

Tek Örneklem Parametrik Olmayan İstatistiksel Testlerin Excel’de Makrosu

İsmail Hakkı Kara

LİSANS TEZİ

Danışman: Dr.Öğr.Üy. Levent TERLEMEZ

Eskişehir Teknik Üniversitesi Fen Fakültesi İstatistik Bölümü

Haziran/21

# Özet

Bu çalışmada parametrik olmayan tek örneklem hipotez testler için VBA makro dili kullanılarak Excel makroları yazılmıştır. Bu makrolar hem fonksiyon olarak hem de form ara yüzü ile uygulama şeklinde hazırlanmış ve Binom, İşaret, Ki-kare Uygunluk testlerini içermektedir. MS Excel yazılımında bu testler için hazır fonksiyonlar mevcut olmadığından dolayı kullanıcıların zaman kayıplarının önüne geçilmesi hedeflenmiştir. Bu çalışmada yazılan makrolar açık kaynak olarak sunulmuştur. Yapılan testlerin tüm aşamaları açıkça anlatılmış olup örnekler ile pekiştirilmiştir. Aynı şekilde VBA içinde bilinmesi gereken tüm temel kavramlar açıkça anlatılmış ve kullanım örnekleri verilmiştir.

**Anahtar Kelimeler:** VBA; Excel; Tek Örneklem Parametrik Olmayan İstatistiksel Testler.

# Abstract

This study aimed to create macros for non-parametric one sample statistical tests. These macros are created both as a function and as an application with a form interface. Macros are created for Binomial, Sign, Chi-square Fit tests and these macros are created in VBA. Since Excel does not have ready-made functions for these tests, it is aimed not to waste time for users. The macros created in this study are presented as open source. All steps of the tests are clearly explained and supported with examples. Likewise, all the basic concepts that should be known in VBA are explained clearly and usage examples are given.

**Keywords:** VBA; Excel; One Sample Non-Parametric Statistical Test.

İçindekiler

[Özet I](#_Toc75894319)

[Abstract I](#_Toc75894320)

[Birinci Bölüm 1](#_Toc75894321)

[Giriş 1](#_Toc75894322)

[Çalışmanın Önemi 1](#_Toc75894323)

[Çalışmanın Amacı 2](#_Toc75894324)

[İkinci Bölüm 3](#_Toc75894325)

[1. İstatiksel Testler 3](#_Toc75894326)

[1.2. Parametrik Testler 4](#_Toc75894327)

[1.3. Parametrik Olmayan Testler 4](#_Toc75894328)

[1.3.1. Binom Testi 6](#_Toc75894329)

[1.3.2. İşaret Testi 8](#_Toc75894330)

[1.3.3. Ki-kare Uygunluk Testi 11](#_Toc75894331)

[1.3.4. Wilcoxon İşaret Sıralaması Testi 12](#_Toc75894332)

[1.3.5. Kolmogorov Smirnov testi 14](#_Toc75894333)

[1.3.6. Tek Örneklem Dizi Testi 15](#_Toc75894334)

[Üçüncü Bölüm 16](#_Toc75894335)

[1. Programlama 16](#_Toc75894336)

[2. Visual Basic ve Visual Basic For Application 16](#_Toc75894337)

[2.1. Sabitler 16](#_Toc75894338)

[2.2. Değişken Tipleri 17](#_Toc75894339)

[2.3. Değişken Türleri 19](#_Toc75894340)

[2.4. Diziler 20](#_Toc75894341)

[2.5. Operatörler 21](#_Toc75894342)

[2.6. Modüller 22](#_Toc75894343)

[2.7. Yordamlar 22](#_Toc75894344)

[2.8. Koşullar 23](#_Toc75894345)

[2.9. Döngüler 24](#_Toc75894346)

[2.10. Excel Nesneleri 27](#_Toc75894347)

[3. Excel 30](#_Toc75894348)

[3.1. Excel Uygulamasının Genel Yapısı 30](#_Toc75894349)

[3.2. Excel’de Makro Geliştirme 31](#_Toc75894350)

[3.3. Örnek Excel Makrosu 32](#_Toc75894351)

[Dördüncü Bölüm 33](#_Toc75894352)

[1. Binom Testi Makrosu 33](#_Toc75894353)

[1.1. Uygulama (Form Nesnesi Kullanılarak) 33](#_Toc75894354)

[1.2. Fonksiyon (Form Nesnesi Kullanılmadan) 40](#_Toc75894355)

[1.3. Makroyu İçeren Excel Dosyası 41](#_Toc75894356)

[2. İşaret Testi Makrosu 41](#_Toc75894357)

[2.1. Uygulama (Form Nesnesi Kullanılarak) 41](#_Toc75894358)

[2.2. Fonksiyon (Form Nesnesi Kullanılmadan) 47](#_Toc75894359)

[2.3. Makroyu İçeren Excel Dosyası 48](#_Toc75894360)

[3. Ki-Kare Uyum İyiliği Testi Makrosu 48](#_Toc75894361)

[3.1. Uygulama (Form Nesnesi Kullanılarak) 48](#_Toc75894362)

[3.2. Fonksiyon (Form Nesnesi Kullanılmadan) 52](#_Toc75894363)

[3.3. Makroyu İçeren Excel Dosyası 52](#_Toc75894364)

[4. Wilcoxon İşaret Sıralaması Testi 53](#_Toc75894365)

[Ekler 54](#_Toc75894366)

[Şekiller 112](#_Toc75894367)

[Kaynakça 114](#_Toc75894368)

# 

# Birinci Bölüm

## Giriş

Excel Microsoft tarafından yazılan bir tablolama programıdır. Dünya üzerinde birçok işyerinde Excel büyük ölçüde kullanılmaktadır. Detaylı finansal çözümlemelerin, tablolamaların, grafik oluşturmanın yapılabildiği Excel, sunduğu birçok matematiksel ve/veya istatistiksel hazır fonksiyonların yanı sıra Visual Basic makro programlama dili sayesinde kendi türünde şu anda dünyadaki en popüler yazılımlardan biridir. İstatistik, mühendislik ve finansal ihtiyaçları karşılayan fonksiyonlara sahiptir. Ancak bu hazır fonksiyonlar birçok iş yükünü azaltsa da bazı araştırmalarda ve/veya işlemlerde mutlak yeterlilik sağlamamaktadır. Bu sebeple sunduğu makro sistemi sayesinde hazır fonksiyonların yetersiz kaldığı durumda istenilen her türlü hesaplama ve işlem Visual Basic ile Excel tarafında makro yazılarak yapılabilmektedir.

Şirketlerin araştırma ve geliştirme bölümlerindeki çalışmalarda, reklam sektöründe, üretilen üründe yapılacak yeniliklerde ve/veya yapılması gereken yeniliklerde, ekonomik, finansal işlemlerde ve daha birçok yerde istatistiğin ve istatistikçinin önemi büyüktür. İstatistikçi, araştırmasında öncelikle gerekli verileri toplamaktadır. Bu veriler ile gerekli testleri yapıp, testlerin verdiği cevapları yorumlayarak sonuca ulaşmaktadır. Tüm dünyada insanlar için zamanın önemi çok büyük olduğundan bir istatistikçi de yapacağı testleri el ile yapmak yerine bunun için tasarlanmış paket programları kullanmaktadır. Veri toplama, derleme gibi işlemler yine istatistikçi için zaman alan bir durum olduğundan bunun için de birçok program ve sağlayıcı bulunmaktadır. Excel kullanımında sağlayıcıdan alınan veriler kolaylıkla içeri aktarılabilmekte ve gerekli testler için kullanılabilmektedir.

## Çalışmanın Önemi

İstatistikçi için verilerin normal dağılması oldukça önemli bir durumdur. İstatiksel testleri parametrik ve parametrik olmayan testler olarak ikiye ayırabiliriz. Yapılacak testi seçerken gerekli varsayımların sağladığı takdirde parametrik testlerin seçilmesi daha duyarlı bir sonuca ulaşılmasını sağlar. Ancak varsayımların sağlanmadığı durumda parametrik test uygulanamaz. Excel’de parametrik testler için hazır fonksiyonlar bulunmaktadır. Ancak parametrik olmayan testler için hazır fonksiyon bulunmamaktadır. Yapılacak araştırmada alternatif bir test yapılmak istendiğinde veya veriler gerekli varsayımları sağlamadığında parametrik olmayan testler uygulanır. Bu testler için Excel’de hazır fonksiyon bulunmamaktadır. Araştırmacı bunun için alternatif bir paket program kullanabilir veya Excel’de yapmak istediği test için gerekli makroyu yazabilir. Ancak bu yukarıda da belirtildiği üzere araştırmacının zamanını alacaktır. Bu yüzden zaman her anlamda insanlar için en önemli kısıttır.

## Çalışmanın Amacı

Konu parametrik olmayan testler olduğunda bunun için Excel’de hazır fonksiyon bulunmadığı yukarıda belirtilmişti. Bu çalışmada tek örneklem parametrik olmayan testler için Excel’de makrolar yazılacaktır. Bu makrolar sayesinde araştırmacı tek örneklem parametrik olmayan bir test yapmak istediğinde bunun için makro yazmak ile veya alternatif bir paket program aramakla zaman kaybetmeyecektir. Bu çalışma sırasında gerekli tüm bilgiler açıklanacaktır. Gerekli olmayan alt başlıklar kısaca açıklanacaktır.

# İkinci Bölüm

## İstatiksel Testler

İstatiksel testler hipotez testleri adı altında görülür. Bu kavram uygulanan test sonucunda başlangıçta ortaya atılan hipotezlerin doğru veya yanlış olup olmadığını sınamaktadır. Araştırmacı için doğru test seçimi en önemli unsurlardan biridir. Çoğu zaman yapılan araştırmada kullanılabilecek birden fazla test bulunmaktadır. Bu testlerden bazıları diğer testlere göre araştırmaya bağlı olarak da daha duyarlı sonuçlar vermektedir. İstatiksel testin seçiminde birçok etken vardır. Bu etkenlere bağlı olarak başta yapılacak testten önce, parametrik testlerin mi yoksa parametrik olmayan testlerin mi daha uygun olacağıdır. Yapılacak araştırmada parametrik olmayan testler için bir önkoşul gerekmezken, parametrik olmayan testler için bazı varsayımlar söz konusudur. Bu nedenle bir araştırmada parametrik test yapılabiliyorken, parametrik olmayan test de yapılabilmektedir. Ancak bu durum daha az duyarlı sonuçlar verecektir. Tersi ise geçerli bir durum değildir. Yani araştırmada parametrik olmayan test kullanılabilmesi, parametrik testlerinde kullanılabileceği anlamına gelmez. Yukarıda da belirtildiği üzere söz konusu varsayımların sağlanması gerekmektedir.

Yapılacak testin parametrik bir test mi yoksa parametrik olmayan bir test mi olacağına karar verdikten sonra hangi testin seçileceğine karar verilmelidir. Bunun için aşağıdaki maddeler veriler için sınanmalıdır.

* **Değişken yapısı** 
  + Bağımlı
  + Bağımsız
* **Değişken tipi** 
  + Nitel
  + Nicel
* **Değişken Sayısı**
  + Tek grup
  + Çift grup
  + 3 veya daha fazla grup

### 1.2. Parametrik Testler

Parametrik testlerin kullanılabilmesi için verilerin taşıması gereken bazı özellikler vardır.

