisi WGS 84 sisteminde ondalık derece ile gösterilir. Uygulama yazılımı kazayı, kazaların dağılımı üzerinde etkisi kontrol edilecek mekansal sınıfı, yol ağının düğümlerini SpatialPoint sınıfından türetilen nesneler ile tanımlar.

Rectangle: Mekansal analize konu olan çalışma alanının dikdörtgen şeklinde tanımlanmasını sağlar. Uygulama yazılımı enlem ve boylam noktalarının en uç değerleri arasındaki alanı çalışma alanı olarak tanımlar. Rectangle sınıfı ilki kuzey-doğuyu belirten 4 SpatialPoint tanımından oluşur.

PointList: Bağımsız değerleri (dağılım üzerindeki etkisi araştırılacak mekansal sınıfları) temsil eden mekansal sınıflara ait tanımdır. PointList sınıfı tekil numaraya, tipini belirten metin değere ve verileri içeren SpatialPoint sınıfından türetilmiş listeye sahiptir.

EK-5 (Devam) Geliştirilen yazılımın mimarisi

Link: Yol ağını oluşturan bağlantı tanımlarıdır. Link sınıfı tekil numaraya, başlangıç noktasını ve bitiş noktalarını temsil eden SpatialPoint tanımlarına sahiptir. Span yazılımı iki nokta arasındaki bağlantıyı, başlangıç ve bitiş noktaları ve yön ile tanımlar. Bu nedenle yolun çift ya da tek yönlü olduğunu dikkate almak üzere çift yönlü yollarda düğüm çifti için iki, tek yönlü yollarda ise tek bağlantı oluşturulur.

Network: Düğüm ve bağlantı tanımlarından oluşmuş yol ağı tanımıdır. Tekil numaraya, SpatialPoint sınıfını barındıran düğüm listesine ve düğümler arası bağlantıları içerir Link sınıfı listesine sahiptir. Yol ağında bulunan tüm bağlantıların uzunluklarını toplayarak double değer ile veren CalculateLength metodu bulunmaktadır.

NumericType: Kazaya ait nümerik veri tiplerinin tanımlandığı ögedir. Kazaya ait matematiksel operatörlerin işlem yapacağı veri tiplerini barındırır. Tekil numaraya ve metin olarak isminin belirtildiği iki niteliğe sahiptir.

NominalType: Kazaya ait nominal veri tipi tanımlarını barındıran ögedir. Kazaya ait verilerden mantıksal operatörlerin uygulanabileceği veri tiplerinin tanımlarının yapıldığı sınıftır. Kazaya ait tanımlayıcı tekil numara, metin olarak isminin ve değerlerin belirtildiği tanımlara sahiptir.

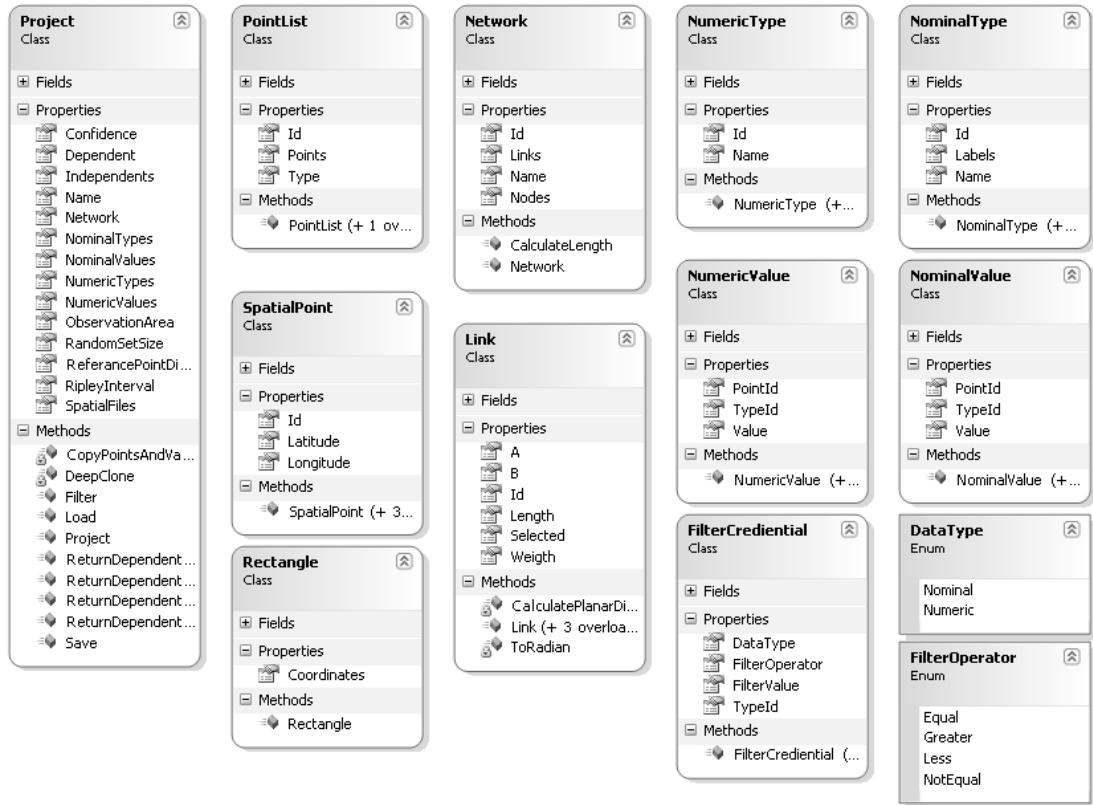
NumericValue: Kazaya ait nümerik veri tanımıdır. Kazalara ait nümerik verilerin tutulmasını sağlar. Kazaya ait tanımlayıcı tekil numara, tanımlanan nümerik veri tipine ait tekil numara ve değer ile birlikte tutulur.

NominalValue: Kazaya ait nominal veri tanımıdır. Kazaya ait tanımlayıcı tekil numara, tanımlanan nominal veri tipine ait tekil numara ve değer ile birlikte tutulur.

EK-5 (Devam) Geliştirilen yazılımın mimarisi

FilterOperator: Nümerik verilerin kıyaslanmasında kullanılacak matematik operatörleri içerir. Eşit, eşit değil, büyük ve küçük operatörlerini tanımlar.

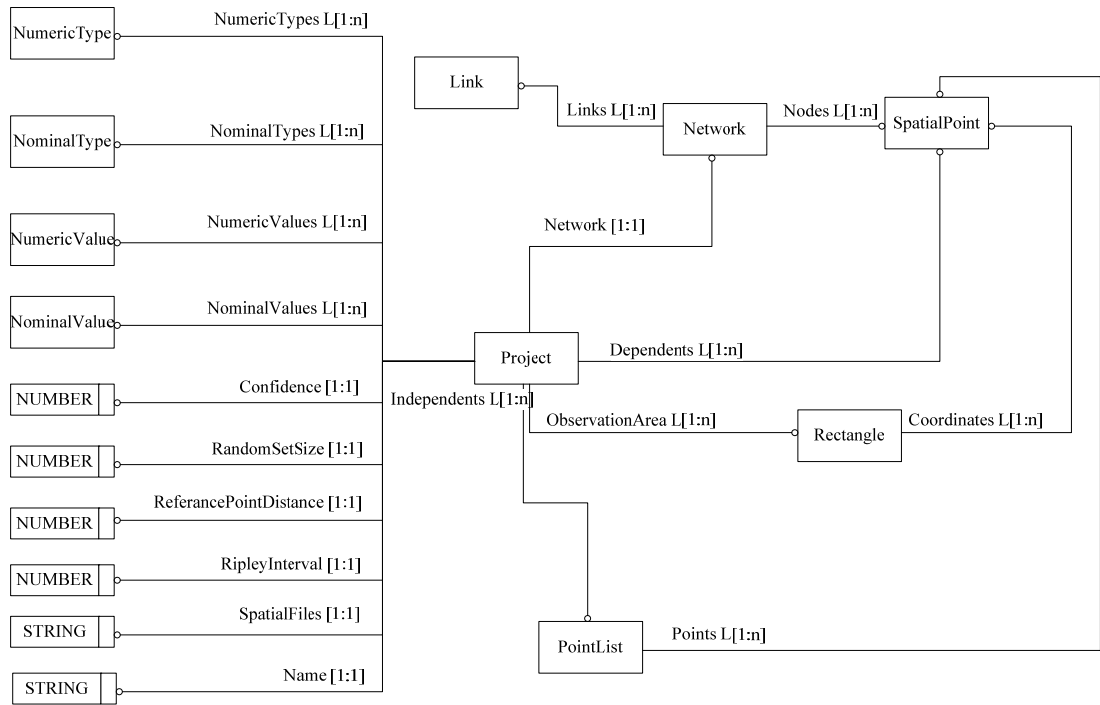
FilterCredential: Kaza örnekleminde alt küme oluşturmak üzere filtre tanım ve operatörlerini içerir öğedir. Uygulanacak olan filtrenin nümerik veya nominal olduğunu gösterir veri tipi, filtre operatörü, filtrede kıyas alınacak değer ve veri tipini tanımlayan tekil numara ile tanımlanır.



Şekil 5.2. DataEngine modülünün sınıf diyagramı

EK-5 (Devam) Geliştirilen yazılımın mimarisi

DataEngine paketi bu tanımlamaların yanında kaza örnekleme alt kümeler üretmek üzere filtreleme fonksiyonuna da sahiptir. Filtreleme fonksiyonu sayesinde; nümerik verilere matematiksel operatörler, nominal verilere ise mantıksal operatörler uygulanarak kaza kayıtlarının seçilmesi sağlanır. Oluşturulan yeni küme üzerinde analiz çalışmaları sürdürülür.



Şekil 5.3. DataEngine modülü veri modelini gösterir şema¹

StatisticsEngine modülü: Uygulama yazılımının tanımlayıcı nokta desen analizlerinin üretilmesini sağlar. Nümerik ve nominal değerler için sonuçları adapte eden NumericStatisticsResults ve NominalStatisticsResults adı ile sunulan iki sınıfı mevcuttur.

¹ Veri modeli ISO 10303-11: Industrial automation systems and integration standardına bağlı kalınarak çizilmiştir.

EK-5 (Devam) Geliştirilen yazılımın mimarisi

StatisticsEngine modülünde bulunan DescriptiveStatistics sınıfı şu fonksiyonlara sahiptir:

CalculateHistogram: Nümerik değer listesinden histogram üretilmesinde kullanılacak sayı çiftlerini belirler. Histogramın üretilmesinde veri aralıklarının genişliğinin belirlenmesi önem taşımaktadır. Veri aralığının belirlenmesinde Scott tarafından ortaya konulan formül esas alınmıştır[44]:

$$h = 3,49SEn^{1/3} \quad (3.1)$$

Eş. 3.1’de, h aralıkta gösterilecek maksimum veri adedini, SE standart sapmayı, n örneklem adedini ifade etmektedir.

CalculateKurtosis: Nümerik değer listesinin basıklık değerini verir.

CalculateMax: Nümerik değer listesinin en büyük değerini hesaplar.

CalculateMin: Nümerik değer listesinin en küçük değerini hesaplar.

CalculateMean: Nümerik değer listesinin ortalama değerini hesaplar.

CalculateSkewness: Nümerik değer listesinin çarpıklık değerini hesaplar.

CalculateStandardDeviation: Nümerik değer listesinin standart sapmasını hesaplar.

CalculateSumErrors: Nümerik değer listesinin toplam sapmasını hesaplar.

CalculateVariance: Nümerik değer listesinin varyansını hesaplar.

EK-5 (Devam) Geliştirilen yazılımın mimarisi

GetCorrelation: İki nümerik değer kümes arasındaki korelasyonu hesaplar.

DescriptiveStatistics sınıfında yer alan nitelikleri ise şunlardır:

Histogram: Nümerik değerlerin dağılımını göstermek üzere değerleri metinsel ve değer sayımlarını ise nümerik olarak tanımlayan niteliklerdir.

Kurtosis: Değerlerin basıklığını nümerik olarak tanımlayan niteliklerdir.

Max: Değerlerin en büyüğünü tanımlayan niteliklerdir.

Mean: Değerlerin aritmetik ortalamasını tanımlayan niteliklerdir.