1. Anakütleden örnekleme kurallarına uygun olarak birimlerin örnekleme seçilmiş olmaları gerekir. Örnekleme için seçilen her birim diğerinden bağımsız olarak seçilmelidir. Bir birimin örnekleme seçilmesi diğerinin de seçilme şansını olumlu veya olumsuz bir şekilde etkilememesi gerekir.
2. Anakütlenin mutlaka normal dağılıyor olması ya da bu varsayımın geçerli olması gerekmektedir.
3. İki anakütlenin karşılaştırılması ya da bunlardan çekilen örneklemlerin karşılaştırılması söz konusu olduğunda bu anakütlelerin varyanslarının homojen olması gerekmektedir.
4. Parametrik bir test yapabilmek için verilerin en azından aralıklı bir ölçekle ölçümlenmiş olması gerekir. Bir değişkenin birimlerindeki durumu belirlemek üzere kullanılan ölçeklerden sıralama ölçeği, parametrik testi uygulamak için elverişli değerleri sağlayamaz.
5. Parametrik bir test olan varyans analizi yapılacaksa, deney düzenlemesiyle ortaya çıkan sıralardan ve sütunlardan doğan etkilerin doğrusal bileşkesi, anakütle ortalamasını oluşturmalıdır.

### 1.3. Parametrik Olmayan Testler

Parametrik olmayan testler, anakütle dağılımı nasıl olursa olsun uygulanabilen testlerdir. Bunun yanında parametrik testlerin uygulanamadığı durumlarda uygulanabilen kolay testlerdir. Parametrik olmayan testler sağladıkları kolaylıklar ile geniş bir kullanım alanına sahiptir.

1. Parametrik olmayan test için anakütlenin normal olması veya böyle bir varsayımda bulunulması gibi bir zorunluluk yoktur. Ayrıca anakütlenin nasıl dağıldığını bilmek veya buna dair bir varsayımda bulunma gereği bile yoktur. Çünkü parametrik olmayan testler dağılıma göre olmayan testlerdir.
2. Parametrik olmayan testten elde edilen olasılıklar tam olasılıklardır. Zira ilgilenilen anakütle çok küçüktür ve olasılık ifadeleri bu küçük anakütleye aittir.
3. Parametrik olmayan test uygulayabilmek için örneklem büyüklüğünün önemi yoktur. Örneklem hacmi (n), 6 ‘dan küçük bile olsa kullanılabilecek parametrik olmayan test vardır.
4. Parametrik olmayan testler eldeki verilerin durumu ne olursa olsun uygulanabilir.
5. Parametrik olmayan testler birkaç farklı yapıdaki anakütleden çekilen örneklemlerde uygulanabilir. Bu anakütleler için gerçek bilgilere ve varsayımlara ihtiyaç duyulmadığı için anakütlelerin durumu nasıl olursa olsun farklı anakütlelere ait örneklemlere ilişkin istatistikler hesaplanabilir.
6. Parametrik testlerin ilgili değişkenin sürekli bir değişken olduğu varsayımı, parametrik olmayan testler içinde geçerlidir. Bu varsayım her iki test türü içinde geçerlidir.
7. Örneklem birim sayısı arttıkça, parametrik olmayan testin gücüde aynı oranda artar.

Tek Örneklem Parametrik Olmayan İstatiksel Testler

* Binom testi
* İşaret testi
* Ki-kare uygunluk testi
* Wilxocon işaretlinmiş sıra testi
* Kolmogrov-Smirnov testi
* Tek örneklem dizi testi

### 1.3.1. Binom Testi

Belli bir olayın π oranında gözlemlendiği bir anakütleden, n birimlik bir rassal örneklem çekildiğinde, bu örneklemde de aynı olaya ait gözlenen p oranının; anakütle düzeyinde olup olmadığı Binom testi ile test edilir.

Binom testi, gözlenen değerlerin yalnızca iki olası sonuca sahip olduğu durumlarda kullanılır. Gözlenen her birimde olası sonuçlardan yalnızca biri gerçekleşir. Hacmi n olan ve binom dağılımı gösteren bir örneklemde ortaya çıkabilecek sonuçlardan birisi r adet gözlenmişse, diğeri n-r adet gözlenmiş olur. Böylesi denemelerde her olayın ortaya çıkış olasılığı aşağıdaki gibidir.



**Varsayımlar**

* Ölçmenin sınıflayıcı ölçekte olduğu ve yalnızca iki olası sonucun olduğu durumlarda kullanılır.
* Veri n kez tekrar eden Bernoulli denemelerinin sonuçlarından oluşur.
* Tekrarlanan denemeler birbirinden bağımsızdır.
* Olası iki sonucun olasılıkları denemeden denemeye değişmez, sabittir.
* Örneklem hacminin, 25’e eşit veya küçük olduğu durumlarda Binom tablosu kullanılır.
* Örneklem hacminin, 25’den büyük olduğu durumlarda npq çarpımları da 9’dan büyük veya eşit ise süreklilik düzeltmesi kullanılır ve z tablosu ile karara varılır.
  + - * Süreklilik düzeltmesi ile yeni dağılım:

**Süreklilik Düzeltmesi**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |

**Hipotezler**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

#### Örnek

Annenin yaşının ileri olmasının, doğacak bebeğin cinsiyetini etkilediği ve ileri yaşta anne olanların daha az kız çocuk sahibi olduğu biçimindeki öngörüyü, irdelemek üzere yapılan araştırmada; yaşları 35’ in üstünde olan 20 anne adayı izlenmiştir ve bunlardan doğan 20 bebeğin 7’ sinin kız olduğu görülmüştür. Gerçekten de ileri yaşta anne olanlar daha az kız çocuk sahibi mi olmaktadır? Bu öngörüyü %95 güven düzeyinde test edelim.

**Çözüm:**

Hipotezler:

Annenin yaşının doğacak çocuğun cinsiyeti üzerinde belirleyici bir etkisi bulunmamaktadır.

İleri yaşlardaki annelerden doğan çocuklardan kızların oranı (p), erkeklerin oranından (q) küçüktür. Aradaki fark tesadüfle açıklanamayacak kadar fazladır.

İşlem ve Karar:

.

.

.

Bu olasılık değeri anlamlılık düzeyi değeri olan ‘den küçük olduğundan H0 hipotezi reddedilemez. Annenin yaşının doğacak çocuğun cinsiyeti üzerinde belirleyici bir etkisi bulunmamaktadır. Aradaki fark %95 güven ile istatistiksel olarak anlamlı değildir.

### 1.3.2. İşaret Testi

İşaret testi, değişken değerlerinin medyana göre konumlarına dayanmaktadır. Anakütle ortalaması için yapılan anlamlılık testinin parametrik olmayan karşılığıdır. Anakütle ortalaması üzerinde kurulmuş hipotezlerin test edilmesinde kullanılır. Medyandan küçük olanlar “–” işareti alırken medyandan büyük olan gözlem değerleri “+” işareti almaktadır. Eğer medyana eşit değerler varsa, o değerler örneklemden çıkarılır. Örneklem hacminin 25’e eşit ve küçük olduğu durumlarda binom tablosundan yararlanılarak hipotezler test edilir.

Ancak örneklem hacminin 25’den büyük olduğu durumlarda örnekleme dağılımı normale yaklaşacağından süreklilik düzeltmesi uygulanır ve normal dağılım tablosu ile hipotezler test edilir. Özellikle p ve q olasılık değerleri arasındaki uzaklık azaldıkça normal dağılıma yaklaşım daha da mükemmel olur.

**Süreklilik Düzeltmesi**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |

**Hipotezler**

#### Örnek

Bir A dersini alan öğrencilerin yıl içi başarı medyanı 70 puan olmuştur. Bu öğrencilerin bazıları bu dersi yaz döneminde de almışlar ve notları aşağıdaki tabloda verilmiştir.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 30 | 35 | 42 | 50 | 55 | 58 | 64 | 70 | 70 | 72 | 80 | 86 | 88 |

Bu verilere bakarak, öğrencilerin yaz dönemindeki başarılarının yıl içindekinden daha iyi olduğunu, medyanın 64 olduğu, dolayısıyla 13 birimli bu örneklemin medyanının ’ten anlamlı bir şekilde düşük olduğu söylenebilir mi?

**Çözüm:**

Hipotezler:

Öğrencilerin yıl içindeki aldıkları notların dağılımı ile yaz döneminde aldıkları notların dağılımı arasında anlamlı bir fark yoktur.

Öğrencilerin yaz dönemindeki puanları yıl içindekinden düşüktür.

İşlem ve Karar:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 30 | 35 | 42 | 50 | 55 | 58 | 64 |

|  |  |  |  |
| --- | --- | --- | --- |
| 72 | 80 | 86 | 88 |

olduğundan ile işlemlere devam edilecektir.

.

.

.

Bu olasılık değeri anlamlılık düzeyi değeri olan ‘den küçük olduğundan H0 hipotezi reddedilemez. Yaz döneminde alınan notlar ile yıl içi alınan notlar farklı değildir. Aradaki fark %95 güven ile istatistiksel olarak anlamlı değildir.

### 1.3.3. Ki-kare Uygunluk Testi

K-kare hesaplaması, gözlenen frekanslar (Gi) ve beklenen frekanslar (Bi) arasındaki farkların karelerine dayanır. Gözlenen frekansla beklenen frekanslar arsında uygunluk olup olmadığını ortaya koyan testtir. Yokluk hipotezi olan H0, gözlenen değerler frekansları ile beklenen değerler frekanslarının birbirine eşit olduğunu, aradaki farkın tesadüf ile açıklanabilecek kadar küçük olduğunu savunur. Hipotezin testi için ki-kare hesap değeri bulunur ve tablo değeri ile karşılaştırılarak sonuca varılır.

**Ki-kare Hesap Değeri**

**Hipotezler**

|  |
| --- |
|  |
|  |

#### Örnek

Bir mağazanın bir yıl içerisindeki aylık TV satışları aşağıdaki tabloda verilmiştir.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 18 | 19 | 25 | 24 | 20 | 18 | 16 | 22 | 30 | 35 | 28 | 21 |

Verilere göre aylık TV satışları arasında anlamlı bir fark olmadığı söylenebilir mi?

**Çözüm:**

Hipotezler:

Aylık TV satışları arasında bir fark yoktur. Gözlenen frekanslar ile beklenen frekanslar arasında anlamlı bir fark yoktur.

Aylık TV satışları birbirinden farklıdır. Gözlenen frekansalar ile beklenen frekanslar arasında bir fark vardır.

İşlem ve Karar:

Mağazanın aylık TV satışları arasında %95 güven ile anlamlı bir fark olmadığı söylenebilir.

### 1.3.4. Wilcoxon İşaret Sıralaması Testi

Parametre olarak medyanı dayanak alan bir konum testidir. *M* medyanına sahip bir örneklemin, *M0* medyanlı bir anakütleden çekilmiş rassal bir örneklem olup olmadığını test etmede kullanılır.

Bir örneklemdeki gözlem değerleri (*xi*) ile anakütle medyanı (*M0*) arasındaki farklar *di* olarak tanımlanır. Bu farklar negatif (*-di*) ve pozitif (*+di*) değerler olarak belirlenir. Gözlem değerleri ile anakütle medyanı arasındaki bu farkların işaretlerine bakılmaksızın büyüklük sırasına konulur. Daha sonra en küçüğünden başlayarak sıra numaraları verilir. Numaralandırıldıktan sonra (-) ve (+) işaretleri de konularak ayrı ayrı toplamları alınır ve bu değerler T- ve T+ olarak adlandırılır. Elde edilen bu T değerlerinden küçük olanının tesadüfe bağlı olup olmadığı araştırılır ve hipotezin reddine ve/veya kabulüne karar verilir.

T- ve T+ değerlerinin H0 hipotezinin reddine yol açabilecek kadar aykırı bir düzeyde olup olmadığı hakkındaki karar, küçük örneklemler () için başka, büyük örneklemler () için başka tekniklerle ele alınır.

Küçük örneklemlerde (), küçük olan T değeri, T tablo değeri ile karşılaştırılır ve hesap değeri tablo değerinden küçükse H0 hipotezi reddedilir.

Büyük örneklemlerde () ise dağılım normale yaklaşacağından T- ve T+ değerlerinin beklenen değerleri ve varyansları birbirine eşit olur ve aşağıdaki gibi hesaplanır.



Buradan hesaplanan herhangi bir T- veya T+ değerinin, beklenen T değerine uzaklığı standardize edilerek, bir z dönüştürümü yapılır. Yapılan dönüşüm ile elde edilen z değeri α ile karşılaştırılarak H0 hipotezinin reddine veya kabulüne karar verilir.

**Hipotezler**

### 1.3.5. Kolmogorov Smirnov testi

Kolmogrov Smirnov testi uyum iyiliği testlerinden bir tanesidir. Testin temeli, gözlenen frekanslar ile beklenen frekansların birbirine ne düzeyde benzediğine dayanır. Ancak burada gözlenen ve beklenen frekans yerine, kümülatif (birikimli) frekansların dağılışının birbirine benzeşimi araştırılır.

H0 hipotezi ile varlığı kabul edilen birikimli dağılım fonksiyonu F0(x) olarak gösterilir. H1 hipotezi ise bu frekansların, en azından birinin tesadüf ile açıklanamayacak kadar büyük olduğu biçimindedir.

* Birikimli frekanslar:   
  + - k: x’e eşit veya küçük gözlenen frekans
* + - k: x’e eşit veya küçük gözlenen frekans

**Hipotezler**

|  |
| --- |
|  |
|  |

### 1.3.6. Tek Örneklem Dizi Testi

Bir anakütleden çekilen örneklemin rasgele bir örneklem olması, gözlem değerlerinin dizilişinin rasgele olması, istatiksel olarak gerekli bir husustur. Bu durum da yine istatiksel testlerle kontrol edilir.

Örneklem genişliğinin n olduğu ve iki olası sonucun olduğu bir deneyde, olası sonuçlardan biri n1 diğeri n2 kez gerçekleşiyor olsun. Bu iki olası sonuçların toplamı örneklem genişliğini (deney sayısını) vermektedir. Diziler testi, bu deneylerin sonucunda, olası sonuçların tesadüfe bağlanıp bağlanmayacağını, örnekleme dağılımı içerisinde ilgilenilen sonucun hangi olasılıkla gözlenen miktarı H0 hipotezi altında alabileceğinin belirlenmesinde kullanılır. Bu amaçla ‘*Diziler Tablosu*’ geliştirilmiştir.

Eğer n1 ve/veya n2 gözlemleri 20 den fazla ise örnekleme dağılımı normale yaklaşacağından aşağıdaki formül ile z dönüşümü yapılır ve hipotez testi için z tablosu kullanılır.

# Üçüncü Bölüm

## Programlama

Programlama, bilgisayarın ve elektronik cihazların donanıma nasıl davranacağını anlatan, cihazdan ne istendiğini cihaza adım adım anlatmayı sağlayan bir süreçtir. Programlama yapılabilmesi için öncelikle bir geliştirme ortamına/derleyiciye (IDE) ve burada kullanılacak programlama diline ihtiyaç vardır.

Günümüzde birçok programlama dili ve derleyici vardır. Bazı diller çok yönlü kullanılıyor olmasına karşın bazı diller belli başlı uygulamalar için standartlaştırılmış hale gelmiştir. Bunlardan biri de Microsoft’un geliştirdiği Office uygulamaları için makro yazımında kullanılan Visual Basic For Application (VBA) dilidir.

Visual Basic başlı başına bir programlama/yazılım dili olmasına karşın, Office uygulamaları için (Excel, Word, Access, vb) standartlaştırılmış ve Visaul Basic For Application olarak Microsoft tarafından sunulmuştur. Bu sayede Office uygulamaları için istenilen özelik ve/veya işlem uygulamada bulunmuyor ise kullanıcı VBA ile makro geliştirip ihtiyaç duyduğu özelliği ve/veya işlemi üretebilmektedir.

## Visual Basic ve Visual Basic For Application

Visual Basic, Microsoft tarafından geliştirilmiş, olay yönlendirmeli, üst seviye, nesne tabanlı ve görsel bir programlama dilidir. Öğrenilmesi ve kullanılması da oldukça kolay olan bu dil ile her türlü masaüstü uygulaması oluşturulabilir.

Visual Basic For Application, Visual Basic programlama dilinin standardize edilmiş bir alt modülü gibidir. Tüm Office uygulamalarında gömülü olarak bulunmaktadır. Kullanıcı makro yazarak istenilen işlemi uygulamanın bir özelliğiymiş gibi kalıcı olarak uygulamaya entegre etmiş olur.

### Sabitler

#### Değişmeyen Sabitler

Program içinde programlayıcı tarafından değişkenler ile tanımlanır.

* Sayısal Sabitler
  + initialValue = 1
  + default\_sampleSize = 10
  + mean = 1.4
* Tarihsel Sabitler
  + date = #22/04/1994#
* Metinsel Sabitler
  + testName = “Binom”

#### Sembolik Sabitler

* Const pi = 3.14
* Const testName = “Binom”

#### VB Sabitleri

* vbTab
* vbRed
* vbCrLf

#### Excel VBA Sabitleri

* xlColumn
* xlSaveChanges

### Değişken Tipleri

#### İnteger

* Yalnızca tam sayılar için tanımlıdır.
* Varsayılan değeri 0’dır.
* Hafızada kapladığı alan 2 bayttır.
* -32.768 ile 32.767 arasında değer alabilir.
* Tipik kullanım örneği: *Dim scale\_of\_Sample As Integer*

#### Long

* Yalnızca tam sayılar için tanımlıdır.
* Varsayılan değeri 0’dır.
* Hafızada kapladığı alan 4 bayttır.
* -2.147.483.648 ile 2.147.438.647 arasında değer alabilir.
* Tipik kullanım örneği: *Dim scale\_of\_Universe As Long*

#### Single

* Tek duyarlıklı ondalık sayılar için tanımlıdır.
* Varsayılan değeri 0’dır.
* Hafızada kapladığı alan 4 bayttır.
* ±1.401298E-45 ve ±3.402823E38 aralıklarında değer alabilir.
* Tipik kullanım örneği: *Dim mean As Single*

#### Double

* Çift duyarlıklı ondalık sayılar için tanımlıdır.
* Varsayılan değeri 0’dır.
* Hafızada kapladığı alan 8 bayttır.
* ±4.94065645841247E-324 ve ±1.7976931348623E324 aralıklarında değer alabilir.
* Tipik kullanım örneği: *Dim pValue As Double*

#### Sitring

* Sabit uzunluktaki metinsel ifadeler için tanımlıdır.
* Varsayılan değeri boştur.
* 1 ile 65400 karakterden oluşabilir.
* Tipik kullanım örneği: *Dim testName As Sitring*

#### Currency

* Para birimleri için tanımlıdır.

#### Varyant

* Tanımsız veri tipidir.
* Hafızada kapladığı alan 16 bayttır.

#### Boolean

* Mantıksal veri tipidir.
* Varsayılan değeri “false” dur.
* İki değer alır.
  + true
  + false
* Tipik kullanım örneği: *Dim independent As Boolean*

#### Object

* Nesneler için tanımlıdır.
* Hafızada kapladığı alan 4 bayttır.

#### VBA İçin Tanımlanmış Diğer Değişken Türleri

* Object (Nesne)
* User Defined (Kullanıcı tanımlı)
* Array (Dizi)
* Range (Aralık)

### Değişken Türleri

Değişkenler yazıldıkları yere göre bir kapsama sahiptirler. Yani etkilerini gösterdikleri bir bölgeye sahiptirler. Bir modül fonksiyon ve yordamlardan oluşur. Fonksiyonlar ‘*function*’ sözcüğü ile yordamlar da ‘*sub*’ sözcüğü ile tanımlanırlar. Bu tanımlar ile modül, gerekli olduğu kadar küçük parçaya ayrılmış olur. Değişkenlerimizde ya bu küçük parçaların içinde ya da modül seviyesinde hüküm sürerler. Hatta tüm EXCEL çalışma ortamında hüküm sürmeleri de mümkündür. Bu amaçla değişkenlerin yazıldıkları yer önemlidir. Ancak en az onun kadar kullanılan tanımlama sözcükleri de önemlidir.

#### Yerel Değişken

Yordam veya fonksiyonların içinde tanımlanan değişkenlere yerel değişkenler denir. Yerel değişkenler yalnızca tanımlı oldukları yordam veya fonksiyon içinde tanımlıdır.

#### Modül Düzeyinde Değişkenler

Modülün tanımlamalar bölümünde tanımlanan değişkenlerdir. Burada tanımlanan değişkenler ‘*Private/Dim*’ veya ‘*Public*’ sözcükleri ile tanımlanır.

Tanımlaması ‘*Private/Dim*’ sözcüğü ile yapılmış değişkenler sadece modülün içinde (tüm alt yordamlar ve fonksiyonlar dahil) tanımlıdır ve kullanılabilir.

Tanımlaması ’*Public*’ sözcüğü ile yapılmış değişkenler başka yordamlarda da kullanılabilir.

Tanımlaması ‘*Global*’ sözcüğü ile yapılmış değişkenler tüm modüllerde kullanılabilir.

#### Statik Değişken

Değişkenler yalnızca tanımlı olduğu bölgenin sınırları dahilinde kullanılıyor olsa da bu durumu değiştirmek mümkündür. Tanımlaması ‘*Static*’ sözcüğü ile yapılmış değişkenler bulunduğu bölgenin sınırları ile sınırlı kalmaz ve tüm modüllerde kullanılabilir.

### Diziler

Değişken türü belirlenen değişkenin yanına, parantez içerisinde boyutu yazılarak o değişkenin tek bir değeri değil yazılan boyuttan bir fazlası olacak şekilde bir vektörü temsil ettiği ifade edilir. Parantez içerisine virgül ile ayırmak şartı ile birden fazla boyut yazılması durumunda da değişken bir matrisi ifade eder. Dizilerin ilk terimi 0’dan başlar.

* Tam sayı olarak tanımlanmış 6 terimli bir vektör:
  + Dim series(5) As Integer
* Ondalıklı olarak tanımlanmış 3 boyutlu bir kare matris.
  + Dim table(2,2) As Single

### Operatörler

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Aritmetik** | | **Karşılaştırma** | | **Mantıksal** | |
| Üs Alma | ^ | Eşittir | == | Değil | Not |
| Çarpma | \* | Eşit Değildir | <> | Ve | And |
| Bölme | / | Küçüktür | < | Ya da | Xor |
| Tam Sayı Bölme | \ | Büyüktür | > |  | Eqv |
| Mod Alma | Mod | Küçük Eşittir | <= |  | Imp |
| Toplama | + | Büyük Eşittir | >= |  |  |
| Çıkarma | - |  |  |  |  |
| Metin Birleştirme | & |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **p** | **q** | **Eqv Sonucu** |
| true | true | true |
| false | false | true |
| true | false | false |
| false | true | false |

|  |  |  |
| --- | --- | --- |
| **p** | **q** | **Imp Sonucu** |
| true | true | true |
| true | false | false |
| true | null | null |
| false | true | true |
| false | false | true |
| false | null | true |
| null | true | true |
| null | false | null |
| null | null | null |

### Modüller

VBA ortamında koz yazılabilecek birçok modül vardır. Bu modül sahaları kitabın tamamı, sayfaların her biri veya daha bağımsız şekillerde kod yazmak için oluşturulmuş sahalardır.