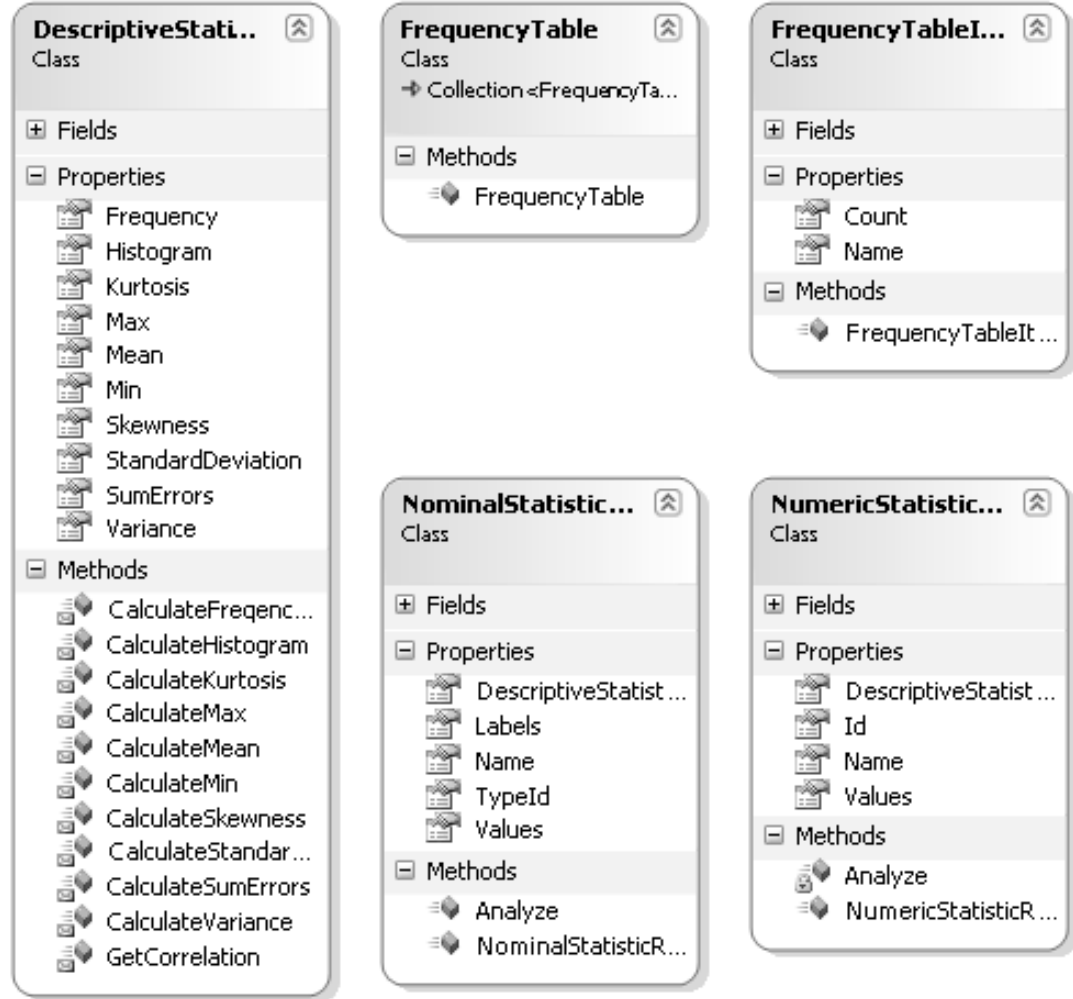
Min: Değerlerin en küçüğünü tanımlayan niteliklerdir.

Skewness: Değerlerin çarpıklığını nümerik olarak tanımlayan niteliklerdir.

StandardDeviation: Değerlerin standart sapmasını verir.

Variance: Değerlerin varyansını verir.

EK-5 (Devam) Geliştirilen yazılımın mimarisi



Şekil 5.4. StatisticsEngine modülünün sınıf diyagramı

StatisticsEngine modülü veri üretmek üzere kullanılan bir modüldür. Modülün gösterim ve kullanıcı etkileşim fonksiyonları bulunmaz.

EK-5 (Devam) Geliştirilen yazılımın mimarisi

SpatialStatisticsEngine modülü: Span yazılımının mekansal işlemleri ve analizleri bu modül tarafından gerçekleştirilir. Kullanıcı tarafından talep edilen analiz işlemleri DataEngine’de yer alan sınıflar ile tanımlanarak bu modüle parametre olarak sağlanır. Analiz çalışmasında elde edilen sonuçlar RipleyResultItem ve ondan türeyen RipleyLocalResultItem sınıfları ile döndürülür.

Modül içerisinde yer alan SpatialOperator sınıfı içerisinde yer alan metotlar şunlardır:

CalculateArea: Rectangle olarak verilen parametrenin alanını verir.

CalculateBearing: Bir noktadan diğer noktaya gitmek için takip edilecek yönü derece olarak verir.

CalculateCenterPoint: İki nokta arasında kalan orta noktayı verir.

CalculateNearestNetworkPoint: Bir referans noktasına yol ağı üzerindeki en yakın düğümü verir.

CalculateNearestPoint: Bir nokta kümesinden referans noktasına en yakın olanını verir.

CalculateNetworkDistance: Yol ağını üzerinde referans düğümünden hedef düğüme giden en kısa yol mesafesini verir.

CalculateNetworkReferancePoint: Yol ağını belirlenen aralıklarda referans noktasına böler.

CalculateNonIntersectedAreaRate: Düzlem üzerinde yer alan dairenin çalışma alanı dışında kalan kısmının toplam alana oranını verir.

EK-5 (Devam) Geliştirilen yazılımın mimarisi

CalculatePlanarDistance: İki nokta arasındaki kuş uçuşu mesafeyi verir.

CalculatePoint: Referans noktadan belirtilen istikamette belirlenen mesafedeki uzaklıktaki noktayı verir.

CalculateRandomNetworkPoint: Yol ağı üzerinde belirtilen sayıda rastgele üretilmiş noktaları verir.

CalculateRandomPlanarPoint: Çalışma alanı üzerinde belirtilen sayıda rastgele üretilmiş noktaları verir.

CalculateShortestEdgeLength: Çalışma alanının en kısa köşe uzunluğunu verir.

DegreeToRadian: Dereceden radyan değere çevirir.

RadianToDegree: Radyandan dereceye çevirir.

Modül içerisinde yer alan SpatialStatistics sınıfı ise şu metotlar ile istatistik üretir:

CalculateMeanCenter: Noktalar kümesinin koordinat merkezini verir.

CalculateNearestNeighborIndex: En yakın komşu indeksini verir.

CalculateRipleyKL: Nokta dağılımının K değerini verir.

CalculateCrossRipleyKL: Nokta dağılımını bağımsız noktalar üzerinden hesaplayarak çapraz K değerini verir.

EK-5 (Devam) Geliştirilen yazılımın mimarisi

CalculateCrossRipleyKLWithReferancePoints: Yol ağı üzerinde belirlenen referans noktalar için nokta dağılımını ve bağımsız mekansal sınıflara olan uzaklığı verir.

SpatialStatistics sınıfının RipleyResultItem sınıfında yer alan metotlar ise şunlardır:

CalculateKAverage: Rastlantısal örneklemin ortalama K değerini verir.

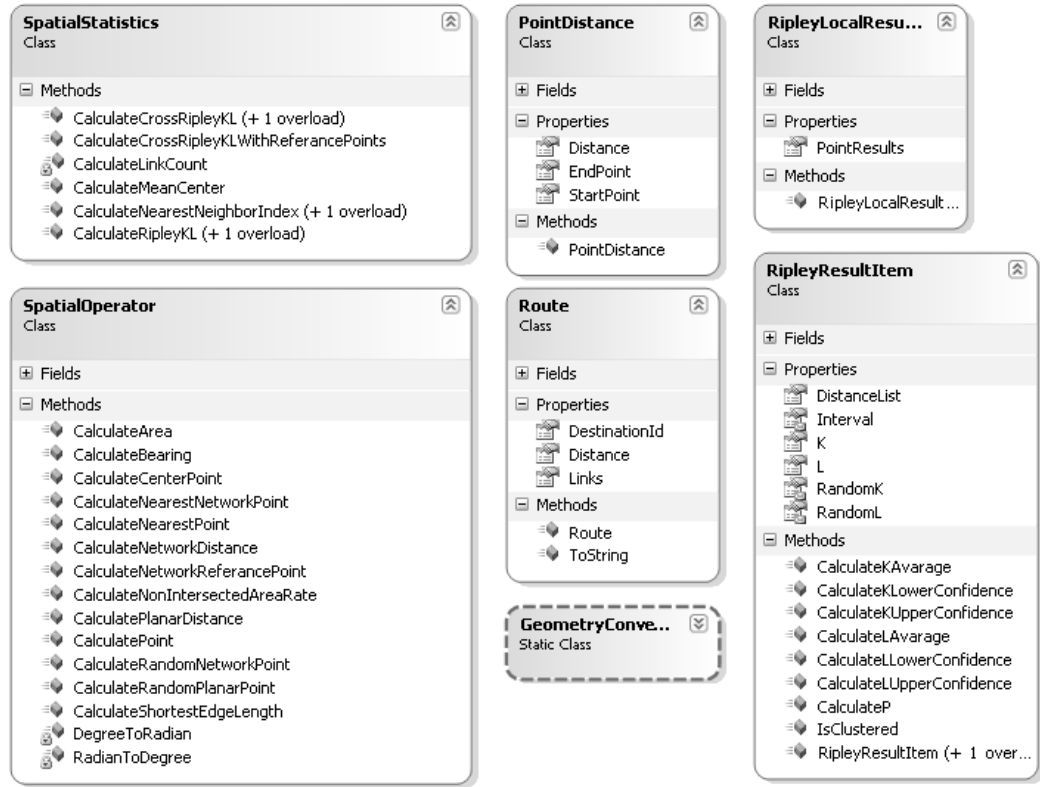
CalculateKLowerConfidence: Rastlantısal örneklemin güven aralığının alt tarafında yer alan ortalama K değerini verir.

CalculateKUpperConfidence: Rastlantısal örneklemin güven aralığının üst tarafında yer alan ortalama K değerini verir.

CalculateP: K değerinin rastlantısal örnekleme ait sonuçlar içerisinde yüzdelik değerini verir.

IsClustered: K değerini rastlantısal örneklemin sonuçları ile kıyaslayarak kümelenme gösterip göstermediğini belirtir.

EK-5 (Devam) Geliştirilen yazılımın mimarisi



Şekil 5.5. SpatialStatisticsEngine modülünün sınıf diyagramı

SpatialStatisticsEngine modülü de StatisticsEngine modülü gibi herhangi bir gösterim öğesi barındırmaz ve kullanıcı ile doğrudan etkileşimi bulunmaz.

Spantare modülü: Spantare modülü uygulama yazılımının kullanıcı etkileşimini ve iş akışını düzenleyen modüldür. Bu modül kullanıcının vermiş olduğu komutları diğer modülleri kullanarak gerçekleştirir ve sonuçlarını uygun arayüzleri kullanarak aktarır.

Spantare modülünde bulunan sınıflar etkileşimi sağlayan formlardan ve gösterime yönelik bileşenlerden oluşur:

frmMain: Span uygulama yazılımının açılış ekranını ve giriş etkileşimini başlatan form sınıfıdır.

EK-5 (Devam) Geliştirilen yazılımın mimarisi

frmProject: Uygulama yazılımında kullanıcı tarafından sağlanacak tüm veriler için etkileşimin sağlandığı form sınıfıdır. frmProject form sınıfı analizde, bağımlı değişken olduğu varsayılan trafik kazalarını, trafik kazalarının, bağımsız mekansal sınıfların ve yol ağının gösteriminde, yol ağının ve veri tiplerinin tanımlanmasında kullanılır. Form aynı zamanda bu verilerin tanımlanması için özelleşmiş form sınıflarına erişimi sağlayan butonları barındırır.

frmFilter: Kaza örnekleminden alt kümeler üretmek üzere filtre uygulanmasını sağlayan formdur.

frmDependent: Uygulama yazılımında kullanıcı tarafından trafik kazaları verilerinin, nümerik ve nominal tipteki veriler ile birlikte girilmesini, düzenlenmesini ve silinmesini sağlayan form sınıfıdır.

frmIndependent: Uygulama yazılımında trafik kazalarının dağılımı üzerinde etkisi kontrol edilecek olan mekansal sınıflara ait verilerin tanımlandığı formdur.

frmDescriptive: Noktalar arası mesafeye ait tanımlayıcı istatistiklerin üretildiği formdur. Bu form üzerinde yer alan bileşenler ile düzlem ve yol ağı üzerindeki kazalar arası mesafe dağılımlarının minimum, maksimum, ortalama, çarpıklık, basıklık, standart sapma, varyans değerleri ve histogram elde edilir.

frmSpatialAnalysis: Nokta dağılımına ait mekansal analiz sonuçlarına erişilen formdur. frmSpatialAnalysis formu kullanılarak; en yakın mesafe indeksi, global, yerel ve çapraz K değerleri gözlemlenir. En yakın mesafe indeksi metinsel veri şeklinde düzlem ve yol ağı için ayrı ayrı kullanıcıya gösterilir. Global K değeri ile grafik üzerinde kaza dağılımına etki eden ikincil bir etkinin varlığı sorgulanır. Yerel K değeri ise yol ağı üzerinde üretilen referans noktaları için kaza dağılımını harita üzerinde ve liste şeklinde gösterir.

EK-5 (Devam) Geliştirilen yazılımın mimarisi

Güven değerinin üzerinde yoğunluğa sahip olan referans noktaları kırmızı gösterilir. Çapraz K gösterimi ise; grafik üzerinde dağılım üzerinde etkisi kontrol edilen mekansal sınıflara ait her bir konum verisi için kaza yoğunluğunun gösterimi ile gerçekleştirilir.