#### Workbook Modülü

Bu modül tüm çalışma kitabını ilgilendiren kodları yazabileceğiniz sahayı gösterir. Çalışma kitabı ile ilgili birtakım olaylarda çalışması gereken kodlar var ise kodlar bu bölümdeki ilgili hazır fonksiyonlara yazılır. Örneğin kitap açıldığında yapılacak işlemler için kodlar, bu modül içindeki VBA tarafından tanımlı olan ‘Open’ fonksiyonunun içine yazılmalıdır.

#### Worksheets Modülleri

Çalışma kitabındaki sayfa sayısı kadar worksheet modülü vardır. Her biri bir çalışma sayfasını temsil eder. Bu sayfalardan birini ilgilendiren kodlar, ilgili modüle yazılır. Örneğin bir sayfadan diğerine geçişte bir şeyler yapılması isteniyorsa o sayfaya ait ‘Worksheet’ kısmında ‘Activate’ metoduna ilgili kodlar yazılır.

#### Serbest Modüller

Tüm çalışma ortamı ile ilgili olan ve hiçbir Excel nesnesine tamamen bağlı olmayan kodlar serbest modüllere yazılırlar. Bu modüllere yordamlar yazılır.

### Yordamlar

VBA da iki tip yordam bulunur. Bunlar ‘*sub*’ yordam ve ‘*function*’ yordamdır.

#### Sub Yordam

Sub (alt) yordamlar, ‘*Sub*’ ve ‘*End Sub*’ ifadeleri ile sınırlandırılan komut satırı dizisidir. Bir alt yordam çalışması için çağırıldığında içindeki komut satırındaki işlemler gerçekleştirilir ve program çağırıldığı yerden sonraki alt satırdaki işlemler ile çalışmaya devam eder. Herhangi bir değer döndürmezler. Eğer bir değer döndürmesi isteniyor ise ‘*Return*’ komutu kullanılarak döndürülmesi istenen değer veya değeri taşıyan değişken yazılabilir.

#### Function Yordam

Function (fonksiyon) yordamlar, ‘*Function*’ ve ‘*End Function*’ ifadeleri ile sınırlandırılan komut satırı dizisidir. Bir fonksiyon yordam çalışması için çağırıldığında içindeki komut satırındaki işlemler gerçekleştirilir, çağrıldığı yere değer döndürür ve program çağırıldığı yerden sonraki alt satırdaki işlemler ile çalışmaya devam eder. Döndürülecek değer, fonksiyon yordamın ismi ile aynı isme sahip olan bir değişken ile yapılır.

### Koşullar

Programlama dendiğinde hemen hemen her programlama dilinde yapısı değişmeksizin, yalnızca yazımı değişen iki tip koşul (sınama) yapısı vardır. Bunlardan biri ‘*If – Else*’ yapısı ve diğeri de ‘*Select Case*’ yapısıdır. Bazı programlama dillerinde ‘*Select Case*’ yapısı ‘*Switch Case*’ ismini almaktadır.

#### If – Else

Bir koşulu kontrol etmek üzere ‘*If* ‘ (eğer) ifadesi kullanılarak koşul sağlanıyor ise ‘*If*’ ve ‘*End If*’ ifadeleri ile sınırlandırılan kod bloğundaki işlemler gerçekleşir. Eğer koşul sağlanmıyor ise bu kod bloğu derleyici tarafından atlanır. Tipik kullanım senaryolarından diğer ikisi de ‘*Else*’ ve ‘*Else If* ’ yapısıdır. İlk sınama ‘*If* ‘ (eğer) ifadesi kullanılarak yazıldıktan sonra ‘*Else If* ’ (eğer değilse) ifadesi ile istenen alternatif koşullar (sınırlama bulunmaksızın) sınanabilir ve ardından bu koşulların hiçbirinin sağlanmadığı durumda yapılacak işlemler için ‘*Else*’ ifadesi kullanılır.

Örnek kullanım aşağıdaki gibidir.

If p\_value <= 0,05 Then

resultInformation = “Null hypothesis is Rejected”

null\_hypothesis\_rejection = True

Else

resultInformation = “Null hypothesis is Not Rejected”

null\_hypothesis\_rejection = False

End If

#### Select Case

Birden fazla durum söz konusu olduğunda ‘*If – Else*’ yapısındansa ‘*Switch Case*’ yapısını kullanmak daha düzenlidir. Aynı zamanda programcı için zamandan tasarruf sağlayabilir ve hatayı minimuma indirebilir. Bu yapıda istenen sayıda kasa açılır ve sınanan değişken hangi kasaya uygun ise o kasa (komut satırı) çalışır.

Örnek kullanım aşağıdaki gibidir.

Select Case p\_value

Case Is <= 0,05

resultInformation = “Null hypothesis is Rejected”

null\_hypothesis\_rejection = True

Case Is > 0,05

resultInformation = “Null hypothesis is Not Rejected”

null\_hypothesis\_rejection = False

Case else

appInformation = “p value not defined”

End Select

### Döngüler

Programlama dendiğinde hemen hemen her programlama dilinde yapısı değişmeksizin, yalnızca yazımı değişen döngü yapıları vardır. Bir değişken, tanımlanan başlangıç değerinden istenen değere kadar, istenen miktarda değişecek biçimde bir döngü oluşturulur. Döngü yapısı sınırları içerisindeki işlemler bu döngünün her adımında çalıştırılır.

#### For Next Döngüsü

Bir değişkenin tanımlanan başlangıç değerinden istenen adımda değişmesi ile istenen değere ulaşana kadarki her adımda istenen işlemlerin yapılmasını sağlayan döngü yapısıdır.

Örnek kullanım aşağıdaki gibidir.

For i=0 To n Step 1

Y = Y + X(i)

Next i

Eğer döngüden döngü bitmeden çıkılmak istenirse ‘*Exit For*’ komutu kullanılır.

#### Do Loop Döngüsü

For Next döngüsü ile yapılan her türlü işlem burada da yapılabilir. Koşul yapılarında olduğu gibi burada da yapılmak istenen işlem iki döngü yapısı ile de yapılabilir. Ancak kullanım yerine ve ihtiyaca göre en uygun olanın seçilmesi programcı için zamandan tasarruf sağlayabilir ve hatayı minimuma indirebilir. Bu döngünün 4 farklı kullanım çeşidi vardır.

##### While

Bu yapı koşul sağlandığı sürece tekrar eder. Koşulun sağlanmaması durumunda döngü biter ve derleyici döngüden sonraki alt satıra geçerek işlemlere devam eder. While kullanımı iki çeşittir.

Eğer ‘*While*’ sözcüğü başta kullanılırsa koşul test edilir ve sağlanıyor ise komutlar çalıştırılır. Koşul sağlanmıyorsa döngüye girilmez. Örnek kullanım aşağıdaki gibidir.

Do While i == n

Y = Y + X(i)

i = i+1

Loop

Eğer ‘*While*’ sözcüğü sonda kullanılırsa koşul test edilmeksizin en az bir kez içeri girilir. Ardından koşul sağlanıyor ise döngü devam eder.

Do

Y = Y + X(i)

i = i+1

Loop While i == n

##### Until

Bu yapı koşul sağlanmadığı sürece tekrar eder. Koşulun sağlanması durumunda döngü biter ve derleyici döngüden sonraki alt satıra geçerek işlemlere devam eder. Until kullanımı iki çeşittir.

Eğer ‘*Until*’ sözcüğü başta kullanılırsa koşul test edilir ve sağlanmıyor ise komutlar çalıştırılır. Koşul sağlanıyorsa döngüye girilmez. Örnek kullanım aşağıdaki gibidir.

Do Until i <> n

Y = Y + X(i)

i = i+1

Loop

Eğer ‘*Until*’ sözcüğü sonda kullanılırsa koşul test edilmeksizin en az bir kez içeri girilir. Ardından koşul sağlanmıyor ise döngü devam eder. Örnek kullanım aşağıdaki gibidir.

Do

Y = Y + X(i)

i = i+1

Loop Until i <> n

Eğer döngü bitmeden döngüden çıkılmak istenirse ‘*Exit Loop*’ komutu kullanılır.

### Excel Nesneleri

Excel gibi ortamlarda temel programlama materyali nesnelerdir. Tüm kodlar bu nesnelerin metotlarına, özelliklerine ve olaylarına yöneliktir. Bu nesneler birbirinin içerisinde olabilir. Her nesne için alt nesneler, metotlar, özellikler ve olaylar bulunur. Bu tip kullanımlarda nesnelerin içerdiği nesneleri çağırmak ve/veya kullanmak için nokta kullanılır.

Excel ortamında bir Excel nesne modelinden bahsetmek mümkündür. En tepede ‘*Workbook*’ nesnesi vardır ve Excel çalışma kitabını temsil eder. Bunun altında çalışma kitabını meydana getiren ‘*Worksheets*’ koleksiyonu ve bu koleksiyonu meydana getiren ‘*Worksheet*’, yani sayfaların her biri bulunur. Bu sayfaları ise ‘*Range*’ (aralık) ve ‘*Cells*’ (hücreler) meydana getirir.

#### Workbooks Nesnesi

Açık olan tüm çalışma kitaplarını temsil eder.

* **Workbooks.count:** Açık çalışma kitaplarının sayısını verir.
* **Workbooks(i).close:** i’ inci çalışma kitabını kapatır.
* **Activeworkbok:** Çalışma kitaplarından aktif olarak açık olan ve penceresi en üstte olan nesneyi temsil eder.

#### Worksheets Nesnesi

Çalışma kitabındaki sayfaların koleksiyonunu temsil eder. Excel çalışma kitabında sayfanın görünen adı aynı zamanda ‘*activesheet*’ olarak adlandırılır.

* **Worksheets.count:** Sayfa sayısını verir.
* **Worksheet(i):** Çalışma kitabındaki i’ inci sayfayı temsil eder.

#### Range Nesnesi

Range nesnesi, bir hücre (cell), bir sıra (row), bir sütun (column) ya da daha fazla bitişik hücre bloklarını temsil eder. Çalışma kitabındaki sayfalara veya sayfadaki istenilen aralığa ulaşmak mümkündür.

* **Worksheets(“Sayfa1”).Range(“A1:C10”).select:** Birinci satır ve birinci sütun ile onuncu satır ve üçüncü sütun arasındaki tüm hücreleri seçer.
* **Worksheets(“Sayfa1”).Range(“A1”).Value:** Birinci satır ve birinci sütunda yer alan hücredeki değeri verir.

##### Özellikleri

* Adress
* Borders
  + xlEdgeBottom
  + xlEdgeLeft
  + xlEdgeRight
  + xlEdgeTop
* Cells
* Columns
* Count
* Font
* Formula
* Interior
* Locked
* Range
* Rows
* Value

##### Metotları

* Active
* AddComment
* Clear
* ClearComment
* ClearContent
* ClearFormats
* Copy
* Delete
* Merge
* Select
* Sort
* Unmerge

#### UserForm Nesnesi

Kullanıcı formu nesnesi ile kullanıcı arabirimi oluşturularak bir pencere ya da iletişim kutusu açılır. Gerekli bilgi girişi, çıkışı ve etkileşimi için kullanıcı formu nesnesi ile form tasarımı yapılır. UserForm nesnesi, ‘Command*Button*’ (buton), ‘*Label*’ (etiket), ‘*TextBox*’ (metin kutusu) gibi nesneleri içerir. Bu nesneler sayesinde kullanıcı etkileşimi sağlanır.

##### Özellikleri

* BorderStyle
* Caption
* Count
* Font
* Height
* Item
* Width

##### Metotları

* Add
* Hide
* Show

##### Userform ile Kullanılabilen Diğer Form Nesneleri

* Checkbox
* Combobox
* CommandButton
* Frame
* Image
* Label
* Listbox
* Multipage
* RefEdit
* Scrollbar
* SpinButton
* TabStrip
* Textbox
* ToggleButton

## Excel

Microsoft şirketinin geliştirdiği birçok amaç için kullanılan Word, Access, Excel, PowerPoint, OneNote, Teams, vb uygulamaların tamamına Microsoft Office Uygulamaları denmektedir. Büyük küçük her türlü şirkette, eğitimde, kişisel işlerde hemen hemen her yerde bu uygulamalar kullanılmaktadır.

Matematiksel, istatiksel ve finansal her türlü işlemin ve testin yapılabileceği, veri ve veriler ile çalışmak istenildiğinde ilk akla gelen uygulamaların başında Microsoft Office uygulamalarından Excel yer almaktadır. Basit anlamda bir tablolama uygulaması olmasına karşın, içinde bulunan özellikleri ve makro yazılarak içinde bulunmayan işlemlerin de yapılabilmesi sayesinde Excel, tablolama uygulamasından fazlasıdır.

### Excel Uygulamasının Genel Yapısı

Bir tablolama uygulaması olan Excel, ‘*Formüller*’ sekmesi altında, matematiksel, finansal, istatiksel ve daha birçok hesaplama içerir. Bu sekmenin altındaki her hesaplama aslında uygulama içine entegre edilmiş makrolardır. Var olan makroların yetersiz kaldığı durumlarda, geliştirici özelliği aktif edilip VBA tabanında istenen makrolar yazılabilir.

Excel uygulamasında makro geliştirmenin tek yolu programlamadan geçmemektedir. Bir diğer seçenek ise makro kaydı başlatılarak, kayıt esnasında yapılan işlemlerin sırasıyla Excel tarafından kaydedilmesidir. Daha sonradan bu işlemleri tekrar yapmak için yalnızca makroyu çalıştırmak yeterli olur. Aslında bu seçenek esnasında VBA tabanında bir yapay sinir ağı algoritması çalışır ve sizin için yapılan işlemleri programlar. Ancak her durumda bu seçenek yeterli olmayacaktır. Bu gibi durumlarda geliştiricilerin makro programlaması daha sağlıklı sonuçlar verir.

Excel’de tablolar üzerinden hesaplamalar yapılabildiği gibi bu hesaplamalar ve sonuçları grafik olarak da çizdirilebilir. Bunun yanı sıra ‘*Veri*’ sekmesi altından birden fazla platformdan veri çekilebilir, veriler üzerinden işlemler yapılabilir ve model oluşturulabilir. Dolayısıyla uygulamanın sunduğu tüm bu alt bölümler sayesinde istatiksel ve finansal birçok paket programın sunduğu işlemler burada da yapılabilir.

### Excel’de Makro Geliştirme

Excel dahil birçok Office uygulaması için Visual Basic (VB) programlama dilinin geliştirilmiş ve/veya standardize edilmiş hali olan Visual Basic For Application (VBA) yapılmıştır. Bu sayede istenilen işlemleri yapmak için geliştiriciler makro programlayabilir.

Excel’de makro programlamak için öncelikle geliştirici seçeneğinin aktif edilmesi gerekir.

Geliştirici seçeneği saçıldıktan sonra Excel’de ‘*Geliştirici*’ sekmesi aktif hale gelir. Bu sekme altından makro yazılabilir, yazılan makrolar çağırılabilir, form nesneleri eklenebilir, tasarım modu açılabilir ve bu mod açıkken form nesneleri düzenlenebilir.

Geliştirici sekmesi altında bulunan ‘*Visual Basic*’ butonu ile VBA açılır. VBA açıldığında sol tarafta ‘*Project-VBAProject*’ başlıklı bir pencere bulunur. Bu pencere altında Çalışma kitabı nesnesi sayfalar ve eğer oluşturmuşsak form nesnesi bulunur.

Form nesnesini açmak için aşağıdaki yol izlenebilir.

Form nesnesinin araçlarına ulaşmak için açılan form penceresi aktif olarak seçili iken aşağıdaki yol izlenebilir.

Excel’de boş bir hücreye veya formül kısmına yazılan fonksiyonu çağırıp hesaplamaların ve/veya işlemlerin yapılması isteniyorsa öncelikle makroyu yazabilmek için VBA’da ‘*Project-VBAProject*’ penceresi altında bulunan ‘*Microsoft Excel Objects*’ klasörüne sağ tıklanır ve ‘*Insert*’ menüsü altından ‘*Module*’ seçeneğine tıklanır. Bu işlemin sonucunda ‘*Project-VBAProject*’ penceresi altında ‘*Modules*’ klasörü ve onunda altında ‘*Module1*’ isimli yeni bir modül oluşur.

### Örnek Excel Makrosu

Örnek olarak ekte (EK 1) kullanıcı tarafından seçilen hücrelerin toplamını hesaplayan Excel makrosu verilmiştir.

# Dördüncü Bölüm

## Binom Testi Makrosu

Binom testi makrosu, hem kullanıcı form ara yüzü kullanarak uygulama biçiminde hem de direkt fonksiyon olarak yapılacaktır.

### Uygulama (Form Nesnesi Kullanılarak)

Boş bir Excel dosyası açılıp ‘*Geliştirici*’ sekmesi altında ‘*Visual Basic*’ isimli butona tıklanarak VBA açılır. Ardından bir form nesnesi oluşturulur ve gerekli özellikleri ‘*Properties*’ penceresi altından aşağıdaki gibi değiştirilir.

***Name:*** *BinomForm*

***Caption:*** *Tek Örneklem Binom Testi*

Bir sonraki aşama olarak, Excel’de form ara yüzünün açılmasını sağlayacak fonksiyon yazılmalıdır. Bunun için bir modüle ihtiyaç vardır. VBA’da iken ‘*Insert*’ sekmesi altında ‘*Module*’ isimli butona tıklanarak yeni bir modül oluşturulur. Bu çalışmada modülün ismi, ‘*Properties*’ penceresi altından ‘*Name*’ özelliğine ulaşılarak ‘*CalcWithForm*’ olarak değiştirilmiştir. Ardından oluşturulan yeni modül içerisine ekteki (EK 2) fonksiyon yazılır. Bu fonksiyon sayesinde kullanıcı form ara yüzü açılacak, gerekli girdiler alınacak, hesaplamalar yapılacak ve sonuçlar verilecektir.

#### Form Nesnesi için Tasarım

‘*BinomForm*’ isimli form nesnesine aşağıdaki nesneler eklenerek, bazı özellikleri aşağıda belirtildiği üzere değiştirilmiştir. Bu değişiklikler üzerine form nesnesinin kod penceresi açılarak (‘*View*’ → ‘*Code*’) ilgili nesne için eklerde verilen Sub Yordamlar yazılmıştır.

##### İlgilenilen Durum

İlgilenilen durumun girişi için ‘*ToolBox*’ penceresi altından bir ‘*TextBox*’ nesnesi, bu nesneyi belirtmek amacı ile bir ‘*Label*’ nesnesi ve açıklama için bir ‘*CommandButton*’ nesnesi oluşturulmuştur. Oluşturulan ‘*TextBox*’ nesnesi için ekte (EK 3) verilen kodlar ve ‘*CommandButton*’ nesnesi için ekte (EK 4) verilen kodlar yazılmıştır.

###### TextBox Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *ValueText*

***Text:*** *0*

***ForeColor:*** *&H00808080&*

***Font:*** *Times New Roman / Italic / 11*

###### Label Nesnesinin Değişen Özellikleri

***Name:*** *ValueLabel*

***Caption:*** *İlgilenilen Durum*

###### CommandButton Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *infButton\_value*

***Caption:*** *i*

##### Test Olasılığı

Test olasılığının girişi için ‘*ToolBox*’ penceresi altından bir ‘*TextBox*’ nesnesi, bu nesneyi belirtmek amacı ile bir ‘*Label*’ nesnesi ve açıklama için bir ‘*CommandButton*’ nesnesi oluşturulmuştur. Oluşturulan ‘*TextBox*’ nesnesi için ekte (EK 5) verilen kodlar ve ‘*CommandButton*’ nesnesi için ekte (EK 6) verilen kodlar yazılmıştır.

###### TextBox Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *RprobValueText*

***Text:*** *0,5*

###### Label Nesnesinin Değişen Özellikleri

***Name:*** *RprobLabel*

***Caption:*** *Test Olasılığı*

###### CommandButton Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *infButtonRprob*

***Caption:*** *i*

##### Veri Adresi (Sütun)

Verilerin hangi sütunda yazılı olduğunu öğrenmek için ‘*ToolBox*’ penceresi altından bir ‘*TextBox*’ nesnesi, bu nesneyi belirtmek amacı ile bir ‘*Label*’ nesnesi ve açıklama için bir ‘*CommandButton*’ nesnesi oluşturulmuştur. Oluşturulan ‘*TextBox*’ nesnesi için ekte (EK 7) verilen kodlar ve ‘*CommandButton*’ nesnesi için ekte (EK 8) verilen kodlar yazılmıştır.

###### TextBox Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *ColumnAdressText*

***Text:*** *1*

###### Label Nesnesinin Değişen Özellikleri

***Name:*** *ColumnAdressLabel*

***Caption:*** *Veri Adresi (Sütun)*

###### CommandButton Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *infButton\_ColAdress*

***Caption:*** *i*

##### Başlangıç Satırı

Verilerin hangi satırdan başladığını öğrenmek için ‘*ToolBox*’ penceresi altından bir ‘*TextBox*’ nesnesi, bu nesneyi belirtmek amacı ile bir ‘*Label*’ nesnesi ve açıklama için bir ‘*CommandButton*’ nesnesi oluşturulmuştur. Oluşturulan ‘*TextBox*’ nesnesi için ekte (EK 9) verilen kodlar ve ‘*CommandButton*’ nesnesi için ekte (EK 10) verilen kodlar yazılmıştır.

###### TextBox Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *FirstRowText*

***Text:*** *1*

###### Label Nesnesinin Değişen Özellikleri

***Name:*** *FirstRowLabel*

***Caption:*** *Başlangıç Satırı*

###### CommandButton Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *infButton\_firstRow*

***Caption:*** *i*

##### Anlamlılık Düzeyi

Testin anlamlılık düzeyi için ‘*ToolBox*’ penceresi altından bir ‘*TextBox*’ nesnesi ve bu nesneyi belirtmek amacı ile bir ‘*Label*’ nesnesi oluşturulmuştur. Oluşturulan ‘*TextBox*’ nesnesi için ekte (EK 11) verilen kodlar yazılmıştır.