Span modülünde metinsel gösterimler şu bileşenler ile gerçekleştirilir:

Metin gösterimleri: Span yazılımında analiz sonuçlarını belirten tekil metin verileri Microsoft Framework içinde yer alan label ve textbox sınıfları ile gösterilir. Listeleme ise aynı kütüphanenin Datagridview sınıfı içinde sağlanır.

Grafik gösterimleri: Analiz sonuçlarının grafik olarak verildiği ekranlarda grafikler LGPL¹ lisansı ile lisanslanmış olan ZedGraph² bileşeni kullanılmıştır. ZedGraph bileşeni herhangi bir hesaplama ya analiz işlemi yapmaz, sadece nümerik gruplar halinde verilen veriyi çizgi veya bar grafik ile gösterir.

Harita gösterimleri: Harita üzerinde analiz çalışmasında ulaşılan yerel K değerlerinin ve kazaların koordinat merkezlerinin gösteriminde, yol ağının oluşturulmasında ve dağılım üzerinde etkisi kontrol edilen mekansal sınıflara ait verilerin girişlerinde harita bileşeni olarak LGPL lisansı ile kamuya kullanımı için lisanslanmış SharpMap bileşeni kullanılmıştır. SharpMap bileşeni herhangi bir hesaplama veya analiz işleminde kullanılmamıştır.

¹ Gnu Kısıtlı Genel Kamu Lisansı serbest yazılımları değiştirme ve paylaşma hakkınızın mahfuz tutulması ve yazılımın bütün kullanıcıları için serbest olması amacı ile üretilmiş bir lisanstır.

² ZedGraph adlı ürünün resmi sitesi <http://zedgraph.org> tur.

EK-6 Geliştirilen uygulama yazılımının mekansal analiz fonksiyonları

Span uygulama yazılımı bütünleşik bir gösterim içinde noktasal dağılıma yönelik bulguları metin, grafik ve harita şeklinde kullanıcılara sunar:

Tanımlayıcı nokta desen analizleri: Span uygulama yazılımının analiz fonksiyonlarını işletebilmesi için öncelikle kazalara ait veri setinin uygulama yazılımı içerisinde tanımlanması gerekmektedir. Veri setinin tanımlanmasını takiben kullanıcılara kazaların mekansal dağılım desenlerini kontrol etmek üzere kazalar arasındaki mesafeler hakkında tanımlayıcı istatistiki bilgiler üretir.

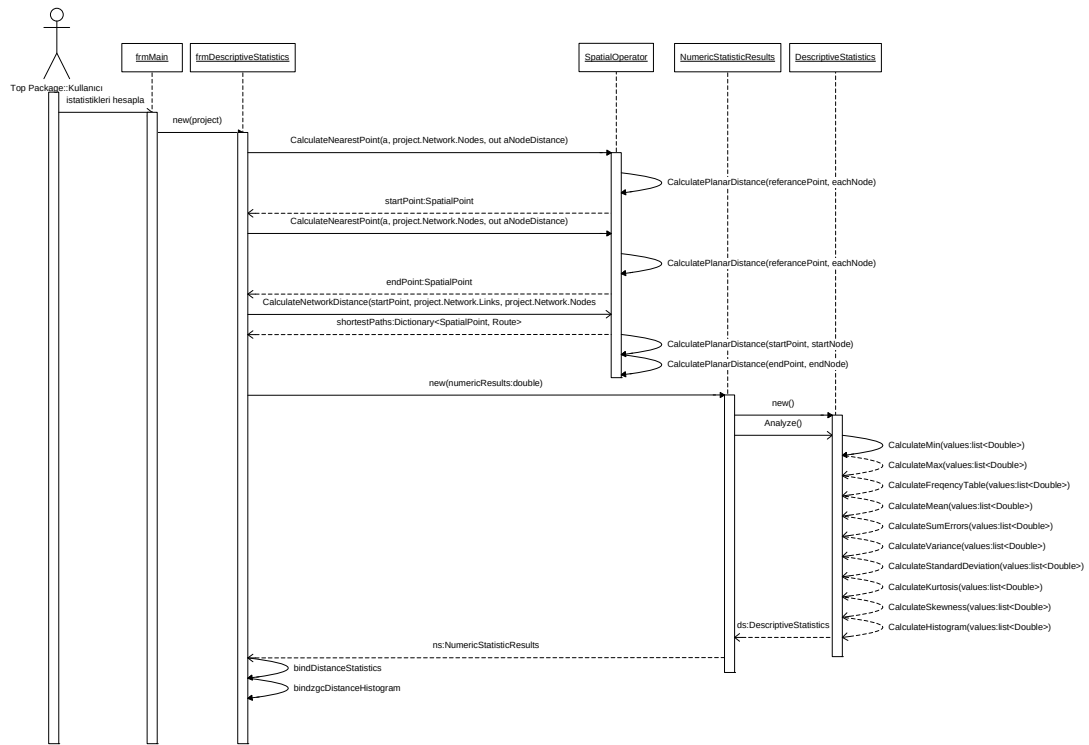
Kaza noktaları uzaklık veri seti düzlem için kuş uçuşu, yol ağı için ise en kısa güzergahın tespit edilmesi ile hesaplanır. Kullanıcı mesafeleri düzlem ve yol ağı için ayrı ayrı listeler.

Kullanıcının Span uygulama yazılımının ana formundan tanımlayıcı nokta desen analizlerini gösteren formu çağırması ile birlikte kazalar arasındaki mesafeyi hesaplamak için SpatialOperator sınıfının yeni bir nesnesi oluşturulur. Yeni oluşturulan SpatialOperator nesnesi, her bir kaza noktası için yol ağı üzerindeki en yakın düğümü ve bu düğüme olan mesafeyi bulan CalculateNearestPoint metodunu çağırır. CalculateNearestPoint metodu yol ağının her bir düğümü ile ele alınan kaza noktasını arasındaki mesafeyi CalculatePlanarDistance metodunu çağırarak hesaplar ve dönen sonuçlardan en yakın yol ağı düğümünü verir.

Yapılan işlem her kaza noktası için tekrar edilerek yol ağı üzerinde başlangıç ve varış noktaları tespit edilir. Başlangıç noktası CalculateNetworkDistance metoduna, yol ağı tanımı ile birlikte parametre olarak gönderilir ve gelen sonuç listesinde varış noktasına ait yol ağı mesafesi belirlenir. Başlangıç ve varış kaza noktalarından yol ağına erişmek için takip edilecek kuş uçuşu mesafe de yol ağı üzerindeki mesafeye eklenerek Link sınıfından türetilen nesnenin ağırlık parametresine atanır.

EK-6 (Devam) Geliştirilen uygulama yazılımının mekansal analiz fonksiyonları

Link sınıfının Length parametresi ise sınıfın oluşturulmasında tanımlan iki nokta arasındaki kuş uçuşu mesafeyi verir. Bu işlem örnekleimde yer alan tüm kazalar için tekrarlandıktan Link sınıfı listesinden oluşmuş listeden kuş uçuşu ve yol ağı mesafeleri double sınıftan oluşmuş ayrı ayrı listelere aktarılır. Listeler yeni oluşturulan NumericStatisticsResults nesnesine parametre olarak gönderilir ve nesnenin tanımlayıcı istatistiksel nitelikleri metin arayüzleri ile kullanıcıya gösterilir. NumericStatisticsResults nesnesinin Histogram niteliği ise ZedGraph nesnesi üzerinde tanımlanarak kazalar arası mesafenin histogramı üretilir.

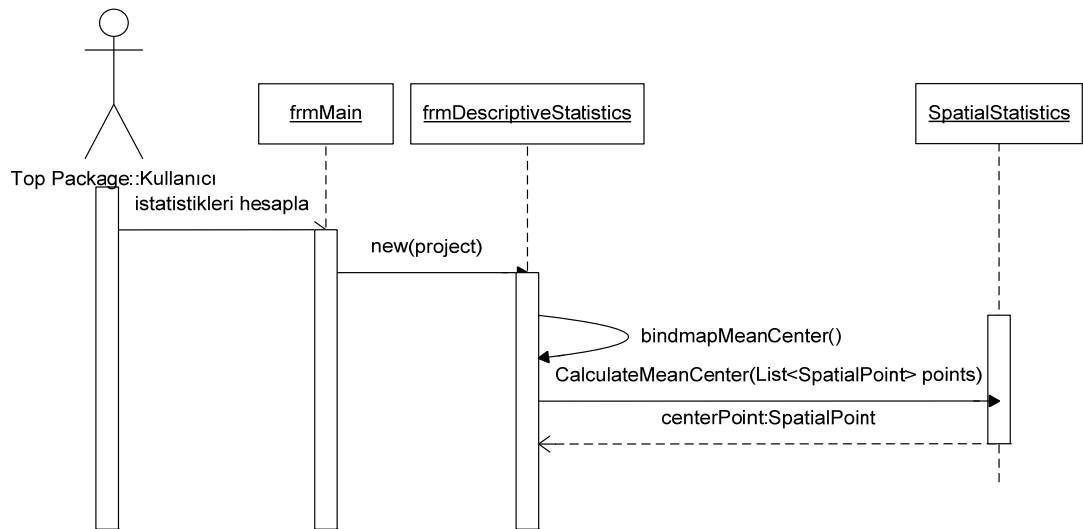


Şekil 6.1. Tanımlayıcı nokta desen analizlerinin üretilmesini gösterir akış diyagramı

EK-6 (Devam) Geliştirilen uygulama yazılımının mekansal analiz fonksiyonları

Kaza merkezi: Span uygulama yazılımının ürettiği bir diğer noktalar arası tanımlayıcı bilgi ise kaza merkezi verisidir. Kaza merkezi verisi özellikle farklı kaza tiplerinin dağılım merkezlerinin kıyaslanmasında rol oynar. Trafik kazaları temsil noktalarının koordinat merkezini, filtrelenmiş kaza örneklemeleri ile kıyaslanmak üzere kullanılır. Kaza merkezi WGS 84 sisteminde kaza enlem ve boylam koordinatlarının derece değerlerinin aritmetik ortalamasıdır.

Span uygulama yazılımı belirlenen koordinat merkezi SharpMap sınıfından türetilen bir Map nesnesi üzerinde kazaları temsil eden koordinatları siyah renkte, kaza koordinat merkezini kırmızı renkte olmak üzere yol ağı ve varsa diğer tanımlı katmanlarla birlikte haritalandırılır.

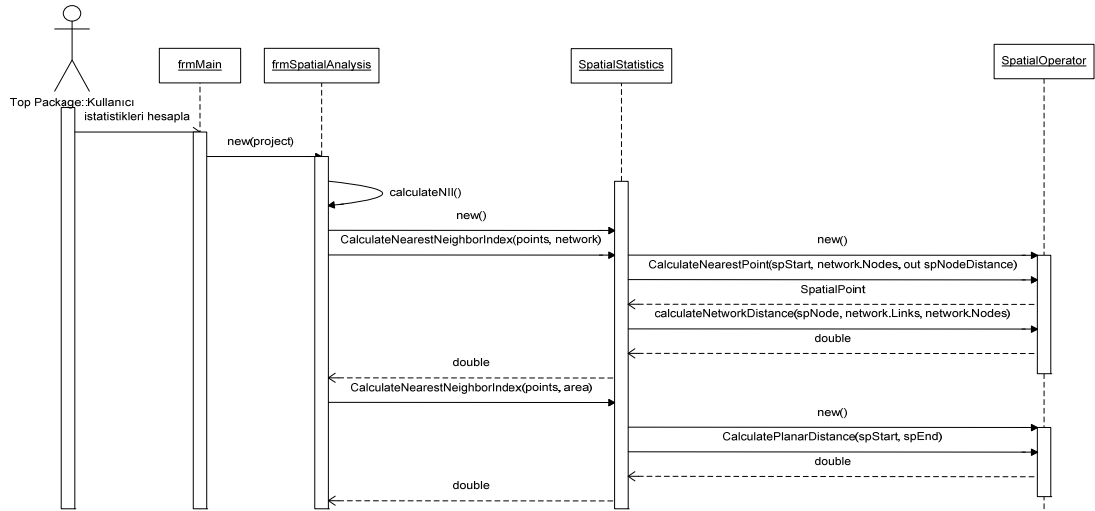


Şekil 6.2. Kaza merkezlerinin hesaplanmasını gösterir akış diyagramı

En yakın komşuluk analizi: En yakın komşuluk analizi gözlemlenen ile beklenen nokta dağılımlarında noktalar arası mesafenin kıyaslanması ile elde edilir. Veri setinin üretilmesini takiben frmSpatialAnalysis formu kullanılarak kaza konum bilgilerinin mekansal analiz sonuçlarına erişilebilir.

EK-6 (Devam) Geliştirilen uygulama yazılımının mekansal analiz fonksiyonları

Forma erişim sağlandıktan sonra mekansal analiz için gerekli hesaplamalar başlar ve hesaplama süresinin sonunda form kullanılabilir hale gelir.