###### TextBox Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *alfaText*

***Text:*** *1*

###### Label Nesnesinin Değişen Özellikleri

***Name:*** *alfaLabel*

***Caption:*** *Anlamlılık Düzeyi*

##### Alternatif Hipotez

Alternatif hipotezin seçimi için ‘*ToolBox*’ penceresi altından birkaç nesne oluşturmak gerekir. Başlık belirtmek için ‘*Label*’ nesnesi ve ilgili başlığı vurgulamak için bir ‘*Frame*’ nesnesi oluşturulmuştur. Alternatif hipotezin seçimi için üç adet ‘*OptionButton*’ nesnesi ve bunları açıklamak için üç adet ‘*Label*’ nesnesi oluşturulmuştur.

###### Label Nesnelerinin Değişen Özellikleri

***Name:*** *HipotezLabel*

***Caption:*** *Alternatif Hipotez*

***Name:*** *twoTailed*

***Caption:*** *p ≠ 𝜋*

***Name:*** *oneTailed\_neg*

***Caption:*** *p <𝜋*

***Name:*** *oneTailed\_poz*

***Caption:*** *p > 𝜋*

##### Hesapla Butonu

Girdileri derleyip işlem yapmak ve takibinde sonucu döndürmek için bir ‘CommandButton’ nesnesi oluşturulmuştur.

###### CommandButton Nesnesinin Değişen Özellikleri

***Name:*** *CalcButton*

***Caption:*** *Hesapla*

##### Hipotez Sonucu

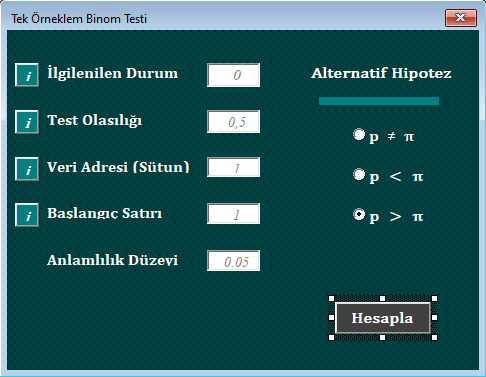
Hipotez sonucunu aktarmak üzere bir ‘*TextBox*’ nesnesi oluşturulmuştur.

###### TextBox Nesnesinin Değişen Özellikleri

***Name:*** *OutText*

***Text:***

##### Tasarımı Tamamlanmış Form Nesnesi (BinomForm)



Şekil 1 (Tek Örneklem Binom Testi Form Arayüzü)

#### Form Nesnesi için Kodlar

Form nesnesi (‘*BinomForm*’) seçiliyken ‘*View*’ sekmesi altından ‘*Code*’ butonu tıklanarak nesnenin kodları girilecek penceresi açılır. Burada, ‘*General / Declarations*’ bölümüne, ekte (EK 12) verildiği üzere gerekli bazı değişken tanımaları yapılmıştır.

Yukarıda belirtilen tanımlamalardan sonra hesaplama ve sonuç döndürme işlemi için ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ yordamında işlemler yapılacaktır. Bu yordamın içine çağırılmak üzere ‘*General’* bölümünde üç adet fonksiyon oluşturulacaktır. Her fonksiyon alternatif hipotezlerden biri için hesaplama yapacak ve sonuç döndürecektir.

##### CalcButton Nesnesinin Click Olay Yordamı

Bu yordam içinde testin olasılık hesabı yapılır ve gerekli hipotez sınanarak sonuç döndürülür. Öncelikle form arayüzündeki TextBox nesnelerinden değerler alınır ve ilgili değişkene aktarılır. Ardından ilgilenilen durum değişkeninin boş bırakılıp bırakılmadığı kontrol edilir. Eğer boş bırakıldıysa kullanıcı bilgilendirilir ve ilgilenilen durum veri setindeki ilk veri olur. Girilen veri sütunundaki satırlar incelenerek boş bir hücreye denk gelene kadar bir döngü oluşturulur ve bu döngü sonucu örneklem hacmi belirlenir. Daha sonra ilgilenilen durum değişkeni ile ilgilenilmeyen durum değişkeninin değerlerini belirlemek amacı ile başka bir döngü kullanılır. Gerekli veriler elde edildiği için test olasılığı form arayüzündeki seçim neticesinde hesaplanır. Bu aşamada test olasılığını hesaplayacak bir fonksiyondan faydalanılır. Test olasılığı hesaplandıktan sonra yine form arayüzünde seçilen hipotez test edilir ve sonuçlar kullanıcıya ‘*MsgBox*’ nesnesi ile verilir. Bu yordama ilişkin kodlar ekte (EK 13) verilmiştir.

##### İlgilenilen Değişkenin Eşit Olması Olasılığını Hesaplayan Fonksiyon

Söz konusu durum için matematiksel gösterim aşağıdaki gibidir.

Ekte (EK 13) görüldüğü üzere ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ olay yordamında alternatif hipotez, ilgilenilen değişkeninin gerçek anakütle değerinden farklı olduğunu savunurken gerekli olasılık hesabını ‘*pCalcNullHyp*’ isimli fonksiyon döndürmektedir. Olasılık hesabı bu fonksiyonda yapılır ancak hipotezin kararına ilişkin sınama yine ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ olay yordamında gerçekleşir. Bu fonksiyona (‘*pCalcNullHyp*’) ilişkin kodlar ekte (EK 14) verilmiştir.

##### İlgilenilen Değişkenin Küçük Olması Olasılığını Hesaplayan Fonksiyon

Söz konusu durum için matematiksel gösterim aşağıdaki gibidir.

Ekte (EK 13) görüldüğü üzere ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ olay yordamında alternatif hipotez, ilgilenilen değişkeninin gerçek anakütle değerinden küçük olduğunu savunurken gerekli olasılık hesabını ‘pCalcIsSmall’ isimli fonksiyon döndürmektedir. Olasılık hesabı bu fonksiyon ile yapılır ancak hipotezin kararına ilişkin sınama yine ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ olay yordamında gerçekleşir. Bu fonksiyona (‘*pCalcIsSmallp*’) ilişkin kodlar ekte (EK 15) verilmiştir.

##### İlgilenilen Değişkenin Büyük Olması Olasılığını Hesaplayan Fonksiyon

Ekte (EK 13) görüldüğü üzere ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ olay yordamında alternatif hipotez, ilgilenilen değişkeninin gerçek anakütle değerinden büyük olduğunu savunurken gerekli olasılık hesabını ‘pCalcIsBig’ isimli fonksiyon döndürmektedir. Olasılık hesabı bu fonksiyon ile yapılır ancak hipotezin kararına ilişkin sınama yine ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ olay yordamında gerçekleşir. Bu fonksiyona (‘*pCalcIsBig*’) ilişkin kodlar ekte (EK 16) verilmiştir.

### Fonksiyon (Form Nesnesi Kullanılmadan)

Excel çalışma sayfasında fonksiyon ismini yazıp veri adresini girerek istenen fonksiyon çalıştırılabilir ve istenen sonuca ulaşılabilir. VBA’da makro yazarak böyle fonksiyonlar oluşturmak için öncelikle ‘*Insert*’ sekmesi altından ‘*Module*’ butonu tıklanarak yeni bir modül oluşturulur. Bu modülün içine yazılan her fonksiyon Excel’de çağırılabilir.

Bu aşamada yeni bir modül oluşturulmuş ve ‘Name’ özelliği ‘CalcWithoutForm’ olarak değiştirilmiştir. Modülün içine ekte (EK 17) verilen ‘*BinomTest*’ isimli fonksiyon yazılmıştır. Bu fonksiyon form Arayüzü kullanılmadığı için gerekli verileri parametreleri ile almaktadır. Boş bırakılan parametreler için varsayılan değer belirlenmişse kullanıcı bilgilendirilerek devam etmesi durumunda varsayılan değerler ile işlemler yapılır. Eğer varsayılan değer belirlenemeyecek olan bir parametre boş bırakılmışsa kullanıcı bilgilendirilir ve değeri girmesi istenir.

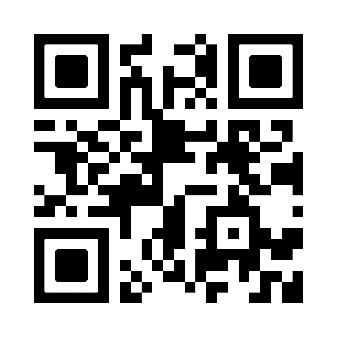
Bu fonksiyon içinde test olasılığı hesaplanmaz. Gerekli veriler parametreler yardımıyla çekilir. Ardından form nesnesi kullanılarak tasarlanan testteki olasılık hesapları için yazılan fonksiyonlara gerekli parametre değerleri gönderilerek olasılık hesaplanır ve sonuç çekilir. Test olasılığı elde edildikten sonra belirlenen hipotez için sınamalar yapılır ve kullanıcı ‘*MsgBox*’ nesnesi ile bilgilendirilir.

Test olasılığını hesaplamak için yeniden bir fonksiyon yazmak veya bu fonksiyonun içinde hesaplamaları yapmak artı bir iş yüküdür. Bu sebeple zaten gerekli hesabı yapan bir fonksiyon varken yeniden bir fonksiyon yazmak sağlıklı bir yaklaşım değildir.

### Makroyu İçeren Excel Dosyası

Binom testi makrosuna ulaşmak için aşağıdaki bağlantıyı kullanabilirsiniz.

<https://github.com/melloBlack/Tez>



## İşaret Testi Makrosu

İşaret testi makrosu, hem kullanıcı form ara yüzü kullanarak uygulama biçiminde hem de direkt fonksiyon olarak yapılacaktır.

### Uygulama (Form Nesnesi Kullanılarak)

Boş bir Excel dosyası açılıp ‘*Geliştirici*’ sekmesi altında ‘*Visual Basic*’ isimli butona tıklanarak VBA açılır. Ardından bir form nesnesi oluşturulur ve gerekli özellikleri ‘*Properties*’ penceresi altından aşağıdaki gibi değiştirilir.

***Name:*** *IsaretForm*

***Caption:*** *Tek Örneklem İşaret Testi*

Daha önce oluşturulan ‘*CalcWithForm*’ isimli modül içerisine ekteki (EK 18) fonksiyon yazılır. Bu fonksiyon sayesinde kullanıcı form ara yüzü açılacaktır. Açılan form üzerinden gerekli girdiler alınacak, hesaplamalar yapılacak ve sonuçlar verilecektir.

#### Form Nesnesi için Tasarım

‘*IsaretForm*’ isimli form nesnesine aşağıdaki nesneler eklenerek, bazı özellikleri aşağıda belirtildiği üzere değiştirilmiştir. Bu değişiklikler üzerine form nesnesinin kod penceresi açılarak (‘*View*’ → ‘*Code*’) ilgili nesne için eklerde verilen Sub Yordamlar yazılmıştır.

##### Veri Adresi (Sütun)

Verilerin hangi sütunda yazılı olduğunu öğrenmek için ‘*ToolBox*’ penceresi altından bir ‘*TextBox*’ nesnesi, bu nesneyi belirtmek amacı ile bir ‘*Label*’ nesnesi ve açıklama için bir ‘*CommandButton*’ nesnesi oluşturulmuştur. Oluşturulan ‘*TextBox*’ nesnesi için ekte (EK 19) verilen kodlar ve ‘*CommandButton*’ nesnesi için ekte (EK 20) verilen kodlar yazılmıştır.