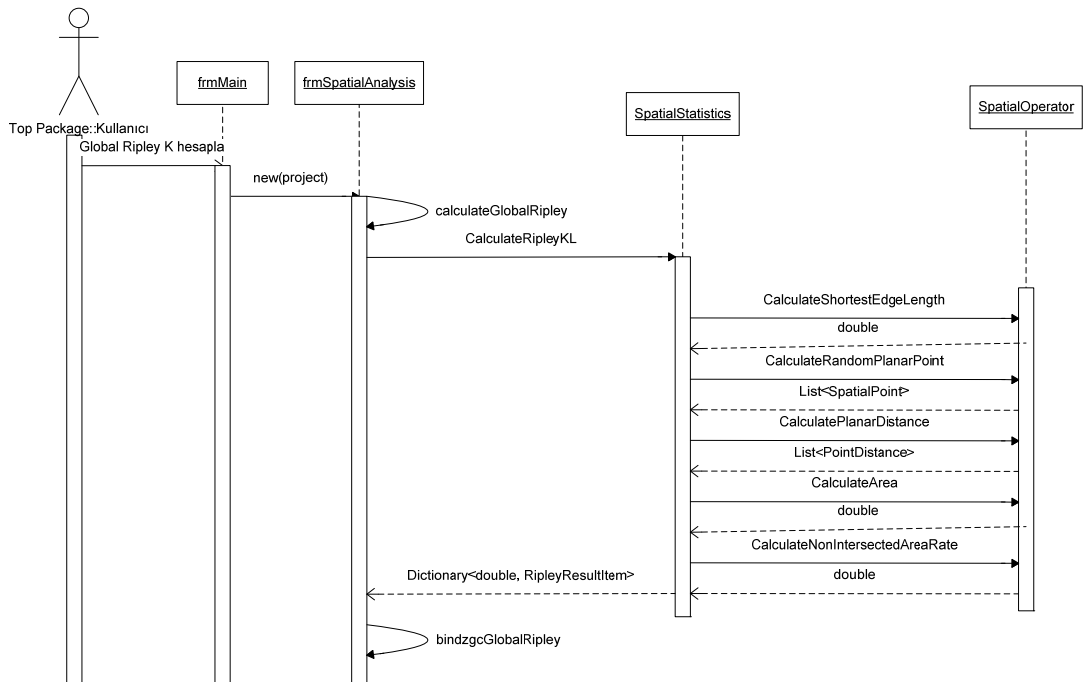


Şekil 6.3. En yakın komşu indeksinin üretilmesini gösterir akış diyagramı

Kullanıcının Span uygulama yazılımında mekansal analiz formunu açması ile birlikte mekansal analize yönelik hesaplamalar form tarafından gerçekleştirilmeye başlar. En yakın komşuluk indeksini hesaplamak için her bir kaza noktasına en yakın kaza noktası düzlem ve yol ağı üzerinden belirlenir. Bu işlem frmSpatialAnalysis formunun yüklenmesi ile birlikte oluşan event sayesinde CalculateNII fonksiyonunun çağırılması ile gerçekleşir. CalculateNII fonksiyonu SpatialStatistics nesnesinin bir kopyasını oluşturur ve ağ üzerindeki en yakın komşu indeksini hesaplamak üzere CalculateNearestNeighborIndex fonksiyonuna kazaların noktasal tanımlarını ve yol ağı tanımını veya çalışma alanını parametre olarak verir.

SpatialStatistics nesnesi SpatialOperator nesnesini kullanarak yol ağı ve düzlem üzerinden noktalar arası mesafe hesaplamalarını yapar. CalculateNII fonksiyonu tüm kazalar için döngü ile mesafe hesaplamalarını elde ettikten sonra en yakın mesafelerin aritmetik ortalamasını rastlantısal en yakın mesafe ortalamasına oranlayarak frmSpatialAnalysis formuna gönderir. Form üzerine en yakın mesafe indekslerinin gösterimi metin kutuları içerisinde olur.

Global K: Trafik kazalarının mekansal analizi denildiğinde kazaların dağılımlarının belirlenmesinde öncelikle K değerinin kontrol edilmesi gerekir. K değeri rastlantısal dağılımın güven değerleri arasında ise dağılımın rastlantısalılığı söz konusudur. Dağılım kullanıcı tarafından belirlenen güven aralığının üst kısmında ise kümelenme alt kısmında düzenlilik göstermektedir.



Şekil 6.4. Global K değerinin üretilmesini gösterir akış diyagramı

EK-6 (Devam) Geliştirilen uygulama yazılımının mekansal analiz fonksiyonları

Kullanıcı K değerini grafik üzerinde görmek için öncelikle kaza konumlarını içeren veri setini oluşturur. Ana pencerede mekansal analiz formunun yüklenmesi ile birlikte mekansal analiz metotları çağırılır. `frmSpatialAnalysis` formunun yüklenmesini takiben `CalculateGlobalRipley` fonksiyonu çağırılarak Global K değerinin hesaplanmasına yönelik işlemler başlatılır. `SpatialStatistics` sınıfından bir nesne türetilerek `CalculateRipleyKL` metodu çağırılır. `SpatialStatistics` sınıfı içerisinde `SpatialOperator` sınıfından türetilen nesnenin öncelikle `CalculateShortestEdgeLength` metodunu çağırılarak maksimum r yarıçapını bulmak üzere örneklem gözlem alanının en küçük köşe uzunluğu belirlenir. Baddeley maksimum r yarıçapının gözlem alanının en küçük köşe uzunluğunun $\frac{1}{4}$ kadar olmasını önermiştir [45].

Gözlemlenen trafik kazalarını rastlantısal veri seti ile kıyaslamak için düzlem üzerinde belirlenen rastgele set sayısı kadar rastlantısal nokta oluşturulur. Bu işlem `SpatialOperator` sınıfından türetilen nesnenin `CalculateRandomPlanarPoint` metodu çağırılarak yapılır.

`CalculatePlanarDistance` metodu çağırılarak hem kaza örnekleme hem de rastlantısal örneklem için noktalar arası mesafeler hesaplanır.

Çalışma alanının sınır çizgilerine yakın yerlerdeki noktalarda ele alınması gereken alan gözlem alanının dışında kalıyor olacağı için her noktada dışarıda kalan alanı dikkate alan bir fonksiyon ile K değerleri hesaplanır.

Sonuçlar `frmSpatialAnalysis` nesnesine `RipleyResultItem` sınıfı içinde tanımlanarak gönderilir. `frmSpatialAnalysis` formunda bulunan `bindGlobalRipley` metodu ile her bir R değeri için `RipleyResultItem` sınıfı kullanılarak R, yüzdelik aralık ve aritmetik ortalama hesaplanır. `ZedGraph` bileşenden türetilen bir grafik nesnesine sonuçlar çizgi grafik şeklinde yansıtılır.

EK-6 (Devam) Geliştirilen uygulama yazılımının mekansal analiz fonksiyonları

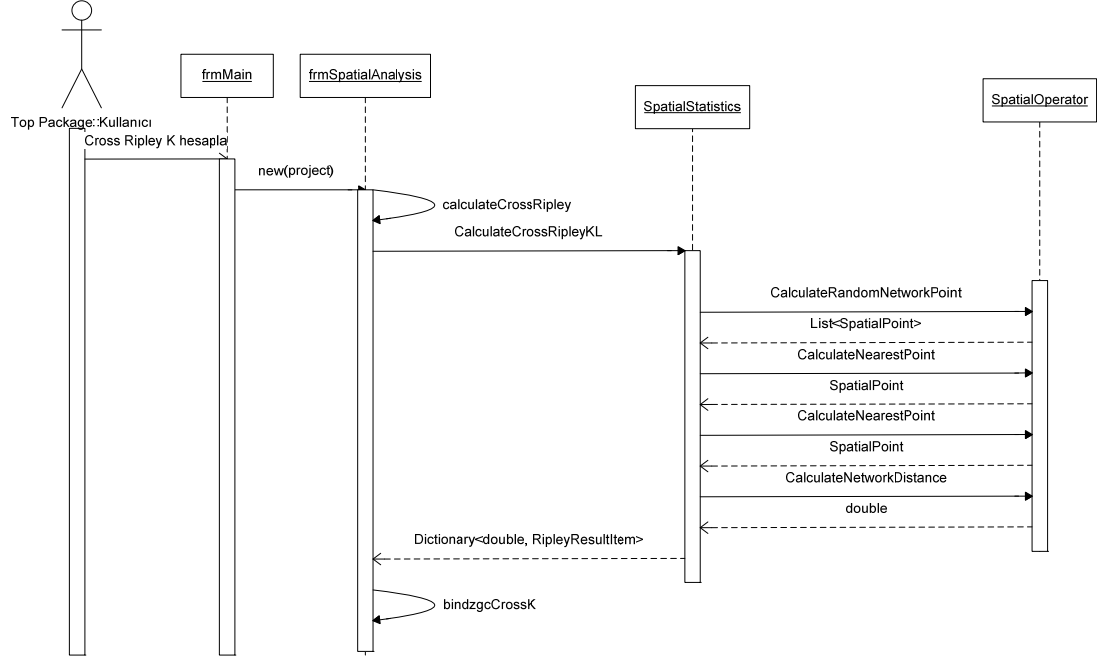
Çapraz K: Mekansal bir sınıfın, kaza dağılımları üzerindeki etkisinin araştırılması için *Çapraz K* değerlerinin hesaplanması ile gerçekleştirilir. Aynı hesaplama kazaların yerini alan rastlantısal kümeler için de tekrar edilerek sonuçlar grafik üzerinde karşılaştırılır.

Kazalara ait değerlerin; güven aralığının içerisinde olması durumunda kazaların kontrol edilen sınıfa göre rastlantısallığı, güven aralığının altında olması durumunda düzenliliği, güven aralığının üzerinde olması durumunda ise kümelenmesi söz konusudur.

Span uygulama yazılımı *çapraz K* değerini yol ağı üzerinde de hesaplar. Hesaplamaya başlamadan önce; kazalar, incelenecek olan mekansal sınıf ve yol ağı verileri tanımlanmış olmalıdır.

Mekansal analiz formunun seçilmesi ile birlikte, form yüklenir. Hesaplama işlemi *CalculateCrossRipley* metodu çağırılarak başlatılır. Takiben *SpatialStatistic* sınıfından türetilen nesnenin *CalculateCrossRipleyKL* metodu çağırılır. Bu metod mekansal hesaplamaları gerçekleştiren *SpatialOperator* nesnesinin metodlarını kullanarak rastlantısal kaza örneklemini temsil etmek üzere yol ağı üzerinde noktalar belirler. Mekansal sınıfın her bir ögesinin; kazalara ve rastlantısal örnekleme olan uzaklığı yol ağı üzerinden belirlenir. Üretilen mesafe listeleri *K* hesaplamalarında kullanılır ve her bir aralık içinde kalan kaza ve rastlantısal nokta sayısı tespit edilir. Hesaplama yol ağı kullanıldığı için *Global K*'da kullanılan düzleme dair düzeltme burada kullanılmaz.

EK-6 (Devam) Geliştirilen uygulama yazılımının mekansal analiz fonksiyonları

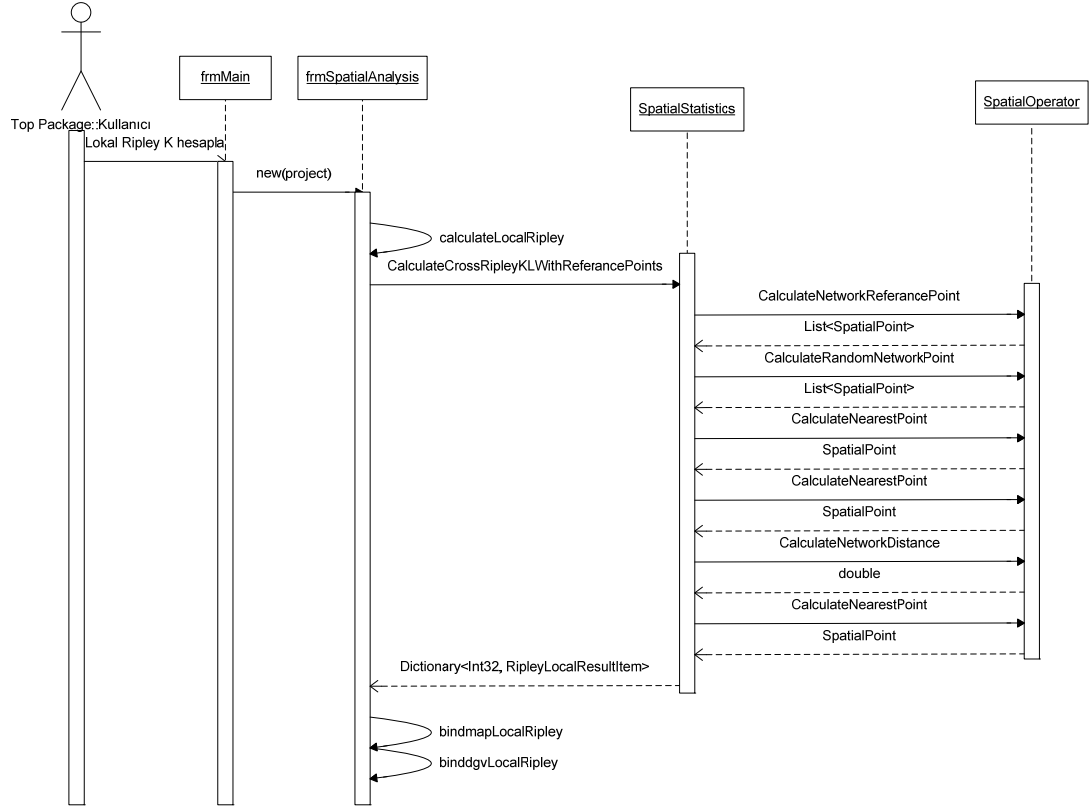


Şekil 6.5. Çapraz Ripley K değerinin üretilmesini gösterir akış diyagramı

Yerel K: Kazaların yol ağı üzerinde nerelerde kümelenme gösterdiğini belirlemek üzere Global K hesaplaması yol ağında tespit edilen referans noktalarında uygulanacak şekilde düzenlenebilir. Bu sayede yol ağı üzerinde seçilen mekansal sınıfların yol ağının referans noktalarından uzaklığı belirlenerek her bir referans noktası için kaza dağılımının nasıl olduğu gözlemlenebilir.

Referans noktaları ile çalışmada kazaların dağılımları üzerindeki etkisi sorgulanan mekansal sınıflara olan mesafelerde aynı hesaplama da üretilir. Hesaplama neticesinde her bir referans noktası için K değerini harita üzerinde görmek mümkündür. Liste gösteriminde K değeri, rastlantısal örnekleme ait yüzdelikler ve mekansal sınıfa mesafelerde liste içinde görülebilir. Kullanıcı bu listeyi daha sonra kullanmak üzere panoya kopyalayabilir.

EK-6 (Devam) Geliştirilen uygulama yazılımının mekansal analiz fonksiyonları



Şekil 6.6. Yerel K değerinin üretilmesini gösterir akış diyagramı

Yol ağı üzerindeki referans noktaları için kaza dağılımlarının durumunu görmek isteyen kullanıcı Span uygulama yazılımının Mekansal Analiz formunu açar.

Formun yüklemesi ile birlikte frmSpatialAnalysis formunda bulunan CalculateLocalRipley metodu çağırılır ve SpatialStatistics sınıfından bir nesne türetilir.

SpatialStatistics sınıfından türetilen nesnenin CalculateCrossRipleyKL-WithReferencePoints metodunun ile SpatialOperator sınıfı kullanılarak yol ağı üzerinde kullanıcı tarafından belirtilen aralıklarda referans noktaları oluşturulur.

EK-6 (Devam) Geliştirilen uygulama yazılımının mekansal analiz fonksiyonları

Bundan sonraki aşamada bu referans noktaları ile kazalar arasındaki mesafeleri barındıran bir tablo üretilir. Aynı tablo rastlantısal kazaları noktaları temsil etmek üzere üretilmiş veri seti için de üretilir. Her bir referans noktası için mesafe tabloları kullanılarak K değerleri üretilir.

Sonuç setine her referans noktası için en yakın mekansal sınıf noktasına olan uzaklıkta hesaplanarak eklenir.

frmSpatialAnalysis döndürülen sonuçlardan veri kümelerindeki rastlantısal yüzde karşılığı ve K değeri harita üzerinde gösterilir. Noktalar arası uzaklık veri seti ve K değerleri kullanıcıya ayrıntılı olarak gösterilmek üzere DataGridViewList bileşenine aktarılır.

EK-7 Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```

using System;
using System.Collections.Generic;
using System.Linq;
using Span.DataEngine;
namespace Span.SpatialStatisticsEngine
{
    public class SpatialOperator
    {
        Random r = new Random();
        #region public methods
        public Dictionary<SpatialPoint, Route> CalculateNetworkDistance(SpatialPoint
l, List<Link> links, List<SpatialPoint> nodes)
        {
            Dictionary<SpatialPoint, Route> shortestPaths = new
Dictionary<SpatialPoint, Route>();
            List<SpatialPoint> calculatedLocations = new List<SpatialPoint>();
            foreach (SpatialPoint location in nodes)
            {
                shortestPaths.Add(location, new Route(location.Id));
            }
            SpatialPoint A = new SpatialPoint();
            foreach (KeyValuePair<SpatialPoint, Route> kvp in shortestPaths)
            {
                if (kvp.Key.Latitude == l.Latitude && kvp.Key.Longitude == l.Longitude)
                {
                    A = kvp.Key;
                }
            }
        }
    }
}

```

EK-7 (Devam) Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```

    }

    shortestPaths[A].Distance = 0;

    while (calculatedLocations.Count != nodes.Count)
    {
        List<SpatialPoint> myPoints = (List<SpatialPoint>)(from s in
shortestPaths
                                                                    orderby s.Value.Distance
                                                                    select s.Key).ToList();

        SpatialPoint wPoint = null;
        foreach (SpatialPoint myPoint in myPoints)
        {
            if (!calculatedLocations.Contains(myPoint))
            {
                if (shortestPaths[myPoint].Distance == double.MaxValue)
                    return shortestPaths;
                wPoint = myPoint;
                break;
            }
        }
    }

    var myLinks = from c in links
                    where c.A.Latitude == wPoint.Latitude &&
                    c.A.Longitude == wPoint.Longitude
                    select c;

    SpatialPoint keyB = new SpatialPoint();
    SpatialPoint keyA = new SpatialPoint();
    foreach (Link myLink in myLinks)
    {
        foreach (KeyValuePair<SpatialPoint, Route> kvp in shortestPaths)

```

EK-7 (Devam) Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```
{
    if (kvp.Key.Latitude == myLink.B.Latitude && kvp.Key.Longitude
== myLink.B.Longitude)
    {
        keyB = kvp.Key;
    }
    if (kvp.Key.Latitude == myLink.A.Latitude && kvp.Key.Longitude
== myLink.A.Longitude)
    {
        keyA = kvp.Key;
    }
}
if (shortestPaths[keyB].Distance > myLink.Length +
shortestPaths[keyA].Distance)
{
    shortestPaths[keyB].Links = shortestPaths[keyA].Links.ToList();
    shortestPaths[keyB].Links.Add(myLink);
    shortestPaths[keyB].Distance = myLink.Length +
shortestPaths[keyA].Distance;
}
}
calculatedLocations.Add(wPoint);
}
return shortestPaths;
}

public SpatialPoint CalculateNearestPoint(SpatialPoint startPoint,
List<SpatialPoint> destinationPoints, out double shortestDistance)
{
    shortestDistance = double.MaxValue;
```

EK-7 (Devam) Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```

        double pDistance = 0;
        SpatialPoint nPoint = new SpatialPoint();
        foreach (SpatialPoint dp in destinationPoints)
        {
            pDistance = CalculatePlanarDistance(startPoint, dp);
            if (pDistance < shortestDistance)
            {
                shortestDistance = pDistance;
                nPoint = dp;
            }
        }
        return nPoint;
    }

    public SpatialPoint CalculateNearestNetworkPoint(SpatialPoint startPoint,
List<SpatialPoint> destinationPoints, Network network, out double shortestDistance)
    {
        shortestDistance = double.MaxValue;

        double pDistance = 0;
        SpatialPoint nPoint = new SpatialPoint();
        foreach (SpatialPoint dp in destinationPoints)
        {
            pDistance = CalculateNetworkDistance(startPoint, network.Links,
network.Nodes)[dp].Distance;
            if (pDistance < shortestDistance)
            {
                shortestDistance = pDistance;
                nPoint = dp;
            }
        }
    }

```

EK-7 (Devam) Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```

    }
        return nPoint;
    }
    public double CalculatePlanarDistance(SpatialPoint pA, SpatialPoint pB)
    {
        Double r = 6371000; //in meters
        Double dLat = DegreeToRadian(pB.Latitude - pA.Latitude);
        Double dLon = DegreeToRadian(pB.Longitude - pA.Longitude);
        double a =
            Math.Sin(dLat / 2) * Math.Sin(dLat / 2) +
            Math.Cos(DegreeToRadian(pA.Latitude)) *
            Math.Cos(DegreeToRadian(pB.Latitude)) *
            Math.Sin(dLon / 2) * Math.Sin(dLon / 2);
        double c = 2 * Math.Atan2(Math.Sqrt(a), Math.Sqrt(1-a));
        double d = r * c;
        return d;
    }
    private double RadianToDegree(double angle)
    {
        return angle * 180 / Math.PI;
    }
    private double DegreeToRadian(double angle)
    {
        return Math.PI * angle / 180;
    }
    public double CalculateBearing(SpatialPoint A, SpatialPoint B)
    {
        double latitude1 = DegreeToRadian(A.Latitude);
        double latitude2 = DegreeToRadian(B.Latitude);

```

EK-7 (Devam) Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```

double longitudeDifference = DegreeToRadian(B.Longitude - A.Longitude);
    double y = Math.Sin(longitudeDifference) * Math.Cos(latitude2);
    double x = Math.Cos(latitude1) * Math.Sin(latitude2) -
        Math.Sin(latitude1) * Math.Cos(latitude2) *
Math.Cos(longitudeDifference);
    return (RadianToDegree(Math.Atan2(y, x)) + 360) % 360;
}

public SpatialPoint CalculatePoint(SpatialPoint point, double numericDegrees,
double meter)
{
    double R = 6371000; // in meters
    double lat1 = DegreeToRadian(point.Latitude);
    double lon1 = DegreeToRadian(point.Longitude);
    double brng = DegreeToRadian(numericDegrees);
    double lat2 = Math.Asin(Math.Sin(lat1) * Math.Cos(meter / R) +
        Math.Cos(lat1) * Math.Sin(meter / R) * Math.Cos(brng));
    double lon2 = lon1 + Math.Atan2(Math.Sin(brng) * Math.Sin(meter / R) *
Math.Cos(lat1),
        Math.Cos(meter / R) - Math.Sin(lat1) * Math.Sin(lat2));
    lon2 = (lon2 + Math.PI) % (2 * Math.PI) - Math.PI;
    return new SpatialPoint(RadianToDegree(lat2), RadianToDegree(lon2));
}

public double CalculateNonIntersectedAreaRate(SpatialPoint circleCenter,
double circleRmeter, Rectangle observationarea)
{
    GisSharpBlog.NetTopologySuite.Geometries.Coordinate[]
observationAreaPoints = new
GisSharpBlog.NetTopologySuite.Geometries.Coordinate[5];

```

EK-7 (Devam) Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```

GisSharpBlog.NetTopologySuite.Geometries.Coordinate[] circlePoints = new
GisSharpBlog.NetTopologySuite.Geometries.Coordinate[360];
    SpatialPoint spN = new SpatialPoint();
    Int32 i = 0;
    foreach (SpatialPoint sp in observationarea.Coordinates)
    {
        observationAreaPoints[i] = new
GisSharpBlog.NetTopologySuite.Geometries.Coordinate(sp.Latitude, sp.Longitude);
        i++;
    }

    observationAreaPoints[4] = new
GisSharpBlog.NetTopologySuite.Geometries.Coordinate(observationarea.Coordinate
s[0].Latitude, observationarea.Coordinates[0].Longitude);
    for (int ti = 0; ti < 359; ti += 1)
    {
        spN = CalculatePoint(circleCenter, ti, circleRmeter);
        circlePoints[ti] = new
GisSharpBlog.NetTopologySuite.Geometries.Coordinate(spN.Latitude,
spN.Longitude);
    }
    spN = CalculatePoint(circleCenter, 0, circleRmeter);
    circlePoints[359] = new
GisSharpBlog.NetTopologySuite.Geometries.Coordinate(spN.Latitude,
spN.Longitude);

    GisSharpBlog.NetTopologySuite.Geometries.Polygon observationArea =
new GisSharpBlog.NetTopologySuite.Geometries.Polygon(
    new
GisSharpBlog.NetTopologySuite.Geometries.LinearRing(observationAreaPoints));

```

EK-7 (Devam) Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```

GisSharpBlog.NetTopologySuite.Geometries.Polygon    circleArea    =    new
GisSharpBlog.NetTopologySuite.Geometries.Polygon(
    new
GisSharpBlog.NetTopologySuite.Geometries.LinearRing(circlePoints));

    return    Math.Round((circleArea.Area    -
circleArea.Intersection(observationArea).Area)/100,2);
    }
    public List<SpatialPoint> CalculateRandomNetworkPoint(int sampleCount,
Network network)
    {

        Int32 linksLength = Convert.ToInt32(network.CalculateLength());
        double tempLength = 0;
        List<SpatialPoint> randomPoints = new List<SpatialPoint>();

        for (int i = 1; i <= sampleCount; i++)
        {
            Int32 randomDistance = r.Next(0, linksLength);
            tempLength = 0;
            foreach (Link link in network.Links)
            {
                tempLength += link.Length;
                if (tempLength > randomDistance)
                {
                    SpatialPoint rPoint = new SpatialPoint(CalculatePoint(link.A,
                        CalculateBearing(link.A, link.B),
                        r.Next(0, Convert.ToInt32(link.Length))));
                }
            }
        }
    }
}