###### TextBox Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *ColumnAdressText*

***Text:*** *1*

###### Label Nesnesinin Değişen Özellikleri

***Name:*** *ColumnAdressLabel*

***Caption:*** *Veri Adresi (Sütun)*

###### CommandButton Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *infButton\_ColumnAdress*

***Caption:*** *i*

##### Başlangıç Satırı

Verilerin hangi satırdan başladığını öğrenmek için ‘*ToolBox*’ penceresi altından bir ‘*TextBox*’ nesnesi, bu nesneyi belirtmek amacı ile bir ‘*Label*’ nesnesi ve açıklama için bir ‘*CommandButton*’ nesnesi oluşturulmuştur. Oluşturulan ‘*TextBox*’ nesnesi için ekte (EK 21) verilen kodlar ve ‘*CommandButton*’ nesnesi için ekte (EK 22) verilen kodlar yazılmıştır.

###### TextBox Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *FirstRowText*

***Text:*** *1*

###### Label Nesnesinin Değişen Özellikleri

***Name:*** *FirstRowLabel*

***Caption:*** *Başlangıç Satırı*

###### CommandButton Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *infButton\_firstData*

***Caption:*** *i*

##### Anakütle Medyanı

Anakütle medyanını öğrenmek için ‘*ToolBox*’ penceresi altından bir ‘*TextBox*’ nesnesi, bu nesneyi belirtmek amacı ile bir ‘*Label*’ nesnesi ve açıklama için bir ‘*CommandButton*’ nesnesi oluşturulmuştur. Oluşturulan ‘*TextBox*’ nesnesi için ekte (EK 23) verilen kodlar ve ‘*CommandButton*’ nesnesi için ekte (EK 24) verilen kodlar yazılmıştır.

###### TextBox Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *AMText*

***Text:*** *-*

###### Label Nesnesinin Değişen Özellikleri

***Name:*** *AM\_label*

***Caption:*** *Anakütle Medyanı*

###### CommandButton Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *infButton\_Median*

***Caption:*** *i*

##### Anlamlılık Düzeyi

Testin anlamlılık düzeyi için ‘*ToolBox*’ penceresi altından bir ‘*TextBox*’ nesnesi ve bu nesneyi belirtmek amacı ile bir ‘*Label*’ nesnesi oluşturulmuştur. Oluşturulan ‘*TextBox*’ nesnesi için ekte (EK 25) verilen kodlar yazılmıştır.

###### TextBox Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *alfaText*

***Text:*** *1*

###### Label Nesnesinin Değişen Özellikleri

***Name:*** *alfaLabel*

***Caption:*** *Anlamlılık Düzeyi*

##### Alternatif Hipotez

Alternatif hipotezin seçimi için ‘*ToolBox*’ penceresi altından birkaç nesne oluşturmak gerekir. Başlık belirtmek için ‘*Label*’ nesnesi ve ilgili başlığı vurgulamak için bir ‘*Frame*’ nesnesi oluşturulmuştur. Alternatif hipotezin seçimi için üç adet ‘*OptionButton*’ nesnesi ve bunları açıklamak için üç adet ‘*Label*’ nesnesi oluşturulmuştur.

###### Label Nesnelerinin Değişen Özellikleri

***Name:*** *HipotezLabel*

***Caption:*** *Alternatif Hipotez*

***Name:*** *Option\_NotEqual*

***Caption:*** *M ≠ M0*

***Name:*** *Option\_LessThen*

***Caption:*** *M <M0*

***Name:*** *Option\_GreaterThan*

***Caption:*** *M >M0*

##### Hesapla Butonu

Girdileri derleyip işlem yapmak ve takibinde sonucu döndürmek için bir ‘CommandButton’ nesnesi oluşturulmuştur.

###### CommandButton Nesnesinin Değişen Özellikleri

***Name:*** *CalcButton*

***Caption:*** *Hesapla*

##### Hipotez Sonucu

Hipotez sonucunu ve örneklem medyanını aktarmak üzere iki ‘*TextBox*’ nesnesi oluşturulmuştur.

###### TextBox Nesnesinin Değişen Özellikleri

***Name:*** *OutText*

***Text:***

###### TextBox Nesnesinin Değişen Özellikleri

***Name:*** *SMText*

***Text:***

##### Tasarımı Tamamlanmış Form Nesnesi (IsaretTestiForm)



Şekil 2 (Tek Örneklem İşaret Testi Form Arayüzü)

#### Form Nesnesi için Kodlar

Hesaplama ve sonuç döndürme işlemi için ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ yordamında işlemler yapılacaktır. Bu yordamın içine çağırılmak üzere ‘*General’* bölümünde üç adet fonksiyon oluşturulacaktır. Her fonksiyon alternatif hipotezlerden biri için hesaplama yapacak ve sonuç döndürecektir.

##### CalcButton Nesnesinin Click Olay Yordamı

Bu yordam içinde testin olasılık hesabı yapılır ve gerekli hipotez sınanarak sonuç döndürülür. Öncelikle form arayüzündeki TextBox nesnelerinden değerler alınır ve ilgili değişkene aktarılır. Eğer boş bırakılan bir “TextBox” nesnesi var ise kullanıcı bilgilendirilir. Girilen veri sütunundaki satırlar incelenerek boş bir hücreye denk gelene kadar bir döngü oluşturulur ve bu döngü sonucu örneklem hacmi belirlenir. Daha sonra verileri küçükten büyüğe sıralamak için iç içe döngü kullanılır. Veriler sıralandıktan sonra anakütle medyanından küçük olan değerlerin adedini bir değişkene ve büyük olan değerlerin adedini bir değişkene atmak için bir döngü içinde sınama yapılır. Bu sınama esnasında anakütle medyanına eşit olan değerler iki değişkeni de arttırmaz. Dolayısıyla testin kuralı gereği anakütle medyanına eşit olan değerler örneklemden çıkartılır. Gerekli veriler elde edildiği için test olasılığı form arayüzündeki seçim neticesinde hesaplanır. Bu aşamada test olasılığını hesaplayacak bir fonksiyondan faydalanılır. Test olasılığı hesaplandıktan sonra yine form arayüzünde seçilen hipotez test edilir ve sonuçlar kullanıcıya ‘*MsgBox*’ nesnesi ile verilir. Bu yordama ilişkin kodlar ekte (EK 26) verilmiştir.

##### Eşit Olmaması Hipotezi Olasılığını Hesaplayan Fonksiyon

Söz konusu durumu sınamak için hesaplanması gereken olasılığın matematiksel gösterimi aşağıdaki gibidir.

Ekte (EK 26) görüldüğü üzere ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ olay yordamında alternatif hipotez, ilgilenilen değişkeninin gerçek anakütle değerinden farklı olduğunu savunurken gerekli olasılık hesabını ‘*pCalcNullHyp*’ isimli fonksiyon döndürmektedir. Olasılık hesabı bu fonksiyonda yapılır ancak hipotezin kararına ilişkin sınama yine ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ olay yordamında gerçekleşir. Bu fonksiyona (‘*pCalcNullHyp*’) ilişkin kodlar ekte (EK 27) verilmiştir.

##### Küçük Olması Hipotezi Olasılığını Hesaplayan Fonksiyon

Söz konusu durumu sınamak için hesaplanması gereken olasılığın matematiksel gösterimi aşağıdaki gibidir.

Ekte (EK 26) görüldüğü üzere ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ olay yordamında alternatif hipotez, ilgilenilen değişkeninin gerçek anakütle değerinden küçük olduğunu savunurken gerekli olasılık hesabını ‘pCalcIsSmall’ isimli fonksiyon döndürmektedir. Olasılık hesabı bu fonksiyon ile yapılır ancak hipotezin kararına ilişkin sınama yine ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ olay yordamında gerçekleşir. Bu fonksiyona (‘*pCalcIsSmallp*’) ilişkin kodlar ekte (EK 28) verilmiştir.

##### Büyük Olması Hipotezi Olasılığını Hesaplayan Fonksiyon

Söz konusu durumu sınamak için hesaplanması gereken olasılığın matematiksel gösterimi aşağıdaki gibidir.

Ekte (EK 26) görüldüğü üzere ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ olay yordamında alternatif hipotez, ilgilenilen değişkeninin gerçek anakütle değerinden büyük olduğunu savunurken gerekli olasılık hesabını ‘pCalcIsBig’ isimli fonksiyon döndürmektedir. Olasılık hesabı bu fonksiyon ile yapılır ancak hipotezin kararına ilişkin sınama yine ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ olay yordamında gerçekleşir. Bu fonksiyona (‘*pCalcIsBig*’) ilişkin kodlar ekte (EK 29) verilmiştir.

### Fonksiyon (Form Nesnesi Kullanılmadan)

Daha önceden oluşturulmuş ve ‘Name’ özelliği ‘CalcWithoutForm’ olarak değiştirilmiş olan modülün içine ekte (EK 30) verilen ‘*IsaretTesti*’ isimli fonksiyon yazılmıştır. Bu fonksiyon form Arayüzü kullanılmadığı için gerekli verileri parametreleri ile almaktadır. Boş bırakılan parametreler için varsayılan değer belirlenmişse kullanıcı bilgilendirilerek devam etmesi durumunda varsayılan değerler ile işlemler yapılır. Eğer varsayılan değer belirlenemeyecek olan bir parametre boş bırakılmışsa kullanıcı bilgilendirilir ve değeri girmesi istenir.

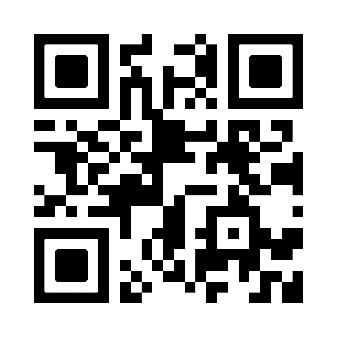
Bu fonksiyon içinde test olasılığı hesaplanmaz. Gerekli veriler parametreler yardımıyla çekilir. Ardından form nesnesi kullanılarak tasarlanan testteki olasılık hesapları için yazılan fonksiyonlara gerekli parametre değerleri gönderilerek olasılık hesaplanır ve sonuç çekilir. Test olasılığı elde edildikten sonra belirlenen hipotez için sınamalar yapılır ve kullanıcı ‘*MsgBox*’ nesnesi ile bilgilendirilir.

Test olasılığını hesaplamak için yeniden bir fonksiyon yazmak veya bu fonksiyonun içinde hesaplamaları yapmak artı bir iş yüküdür. Bu sebeple zaten gerekli hesabı yapan bir fonksiyon varken yeniden bir fonksiyon yazmak sağlıklı bir yaklaşım değildir.

### Makroyu İçeren Excel Dosyası

İşaret testi makrosuna ulaşmak için aşağıdaki bağlantıyı kullanabilirsiniz.

<https://github.com/melloBlack/Tez>



## Ki-Kare Uyum İyiliği Testi Makrosu

Ki-kare uyum iyilği testi makrosu, hem kullanıcı form ara yüzü kullanarak uygulama biçiminde hem de direkt fonksiyon olarak yapılacaktır.

### Uygulama (Form Nesnesi Kullanılarak)

Boş bir Excel dosyası açılıp ‘*Geliştirici*’ sekmesi altında ‘*Visual Basic*’ isimli butona tıklanarak VBA açılır. Ardından bir form nesnesi oluşturulur ve gerekli özellikleri ‘*Properties*’ penceresi altından aşağıdaki gibi değiştirilir.

***Name:*** *KiKareForm*

***Caption:*** *Tek Örneklem Ki-Kare Uyum İyiliği Testi*

Daha önce oluşturulan ‘*CalcWithForm*’ isimli modül içerisine ekteki (EK 31) fonksiyon yazılır. Bu fonksiyon sayesinde kullanıcı form ara yüzü açılacaktır. Açılan form üzerinden gerekli girdiler alınacak, hesaplamalar yapılacak ve sonuçlar verilecektir.