```

EK-7 (Devam) Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```

randomPoints.Add(new SpatialPoint(i, rPoint.Latitude, rPoint.Longitude));
        break;
    }

    }
}
return randomPoints;
}

public double CalculateArea(Rectangle observationarea)
{
    List<double> Latitudes = new List<double>();
    List<double> Longitudes = new List<double>();
    foreach (SpatialPoint p in observationarea.Coordinates)
    {
        Latitudes.Add(p.Latitude);
        Longitudes.Add(p.Longitude);
    }
    SpatialPoint sp1 = new SpatialPoint(Latitudes.Max(), Longitudes.Max());
    SpatialPoint sp2 = new SpatialPoint(Latitudes.Max(), Longitudes.Min());
    SpatialPoint sp3 = new SpatialPoint(Latitudes.Min(), Longitudes.Min());
    SpatialPoint sp4 = new SpatialPoint(Latitudes.Min(), Longitudes.Max());
    return CalculatePlanarDistance(sp1, sp4) * CalculatePlanarDistance(sp4,
sp3);
}

public double CalculateShortestEdgeLength(Rectangle observationarea)
{
    double result = 0;
    List<double> Latitudes = new List<double>();
    List<double> Longitudes = new List<double>();

```

EK-7 (Devam) Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```

foreach (SpatialPoint p in observationarea.Coordinates)
{
    Latitudes.Add(p.Latitude);
    Longitudes.Add(p.Longitude);
}
SpatialPoint sp1 = new SpatialPoint(Latitudes.Max(),
Longitudes.Max()); //right top
SpatialPoint sp2 = new SpatialPoint(Latitudes.Max(),
Longitudes.Min()); //left top
SpatialPoint sp3 = new SpatialPoint(Latitudes.Min(), Longitudes.Min()); //left
lower
SpatialPoint sp4 = new SpatialPoint(Latitudes.Min(),
Longitudes.Max()); //right lower
if (CalculatePlanarDistance(sp1, sp4) < CalculatePlanarDistance(sp4, sp3))
{
    result = CalculatePlanarDistance(sp1, sp4);
}
else
{
    result = CalculatePlanarDistance(sp4, sp3);
}
return result;
}

public List<SpatialPoint> CalculateRandomPlanarPoint(int sampleCount,
Rectangle observationarea)
{
    double lat = 0;
    double lon = 0;

```

EK-7 (Devam) Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```

Int32 latDif = 0;
    Int32 lonDif = 0;
    double latR = 0;
    double lonR = 0;
    List<double> Latitudes = new List<double>();
    List<double> Longitudes = new List<double>();
    List<SpatialPoint> randomPoints = new List<SpatialPoint>();
    foreach (SpatialPoint p in observationarea.Coordinates)
    {
        Latitudes.Add(p.Latitude);
        Longitudes.Add(p.Longitude);
    }

    for (int i = 1; i <= sampleCount; i++)
    {
        latDif = Convert.ToInt32((Latitudes.Max() - Latitudes.Min())*1000000);
        lonDif = Convert.ToInt32((Longitudes.Max() -
Longitudes.Min())*1000000);
        latR = r.Next(0, latDif);
        lonR = r.Next(0, lonDif);
        lat = Latitudes.Min() + Convert.ToDouble(latR / 1000000);
        lon = Longitudes.Min() + Convert.ToDouble(lonR / 1000000);
        randomPoints.Add(new SpatialPoint(i,lat,lon));
    }
    return randomPoints;
}

public List<SpatialPoint> CalculateNetworkReferancePoint(double distance,
Network network)
{

```

EK-7 (Devam) Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```

Int32 localSectionNumber;

List<SpatialPoint> referencePoints = new List<SpatialPoint>();

#region split links
foreach (Link link in network.Links)
{
    if ((network.Links.IndexOf(link) & 1) == 1)
    {
        if (link.Length < distance)
        {
            referencePoints.Add(new SpatialPoint(referencePoints.Count,
CalculatePoint(link.A, CalculateBearing(link.A, link.B), link.Length / 2).Latitude,
CalculatePoint(link.A, CalculateBearing(link.A, link.B), link.Length /
2).Longitude));
        }
        else
        {
            localSectionNumber = Convert.ToInt32(Math.Floor(link.Length /
distance));
            if (localSectionNumber != 0)
            {
                for (int i = 1; i <= localSectionNumber; i++)
                {
                    if (distance * i < link.Length)
                    {
                        referencePoints.Add(new SpatialPoint( referencePoints.Count,
CalculatePoint(link.A, CalculateBearing(link.A, link.B), distance * i).Latitude,
CalculatePoint(link.A, CalculateBearing(link.A, link.B), distance * i).Longitude));
                    }
                }
            }
        }
    }
}

```

EK-7 (Devam) Geliştirilen yazılımın SpatialOperator.cs sınıfı kaynak kodu

```

        }
    }

    }
}

}

#endregion
return referencePoints;
}

public SpatialPoint CalculateCenterPoint(SpatialPoint A, SpatialPoint B)
{
    return CalculatePoint(A, CalculateBearing(A, B), CalculatePlanarDistance(A,
B)/2);
}

#endregion public methods
}
}

```

EK-8 Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```
using System;
using System.Collections.Generic;
using System.Linq;
using Span.DataEngine;
namespace Span.SpatialStatisticsEngine
{
    public class SpatialStatistics
    {
        #region public methods
        public Dictionary<double, RipleyResultItem> CalculateRipleyKL(
            Int32 intervalCount,
            Int32 randomSampleCount,
            List<SpatialPoint> points,
            Rectangle observationArea,
            Network network)
        {
            SpatialOperator sOperator = new SpatialOperator();
            double maxRmeter =
sOperator.CalculateShortestEdgeLength(observationArea) / 4; //maxR
            # region creaaate random points
            List<List<SpatialPoint>> randomPointsList = new
List<List<SpatialPoint>>(); //random network points
            for (int t = 1; t <= randomSampleCount; t++)
            {
                randomPointsList.Add(new
List<SpatialPoint>(sOperator.CalculateRandomNetworkPoint(points.Count,
network)));
            }
        }
    }
    #endregion
}
```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

# region calculate sample distance list
List<PointDistance> sampleDistanceList = new List<PointDistance>();
//sample point distance
double spNodeDistance = 0; //sample start point distance
double epNodeDistance = 0; //sample end point distance

foreach (SpatialPoint spStart in points)
{
    foreach (SpatialPoint spEnd in points)
    {
        if (spStart != spEnd)
        {
            //find nearest node for start point
            SpatialPoint spNode = new
SpatialPoint(sOperator.CalculateNearestPoint(spStart, network.Nodes, out
spNodeDistance)); ;
            //find nearest node for destination point
            SpatialPoint epNode = new
SpatialPoint(sOperator.CalculateNearestPoint(spEnd, network.Nodes, out
epNodeDistance));
            Dictionary<SpatialPoint, Route> result = new
Dictionary<SpatialPoint, Route>();
            result = sOperator.CalculateNetworkDistance(spNode, network.Links,
network.Nodes);
            double networkDistance = 0;
            foreach (KeyValuePair<SpatialPoint, Route> kvp in result)
            {
                if (kvp.Key.Latitude== epNode.Latitude && kvp.Key.Longitude
== epNode.Longitude)

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        {
            networkDistance = kvp.Value.Distance;
            break;
        }
    }

    sampleDistanceList.Add(new PointDistance(spStart.Id, spEnd.Id, +
spNodeDistance + networkDistance + epNodeDistance));
    }
    else
    {
        sampleDistanceList.Add(new PointDistance(spStart.Id, spEnd.Id, 0));
    }
}

#endregion

# region calculate random point distance list
List<List<PointDistance>> randomDistanceList = new
List<List<PointDistance>>(); //random points distance List
List<PointDistance> randomDistanceSet = new List<PointDistance>();
//random points distance set
double rspNodeDistance = 0; //random start point distance
double repNodeDistance = 0; //random end point distance

foreach (List<SpatialPoint> randompointssample in randomPointsList)
{
    foreach (SpatialPoint rspStart in randompointssample)
    {

        //for each point
    }
}

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        foreach (SpatialPoint rspEnd in randompointssample)
        {
            if (rspStart != rspEnd)
            {
                //find nearest node for start point
                SpatialPoint      rspNode      =      new
                SpatialPoint(sOperator.CalculateNearestPoint(rspStart,      network.Nodes,      out
                rspNodeDistance));
                //find nearest node for destination point
                SpatialPoint      repNode      =      new
                SpatialPoint(sOperator.CalculateNearestPoint(rspEnd,      network.Nodes,      out
                repNodeDistance)); ;
                Dictionary<SpatialPoint,      Route>      resultRandom      =      new
                Dictionary<SpatialPoint, Route>();
                resultRandom      =      sOperator.CalculateNetworkDistance(rspNode,
                network.Links, network.Nodes);
                double networkRandomDistance = 0;
                foreach (KeyValuePair<SpatialPoint,      Route>      kvpRandom      in
                resultRandom)
                {
                    if (kvpRandom.Key.Latitude      ==      repNode.Latitude      &&
                    kvpRandom.Key.Longitude == repNode.Longitude)
                    {
                        networkRandomDistance = kvpRandom.Value.Distance;
                        break;
                    }
                }
                randomDistanceSet.Add(new PointDistance(rspStart.Id, rspEnd.Id,
                +rspNodeDistance + networkRandomDistance + epNodeDistance));
            }
        }
    }
}

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        }
        else
        {
            randomDistanceSet.Add(new PointDistance(rspStart.Id, rspEnd.Id,
0));
        }

    }
}

randomDistanceList.Add(new List<PointDistance>(randomDistanceSet));
randomDistanceSet.Clear();
}

#endregion
#region ripley K L calculation
Dictionary<double, RipleyResultItem> results = new Dictionary<double,
RipleyResultItem>();

    Double    interval    =    Math.Round(Convert.ToDouble(maxRmeter    /
(intervalCount - 1)), 3); //the last interval is the maxRmeter
    Int32 k = 0;
    double distance = 0;
    //link is doubled to reach with two direction so length /2
    double    fanm    =    (network.CalculateLength()/2)    /    (points.Count    *
points.Count);
    for (int i = 0; i <= intervalCount; i++)
    {
        distance += interval; //add interval to distance
        results.Add(distance, new RipleyResultItem(distance));
    }
}

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        results[distance].K = fanm * sampleDistanceList.FindAll(p => p.Distance
<= distance).Count;
        results[distance].L = Math.Sqrt(results[distance].K / Math.PI) - distance;

        k = 0;
        foreach (List<SpatialPoint> randomPointSet in randomPointsList)
        {
            results[distance].RandomK.Add(fanm
randomDistanceList[k].FindAll(p => p.Distance <= distance).Count);
            results[distance].RandomL.Add(Math.Sqrt(fanm
randomDistanceList[k].FindAll(p => p.Distance <= distance).Count / Math.PI) -
distance);
            k++;
        }

    }
    #endregion
    return results;
}

public Dictionary<double, RipleyResultItem> CalculateRipleyKL(
    Int32 intervalCount,
    Int32 randomSampleCount,
    List<SpatialPoint> points,
    Rectangle observationArea)
{
    SpatialOperator sOperator = new SpatialOperator();
    double maxRmeter =
sOperator.CalculateShortestEdgeLength(observationArea) / 4; //maxR is 1/2 shortest
edge

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```
# region creaate random points
List<List<SpatialPoint>>> randomPointsList = new
List<List<SpatialPoint>>>(); //random network points
for (int t = 1; t <= randomSampleCount; t++)
{
    randomPointsList.Add(new
List<SpatialPoint>(sOperator.CalculateRandomPlanarPoint(points.Count,
observationArea)));
}
#endregion

# region calculate sample distance list
List<PointDistance> sampleDistanceList = new List<PointDistance>();
//sample point distance
foreach (SpatialPoint spStart in points)
{
    //for each point
    foreach (SpatialPoint spEnd in points)
    {
        sampleDistanceList.Add(new PointDistance(spStart.Id, spEnd.Id,
sOperator.CalculatePlanarDistance(spStart, spEnd)));
    }
}
#endregion

# region calculate random point distance list
List<List<PointDistance>>> randomDistanceList = new
List<List<PointDistance>>>(); //random points distance List
List<PointDistance> randomDistanceSet = new List<PointDistance>();
//random points distance set
```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        foreach (List<SpatialPoint> randompointssample in randomPointsList)
        {
            foreach (SpatialPoint rspStart in randompointssample)
            {
                foreach (SpatialPoint rspEnd in randompointssample)
                {
                    randomDistanceSet.Add(new PointDistance(rspStart.Id, rspEnd.Id,
sOperator.CalculatePlanarDistance(rspStart, rspEnd)));

                }
            }
            randomDistanceList.Add(new List<PointDistance>(randomDistanceSet));
            randomDistanceSet.Clear();
        }
        #endregion
        #region ripley K L calculation
        Dictionary<double, RipleyResultItem> results = new Dictionary<double,
RipleyResultItem>();
        Double interval = Math.Round(Convert.ToDouble(maxRmeter /
(intervalCount - 1)),2); //the last interval is the maxRmeter
        Int32 k = 0;
        double distance = 0;
        double rPointTotal = 0;
        double randomPointTotal = 0;

        double fanm = sOperator.CalculateArea(observationArea) / (points.Count *
points.Count);
        for (int i = 0; i <= intervalCount; i++)

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

{
    distance += interval; //add interval to distance
    //rPointTotal = sampleDistanceList.FindAll(p => p.Distance <=
distance).Count;
    foreach (SpatialPoint sPoint in points)
    {
        rPointTotal += (sampleDistanceList.FindAll(p => p.Distance <=
distance && p.StartPoint==sPoint.Id).Count * (1+
sOperator.CalculateNonIntersectedAreaRate(sPoint, distance, observationArea)));
    }
    results.Add(distance, new RipleyResultItem(distance));
    results[distance].K = fanm * rPointTotal;
    results[distance].L = Math.Sqrt(results[distance].K / Math.PI) - distance;
    rPointTotal = 0;
    k = 0;
    foreach (List<SpatialPoint> randomPointSet in randomPointsList)
    {
        foreach (SpatialPoint rsP in randomPointSet)
        {
            randomPointTotal += (randomDistanceList[k].FindAll(p =>
p.Distance <= distance && p.StartPoint == rsP.Id).Count * (1 +
sOperator.CalculateNonIntersectedAreaRate(rsP, distance, observationArea)));
        }
        randomPointTotal = randomDistanceList[k].FindAll(p => p.Distance <=
distance).Count;
        results[distance].RandomK.Add(fanm * randomPointTotal);
        results[distance].RandomL.Add(Math.Sqrt(fanm * randomPointTotal /
Math.PI) - distance);
    }
}

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        randomPointTotal = 0;
        k++;
    }

}

#endregion

return results;
}

public Dictionary<double, RipleyResultItem> CalculateCrossRipleyKL(
    Int32 intervalCount,
    Int32 randomSampleCount,
    List<SpatialPoint> dependentpoints,
    List<SpatialPoint> independentpoints,
    Rectangle observationArea,
    Network network)
{
    SpatialOperator sOperator = new SpatialOperator();
    double maxRmeter = 0;
    if ((network.CalculateLength() / 8) > 2000)
    {maxRmeter = 2000;}
    else
    {maxRmeter = network.CalculateLength() / 8;}
    # region create random points
    List<List<SpatialPoint>> randomPointsList = new
List<List<SpatialPoint>>(); //random network points
    for (int t = 1; t <= randomSampleCount; t++)
    {

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        randomPointsList.Add(new
List<SpatialPoint>(sOperator.CalculateRandomNetworkPoint(dependentpoints.Count, network)));
    }
    #endregion
    # region calculate sample distance list
    List<PointDistance> sampleDistanceList = new List<PointDistance>();
//sample point distance
    double spNodeDistance = 0; //sample start point distance
    double epNodeDistance = 0; //sample end point distance
    foreach (SpatialPoint spStart in independentpoints)
    {
        //for each point
        foreach (SpatialPoint spEnd in dependentpoints)
        {
            if (spStart != spEnd)
            {
                //find nearest node for start point
                SpatialPoint spNode = new
SpatialPoint(sOperator.CalculateNearestPoint(spStart, network.Nodes, out
spNodeDistance)); ;
                //find nearest node for destination point
                SpatialPoint epNode = new
SpatialPoint(sOperator.CalculateNearestPoint(spEnd, network.Nodes, out
epNodeDistance));
                Dictionary<SpatialPoint, Route> result = new
Dictionary<SpatialPoint, Route>();
                result = sOperator.CalculateNetworkDistance(spNode, network.Links,
network.Nodes);
            }
        }
    }
}

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        double networkDistance = 0;
        foreach (KeyValuePair<SpatialPoint, Route> kvp in result)
        {
            if (kvp.Key.Latitude == epNode.Latitude && kvp.Key.Longitude
== epNode.Longitude)
            {
                networkDistance = kvp.Value.Distance;
                break;
            }
            sampleDistanceList.Add(new PointDistance(spStart.Id, spEnd.Id,
+spNodeDistance + networkDistance + epNodeDistance));
        }
        else
        {
            sampleDistanceList.Add(new PointDistance(spStart.Id, spEnd.Id, 0));
        }
    }
}

#endregion

# region calculate random point distance list
List<List<PointDistance>> randomDistanceList = new
List<List<PointDistance>>(); //random points distance List
List<PointDistance> randomDistanceSet = new List<PointDistance>();
//random points distance set
double rspNodeDistance = 0; //random start point distance
double repNodeDistance = 0; //random end point distance
foreach (List<SpatialPoint> randompointssample in randomPointsList)
{

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

foreach (SpatialPoint rspStart in randompointsample)
{
    //for each point
    foreach (SpatialPoint rspEnd in independentpoints)
    {
        if (rspStart != rspEnd)
        {
            //find nearest node for start point
            SpatialPoint      rspNode      =      new
SpatialPoint(sOperator.CalculateNearestPoint(rspStart,      network.Nodes,      out
rspNodeDistance));
            //find nearest node for destination point
            SpatialPoint      repNode      =      new
SpatialPoint(sOperator.CalculateNearestPoint(rspEnd,      network.Nodes,      out
repNodeDistance)); ;
            Dictionary<SpatialPoint,      Route>      resultRandom      =      new
Dictionary<SpatialPoint, Route>();
            resultRandom      =      sOperator.CalculateNetworkDistance(rspNode,
network.Links, network.Nodes);
            double networkRandomDistance = 0;
            foreach (KeyValuePair<SpatialPoint,      Route>      kvpRandom      in
resultRandom)
            {
                if (kvpRandom.Key.Latitude      ==      repNode.Latitude      &&
kvpRandom.Key.Longitude == repNode.Longitude)
                {
                    networkRandomDistance = kvpRandom.Value.Distance;
                    break;
                }
            }
        }
    }
}

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        }
        randomDistanceSet.Add(new PointDistance(rspStart.Id, rspEnd.Id,
+spNodeDistance + networkRandomDistance + epNodeDistance));
    }
    else
    {
        randomDistanceSet.Add(new PointDistance(rspStart.Id, rspEnd.Id,
0));
    }
}
}
randomDistanceList.Add(new List<PointDistance>(randomDistanceSet));
randomDistanceSet.Clear();
}
#endregion
#region ripley K L calculation
Dictionary<double, RipleyResultItem> results = new Dictionary<double,
RipleyResultItem>();
    Double interval = Math.Round(Convert.ToDouble(maxRmeter /
(intervalCount - 1)), 3); //the last interval is the maxRmeter
    Int32 k = 0;
    double distance = 0;
    //link is doubled to reach with two direction so length /2
    double fanm = (network.CalculateLength()/2) / (dependentpoints.Count);
    for (int i = 0; i <= intervalCount; i++)
    {
        distance += interval; //add interval to distance
        results.Add(distance, new RipleyResultItem(distance));
    }
}

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        results[distance].K = fanm * sampleDistanceList.FindAll(p => p.Distance
<= distance).Count;
        results[distance].L = Math.Sqrt(results[distance].K / Math.PI) - distance;
        k = 0;
        foreach (List<SpatialPoint> randomPointSet in randomPointsList)
        {
            results[distance].RandomK.Add(fanm *
randomDistanceList[k].FindAll(p => p.Distance <= distance).Count);
            results[distance].RandomL.Add(Math.Sqrt(fanm *
randomDistanceList[k].FindAll(p => p.Distance <= distance).Count / Math.PI) -
distance);
            k++;
        }

    }
    #endregion
    return results;
}

public Dictionary<double, RipleyResultItem> CalculateCrossRipleyKL(
    Int32 intervalCount,
    Int32 randomSampleCount,
    List<SpatialPoint> dependentpoints,
    List<SpatialPoint> independentpoints,
    Rectangle observationArea)
{
    SpatialOperator sOperator = new SpatialOperator();
    Int32 maxRmeter =
Convert.ToInt32(sOperator.CalculateShortestEdgeLength(observationArea) / 4);
    //maxR is 1/2 shortest edge

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```
# region creaate random points
    List<List<SpatialPoint>> randomPointsList = new
List<List<SpatialPoint>>(); //random network points
    for (int t = 1; t <= randomSampleCount; t++)
    {
        randomPointsList.Add(new
List<SpatialPoint>(sOperator.CalculateRandomPlanarPoint(dependentpoints.Count,
observationArea)));
    }
#endregion
# region calculate sample distance list
    List<PointDistance> sampleDistanceList = new List<PointDistance>();
//sample point distance
    foreach (SpatialPoint spStart in independentpoints)
    {
        //for each point
        foreach (SpatialPoint spEnd in dependentpoints)
        {
            sampleDistanceList.Add(new PointDistance(spStart.Id, spEnd.Id,
sOperator.CalculatePlanarDistance(spStart, spEnd)));
        }
    }
#endregion
# region calculate random point distance list
    List<List<PointDistance>> randomDistanceList = new
List<List<PointDistance>>(); //random points distance List
    List<PointDistance> randomDistanceSet = new List<PointDistance>();
//random points distance set
    foreach (List<SpatialPoint> randompointssample in randomPointsList)
```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

{
    foreach (SpatialPoint rspStart in randompointsample)
    {
        foreach (SpatialPoint rspEnd in independentpoints)
        {
            randomDistanceSet.Add(new PointDistance(rspStart.Id, rspEnd.Id,
sOperator.CalculatePlanarDistance(rspStart, rspEnd)));
        }
    }
    randomDistanceList.Add(new List<PointDistance>(randomDistanceSet));
    randomDistanceSet.Clear();
}
#endregion
#region ripley K L calculation
Dictionary<double, RipleyResultItem> results = new Dictionary<double,
RipleyResultItem>();
    Int32 interval = Convert.ToInt32(maxRmeter / intervalCount - 1); //the last
interval is the maxRmeter
    double distance = 0;
    double samplePointCount = 0;
    double rPointTotal = 0;
    Int32 k = 0;
    double randomPointCount = 0;
    double randomPointTotal = 0;
    double n2 = dependentpoints.Count() * independentpoints.Count();
    double factor = sOperator.CalculateArea(observationArea) / n2;
    for (int i = 0; i <= intervalCount; i++)
    {
        distance += interval; //add interval to distance

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        foreach (SpatialPoint sPoint in independentpoints)
        {
            samplePointCount = 0;
            foreach (PointDistance pd in sampleDistanceList)
            {
                if (sPoint.Id == pd.StartPoint & pd.Distance <= distance)
                {
                    samplePointCount++;
                }
            }
            rPointTotal += (samplePointCount *
sOperator.CalculateNonIntersectedAreaRate(sPoint, distance, observationArea));
        }
        results.Add(distance, new RipleyResultItem(distance));
        results[distance].K = factor * rPointTotal;
        results[distance].L = Math.Sqrt(results[distance].K / Math.PI) - distance;
        rPointTotal = 0;
        k = 0;
        foreach (List<SpatialPoint> randomPointSet in randomPointsList)
        {

            foreach (SpatialPoint rsP in independentpoints)
            {
                randomPointCount = 0;
                foreach (PointDistance rpd in randomDistanceList[k])
                {
                    if (rsP.Id == rpd.StartPoint & rpd.Distance <= distance)
                    {
                        randomPointCount++;
                    }
                }
            }
        }
    }
}

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        }
    }
    randomPointTotal += (randomPointCount *
sOperator.CalculateNonIntersectedAreaRate(rsP, distance, observationArea));
    }
    results[distance].RandomK.Add(factor * randomPointTotal);
    results[distance].RandomL.Add(Math.Sqrt((factor * randomPointTotal)
/ Math.PI) - distance);
    rPointTotal = 0;
    k++;
}

}
#endregion
return results;
}

public double CalculateNearestNeighborIndex(
    List<SpatialPoint> points,
    Network network)
{
    SpatialOperator sOperator = new SpatialOperator();
    List<PointDistance> sampleDistanceList = new List<PointDistance>();
//sample point distance
    double spNodeDistance = 0; //sample start point distance
    double epNodeDistance = 0; //sample end point distance
    double dNN = 0; //sample start - end point distance
    double shortestDistance = double.MaxValue;
    double calculatedDistance = 0;
    //this dRan mean expected mean distance odf points

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```
//it could be calculated with totaldistance/n
double dRan = network.CalculateLength() / points.Count;
SpatialPoint spNode = new SpatialPoint();
SpatialPoint epNode = new SpatialPoint();
foreach (SpatialPoint spStart in points)
{
    //find nearest node for start point
    spNode = sOperator.CalculateNearestPoint(spStart, network.Nodes, out
spNodeDistance);
    //for each point
    foreach (SpatialPoint spEnd in points)
    {
        if (spStart.Id != spEnd.Id)
        {
            //find nearest node for destination point
            epNode = sOperator.CalculateNearestPoint(spEnd, network.Nodes,
out epNodeDistance);
            calculatedDistance = sOperator.CalculateNetworkDistance(spNode,
network.Links, network.Nodes)[epNode].Distance + spNodeDistance +
epNodeDistance;
            if (calculatedDistance < shortestDistance)
            {
                shortestDistance = calculatedDistance;
            }
        }
    }
    dNN += shortestDistance / points.Count;
    shortestDistance = double.MaxValue;
}
```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

    }
    return dNN / dRan;
}

public double CalculateNearestNeighborIndex(
    List<SpatialPoint> points,
    Rectangle observationArea)
{
    SpatialOperator sOperator = new SpatialOperator();
    List<PointDistance> sampleDistanceList = new List<PointDistance>();
//sample point distance
    double dNN = 0; //sample start - end point distance
    double shortestDistance = double.MaxValue;
    double calculatedDistance = 0;
    double dRan = 0.5 * Math.Sqrt(sOperator.CalculateArea(observationArea) /
points.Count);
    foreach (SpatialPoint spStart in points)
    {
        //for each point
        foreach (SpatialPoint spEnd in points)
        {
            if (spStart.Id != spEnd.Id)
            {
                //find nearest node for destination point
                calculatedDistance = sOperator.CalculatePlanarDistance(spStart,
spEnd);
                if (calculatedDistance < shortestDistance)
                {
                    shortestDistance = calculatedDistance;
                }
            }
        }
    }
}

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        }
    }
    dNN += shortestDistance / points.Count;
    shortestDistance = double.MaxValue;
}
return dNN / dRan;
}

public SpatialPoint CalculateMeanCenter(
    List<SpatialPoint> points)
{
    SpatialPoint sPoint = new SpatialPoint();
    double latTotal = 0;
    double lonTotal = 0;
    foreach (SpatialPoint sp in points)
    {
        latTotal += sp.Latitude;
        lonTotal += sp.Longitude;
    }
    sPoint.Latitude = latTotal / points.Count();
    sPoint.Longitude = lonTotal / points.Count();
    return sPoint;
}

private Dictionary<Link, int> CalculateLinkCount(
    double linkLength,
    List<SpatialPoint> points,
    Network network)
{
    Dictionary<Link, int> result = new Dictionary<Link, int>();
    List<Link> splitedLinks = new List<Link>();

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

SpatialOperator sOperator = new SpatialOperator();
Int32 localSectionNumber = 0; //floored number of links
double localLength = 0; //
double localLengthTotal = 0; // total length of link that calculated in foreach
loop
SpatialPoint startPoint = new SpatialPoint(); //last spatial point
SpatialPoint endPoint = new SpatialPoint(); //last spatial point
#region split links
foreach (Link link in network.Links)
{
    localSectionNumber    =    Convert.ToInt32(Math.Floor(link.Length    /
linkLength));
    //for each link create new links count of localSectionNumber
    for (int i = 1; i <= localSectionNumber; i++)
    {
        //calculate bearing for each point
        //if this is first splited links calculate bearing for first point of main link
        if (i == 1) { startPoint = link.A; }
        //calculate last remaining legth link
        if (i != localSectionNumber) { localLength = linkLength; } else {
localLength = link.Length - localLengthTotal; }
        //find last point with start point, bearing and linkle
        endPoint            =            sOperator.CalculatePoint(startPoint,
sOperator.CalculateBearing(startPoint, link.B), localLength);
        splitedLinks.Add(new Link(startPoint, endPoint));
        //calculate total distance for last link
        localLengthTotal    +=    sOperator.CalculatePlanarDistance(startPoint,
endPoint);
        //equalize startPoint end point of previous link
    }
}

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        startPoint = endPoint;
    }
}
#endregion
#region add splited links to result set
foreach (Link l in splitedLinks)
{
    result.Add(new Link(l.A, l.B), 0);
}
#endregion
#region find closet links for points
double shortestDistance = double.MaxValue;
double shortestDistanceForB = double.MaxValue;
double distance = double.MaxValue;
SpatialPoint closetPoint1 = new SpatialPoint();
SpatialPoint closetPoint2 = new SpatialPoint();
Link fLink = new Link();
foreach (SpatialPoint sp in points)
{
    foreach (Link l in network.Links)
    {
        distance = sOperator.CalculatePlanarDistance(sp, l.A);
        if (distance < shortestDistance)
        {
            shortestDistance = distance;
            closetPoint1 = l.A;
            shortestDistanceForB = double.MaxValue;
            foreach (Link ln in network.Links)
            {

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

if (ln.A == l.A)
{
    distance = sOperator.CalculatePlanarDistance(sp, ln.B);
    if (distance < shortestDistanceForB)
    {
        shortestDistanceForB = distance;
        closetPoint2 = ln.B;
    }
}
}
}
//add finded
fLink.A = closetPoint1;
fLink.B = closetPoint2;
result[fLink]++;
}
#endregion
return result;
}

public Dictionary<Int32, RipleyLocalResultItem>
CalculateCrossRipleyKLWithReferancePoints(
    Int32 intervalCount,
    Int32 randomSampleCount,
    double referancepointDistance,
    List<SpatialPoint> dependentpoints,
    List<PointList> independentpoints,
    Network network)
{

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

    SpatialOperator sOperator = new SpatialOperator();
    double maxRmeter = 0;
    if ((network.CalculateLength() / 8) > 2000)
    { maxRmeter = 2000; }
    else
    { maxRmeter = network.CalculateLength() / 8; }

    List<SpatialPoint> referancePoints =
        new List<SpatialPoint>(
            sOperator.CalculateNetworkReferancePoint(referancepointDistance,
                network)); //network reference points
    # region creaate random points
    List<List<SpatialPoint>>> randomPointsList = new
    List<List<SpatialPoint>>>(); //random network points
    for (int t = 1; t <= randomSampleCount; t++)
    {
        randomPointsList.Add(new
    List<SpatialPoint>(sOperator.CalculateRandomNetworkPoint(dependentpoints.Coun
t, network)));
    }
    #endregion
    # region calculate sample distance list
    List<PointDistance> sampleDistanceList = new List<PointDistance>();
    //sample point distance
    double spNodeDistance = 0; //sample start point distance
    double epNodeDistance = 0; //sample end point distance
    foreach (SpatialPoint spStart in referancePoints)
    {
        //for each point

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        foreach (SpatialPoint spEnd in dependentpoints)
        {
            if (spStart != spEnd)
            {
                //find nearest node for start point
                SpatialPoint spNode = new
                SpatialPoint(sOperator.CalculateNearestPoint(spStart, network.Nodes, out
                spNodeDistance)); ;
                //find nearest node for destination point
                SpatialPoint epNode = new
                SpatialPoint(sOperator.CalculateNearestPoint(spEnd, network.Nodes, out
                epNodeDistance));
                Dictionary<SpatialPoint, Route> result = new
                Dictionary<SpatialPoint, Route>();
                result = sOperator.CalculateNetworkDistance(spNode, network.Links,
                network.Nodes);
                double networkDistance = 0;
                foreach (KeyValuePair<SpatialPoint, Route> kvp in result)
                {
                    if (kvp.Key.Latitude == epNode.Latitude && kvp.Key.Longitude
                    == epNode.Longitude)
                    {
                        networkDistance = kvp.Value.Distance;
                        break;
                    }
                }
                sampleDistanceList.Add(new PointDistance(spStart.Id, spEnd.Id,
                +spNodeDistance + networkDistance + epNodeDistance));
            }
        }
    
```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        else
        {
            sampleDistanceList.Add(new PointDistance(spStart.Id, spEnd.Id, 0));
        }
    }
}

#endregion

# region calculate random point distance list
Dictionary<SpatialPoint, Dictionary<Int32, List<PointDistance>>>
randomDistanceSet = new Dictionary<SpatialPoint, Dictionary<Int32,
List<PointDistance>>>(); //random points distance List

double rspNodeDistance = 0; //random start point distance
double repNodeDistance = 0; //random end point distance
foreach (SpatialPoint rspStart in referencePoints)
{
    Dictionary<Int32, List<PointDistance>> randomDistanceList = new
Dictionary<Int32, List<PointDistance>>(); //random points distance set
    foreach (List<SpatialPoint> randompointsample in randomPointsList)
    {
        List<PointDistance> setDistances = new List<PointDistance>();

        //for each point in random set
        foreach (SpatialPoint rspEnd in randompointsample)
        {
            if (rspStart != rspEnd)
            {
                //find nearest node for start point

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        SpatialPoint      rspNode      =      new
SpatialPoint(sOperator.CalculateNearestPoint(rspStart,      network.Nodes,      out
rspNodeDistance));

        //find nearest node for destination point
        SpatialPoint      repNode      =      new
SpatialPoint(sOperator.CalculateNearestPoint(rspEnd,      network.Nodes,      out
repNodeDistance)); ;

        Dictionary<SpatialPoint,      Route>      resultRandom      =      new
Dictionary<SpatialPoint, Route>();

        resultRandom      =      sOperator.CalculateNetworkDistance(rspNode,
network.Links, network.Nodes);

        double networkRandomDistance = 0;
        foreach      (KeyValuePair<SpatialPoint,      Route>      kvpRandom      in
resultRandom)
        {
            if      (kvpRandom.Key.Latitude      ==      repNode.Latitude      &&
kvpRandom.Key.Longitude == repNode.Longitude)
            {
                networkRandomDistance = kvpRandom.Value.Distance;
                break;
            }
        }

        setDistances.Add(new      PointDistance(rspStart.Id,      rspEnd.Id,
+spNodeDistance + networkRandomDistance + epNodeDistance));
    }
    else
    {
        setDistances.Add(new PointDistance(rspStart.Id, rspEnd.Id, 0));
    }

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

}

randomDistanceList.Add(randomPointsList.IndexOf(randompointsample),
setDistances);

    }
    randomDistanceSet.Add(rspStart, randomDistanceList);

}
#endregion
#region ripley K L calculation
Dictionary<Int32, RipleyLocalResultItem> results = new Dictionary<Int32,
RipleyLocalResultItem>();
    Double    interval    =    Math.Round(Convert.ToDouble(maxRmeter    /
(intervalCount - 1)), 3); //the last interval is the maxRmeter
    double distance = 0;
    double fanm = (network.CalculateLength()/2) / (dependentpoints.Count);
    for (int i = 0; i <= intervalCount; i++)
    {
        distance += interval; //add interval to distance
        RipleyLocalResultItem localResult = new RipleyLocalResultItem();

        foreach (SpatialPoint sPoint in referencePoints)
        {
            RipleyResultItem pointResult = new RipleyResultItem();
            pointResult.K = fanm * sampleDistanceList.FindAll(p => p.Distance <=
distance & p.StartPoint == sPoint.Id).Count;
            pointResult.L = Math.Sqrt(pointResult.K / Math.PI) - distance;
            Int32 randomCount = 0;

```

EK-8 (Devam) Geliştirilen yazılımın SpatialStatistics.cs sınıfı kaynak kodu

```

        foreach (List<SpatialPoint> randomPointSet in randomPointsList)
        {
            randomCount =
randomDistanceSet[sPoint][randomPointsList.IndexOf(randomPointSet)].FindAll(p
=> p.Distance <= distance & p.StartPoint == sPoint.Id).Count;
            pointResult.RandomK.Add(fanm * randomCount);
            pointResult.RandomL.Add(Math.Sqrt(fanm * randomCount /
Math.PI) - distance);
        }
        double shortestDistance = 0;
        foreach (PointList pl in independentpoints)
        {
            sOperator.CalculateNearestPoint(sPoint, pl.Points, out
shortestDistance);
            pointResult.DistanceList.Add(new PointDistance(sPoint.Id, pl.Id,
shortestDistance));
        }
        localResult.PointResults.Add(sPoint, pointResult);
    }
    results.Add(Convert.ToInt32(distance), localResult);
}
#endregion
return results;
}
#endregion
}
}

```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÖZKAN, Korkut
 Uyruğu : T.C.
 Doğum tarihi ve yeri : 27.02.1976 Kızılcahamam
 Medeni hali : Bekar
 Telefon : 0 (312) 412 47 55
 Faks : 0 (312) 412 47 39
 e-mail : korkutozkan@hotmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üniversitesi /Trf. Plan. ve Uyg.	2002
Lisans	Polis Akademisi	1998
Lise	Polis Koleji	1994

İş Deneyimi

Yıl	Yer	Görev
1998-2011	Emniyet Genel Müdürlüğü	Şube Müdürü

Yabancı Dil

İngilizce

Yayınlar

1. Özkan, K. , Çubuk, K., Özkan M., “Turizmin Trafik Kazaları Üzerindeki Mekansal Etkisi”, *3. Uluslar arası Trafik ve Karayolu Güvenliği Kongresi*, Ankara, 627, 2005.