#### Form Nesnesi için Tasarım

‘*KiKareForm*’ isimli form nesnesine aşağıdaki nesneler eklenerek, bazı özellikleri aşağıda belirtildiği üzere değiştirilmiştir. Bu değişiklikler üzerine form nesnesinin kod penceresi açılarak (‘*View*’ → ‘*Code*’) ilgili nesne için eklerde verilen Sub Yordamlar yazılmıştır.

##### Veri Adresi (Sütun)

Verilerin hangi sütunda yazılı olduğunu öğrenmek için ‘*ToolBox*’ penceresi altından bir ‘*TextBox*’ nesnesi, bu nesneyi belirtmek amacı ile bir ‘*Label*’ nesnesi ve açıklama için bir ‘*CommandButton*’ nesnesi oluşturulmuştur. Oluşturulan ‘*TextBox*’ nesnesi için ekte (EK 32) verilen kodlar ve ‘*CommandButton*’ nesnesi için ekte (EK 33) verilen kodlar yazılmıştır.

###### TextBox Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *ColumnAdressText*

***Text:*** *1*

###### Label Nesnesinin Değişen Özellikleri

***Name:*** *ColumnAdressLabel*

***Caption:*** *Veri Adresi (Sütun)*

###### CommandButton Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *infButton\_ColumnAdress*

***Caption:*** *i*

##### Başlangıç Satırı

Verilerin hangi satırdan başladığını öğrenmek için ‘*ToolBox*’ penceresi altından bir ‘*TextBox*’ nesnesi, bu nesneyi belirtmek amacı ile bir ‘*Label*’ nesnesi ve açıklama için bir ‘*CommandButton*’ nesnesi oluşturulmuştur. Oluşturulan ‘*TextBox*’ nesnesi için ekte (EK 34) verilen kodlar ve ‘*CommandButton*’ nesnesi için ekte (EK 35) verilen kodlar yazılmıştır.

###### TextBox Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *FirstRowText*

***Text:*** *1*

###### Label Nesnesinin Değişen Özellikleri

***Name:*** *FirstRowLabel*

***Caption:*** *Başlangıç Satırı*

###### CommandButton Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *infButton\_firstData*

***Caption:*** *i*

##### Anlamlılık Düzeyi

Testin anlamlılık düzeyi için ‘*ToolBox*’ penceresi altından bir ‘*TextBox*’ nesnesi ve bu nesneyi belirtmek amacı ile bir ‘*Label*’ nesnesi oluşturulmuştur. Oluşturulan ‘*TextBox*’ nesnesi için ekte (EK 36) verilen kodlar yazılmıştır.

###### TextBox Nesnesinin Değişen Özellikleri ve Oluşturulan Yordamları

***Name:*** *alfaText*

***Text:*** *1*

###### Label Nesnesinin Değişen Özellikleri

***Name:*** *alfaLabel*

***Caption:*** *Anlamlılık Düzeyi*

##### Hesapla Butonu

Girdileri derleyip işlem yapmak ve takibinde sonucu döndürmek için bir ‘CommandButton’ nesnesi oluşturulmuştur.

###### CommandButton Nesnesinin Değişen Özellikleri

***Name:*** *CalcButton*

***Caption:*** *Hesapla*

##### Hipotez Sonucu

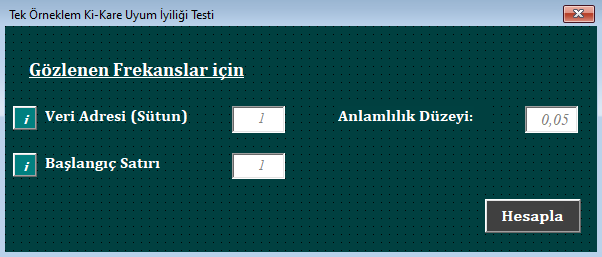
Hipotez sonucunu ve örneklem medyanını aktarmak üzere iki ‘*TextBox*’ nesnesi oluşturulmuştur.

###### TextBox Nesnesinin Değişen Özellikleri

***Name:*** *OutText*

***Text:***

##### Tasarımı Tamamlanmış Form Nesnesi (BinomForm)



Şekil 3 (Tek Örneklem Ki-Kare Uyum İyiliği Testi Form Arayüzü)

#### Form Nesnesi için Kodlar

Hesaplama ve sonuç döndürme işlemi için ‘*CalcButton*’ (Hesapla) isimli butonun ‘*Click*’ yordamında işlemler yapılacaktır.

##### CalcButton Nesnesinin Click Olay Yordamı

Bu yordam içinde testin olasılık hesabı yapılır ve gerekli hipotez sınanarak sonuç döndürülür. Öncelikle form arayüzündeki “*TextBox*” nesnelerinden değerler alınır ve ilgili değişkene aktarılır. Eğer boş bırakılan bir “*TextBox*” nesnesi varsa kullanıcı bilgilendirilir. Girilen veri sütunundaki satırlar incelenerek boş bir hücreye denk gelene kadar bir döngü oluşturulur ve bu döngü sonucu örneklem hacmi ve örneklem birimleri belirlenir. Daha sonra örneklem birimlerinin toplamı ile serbestlik dereceleri belirlenir. Serbestlik derecesi de belirlendikten sonra VBA’da bulunan “*ChiSq\_Inv\_RT*” fonksiyonu yardımı ile ki-kare tablo değeri belirlenir. Bir döngü ile ki-kare hesap değeri hesaplanır ve ardından hipotez sınanır. Sonuçlar kullanıcıya ‘*MsgBox*’ nesnesi ile verilir. Bu yordama ilişkin kodlar ektedir. (EK 37)

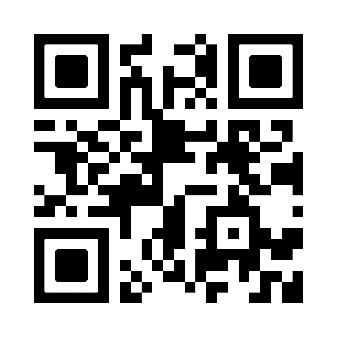
### Fonksiyon (Form Nesnesi Kullanılmadan)

Daha önceden oluşturulmuş ve ‘Name’ özelliği ‘CalcWithoutForm’ olarak değiştirilmiş olan modülün içine ekte (EK 38) verilen ‘*KiKareTesti*’ isimli fonksiyon yazılmıştır. Bu fonksiyon form Arayüzü kullanılmadığı için gerekli verileri parametreleri ile almaktadır. Boş bırakılan parametreler için kullanıcı bilgilendirilir ve değeri girmesi istenir. Ardından hipotez için sınamalar yapılır ve kullanıcı ‘*MsgBox*’ nesnesi ile bilgilendirilir.

### Makroyu İçeren Excel Dosyası

Ki-kare uyum iyiliği testi makrosu için aşağıdaki bağlantıyı kullanabilirsiniz.

<https://github.com/melloBlack/Tez>



## Wilcoxon İşaret Sıralaması Testi

Bu testte, örneklem verilerinden medyan değeri çıkartılarak bu değerler yeni bir dizin değişkene atanmalıdır. Bununla birlikte sıra numaraları verilmek üzere iç içe döngü kullanılmalıdır. Bu döngü içinde sıra numarası negatif olan değerleri bir değişken ve sıra numaraları pozitif olan değerleri başka bir değişken tutmalıdır. Ardından bu iki değişkenden hangisi daha küçük değere sahip ise o değişken test istatistiği olarak kullanılır.

Hipotez sınaması yapılmak üzere tablo değerine ihtiyaç vardır. Tablo değerlerini Excel’de tutabilir veya bu değerleri tutan bir fonksiyon oluşturulabilir. Alternatif bir diğer yol olarak süreklilik düzeltmesi uygulanabilir ve takibinde standart normal dağılım tablosu kullanılabilir. Standart normal dağılım tablo değerleri için Excel’de hazır bir fonksiyon bulunduğundan tablo değerlerini bu yöntem ile elde edilebilir. Ancak unutulmamalıdır ki süreklilik düzeltmesi yapılması testin küçük örneklemlerdeki duyarlılığını azaltabilir.

# Ekler

###### EK 1

Örnek Excel makrosu aşağıdaki gibidir.

‘meanCalc isminde, range olarak tanımlı x parametresine sahip bir fonksiyon tanımlandı

Function meanCalc(x As Range)

Dim rowsCount As Integer, columnsCount As Integer

Dim i As Integer, j As Integer, total As Single, n As Integer

‘Parametre üzerinden satır sayısına ulaşılıp satır sayısı rowsCount değişkenine aktarıldı.

rowsCount = x.Rows.Count

‘Parametre üzerinden sütun sayısına ulaşılıp sütun sayısı columnsCount değişkenine aktarıldı.

columnsCount = x.Columns.Count

‘Satır sayısı ile sütun sayısı çarpılarak toplam veri sayısı hesaplandı ve n değişkenine atandı.

n = rowsCount \* columnsCount

‘i değişkeni 1 den başlayıp satır sayısına ulaşana kadar devam edecek For döngüsü oluşturuldu.

For i = 1 To rowsCount

‘j değişkeni 1 den başlayıp sütun sayısına ulaşana kadar devam edecek For döngüsü oluşturuldu.

For j = 1 To columnsCount

‘Her bir veri bir önceki total değişkeninin değeri ile toplanıyor.

total = total + x.Cells(i,j)

Next j

Next i

‘Fonksiyon verilerin toplamını içeren total değişkeninin, toplam veri sayısını içeren n değişkenine bölümünü (aritmetik ortalamayı) döndürüyor.

meanCalc = total / n

End Function

###### EK 2

Binom testinde form arayüzünü açan fonksiyon aşağıdaki gibidir.

‘Form Nesnesini Açan Fonksiyon

Function BinomTestForm()

BinomForm.Show

End Function

###### EK 3

Binom testinde ilgilenilen durum değişkeni için oluşturulan ‘*TextBox*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı ValueText isimli TextBox nesnesini seçtiğinde TextBox nesnesinin içi temizlenir.

Private Sub ValueText\_Enter()

ValueText.Text = ""

End Sub

‘Kullanıcı ValueText isimli TextBox nesnesine değer girdiğinde yazı rengi siyah olur ve yazı tipinin italik özelliği kapatılır.

Private Sub ValueText\_Change()

ValueText.ForeColor = RGB(0,0,0)

ValueText.Font.Italic = False

End Sub

###### EK 4

Binom testinde ilgilenilen durum değişkeni için oluşturulan ‘*CommandButton*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı infButton\_value butonuna tıkladığında bir MessageBox kutusu açılır ve information değişkeni ile mesaj aktarılır.

Private Sub infButton\_value\_Click()

Dim infromation As String, title As String

information = "Binom dağılmış bir veri setinde 'Yazı/Tura, Evet/Hayır, Doğru/Yanlış, 0/1' gibi iki durum söz konusudur. Buraya ilgilendiğiniz durumun (x) hangisi olduğunu giriniz."

title = "Bilgilendirme (İlgilenilen Durum)"

MsgBox information, VbOkOnly, title

End Sub

###### EK 5

Binom testinde test olasılığı değişkeni için oluşturulan ‘*TextBox*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı RprobValueText isimli TextBox nesnesini seçtiğinde TextBox nesnesinin içi temizlenir.

Private Sub RprobValueText\_Enter()

RprobValue.Text = ""

End Sub

‘Kullanıcı RprobValueText isimli TextBox nesnesine değer girdiğinde yazı rengi siyah olur ve yazı tipinin italik özelliği kapatılır.

Private Sub RprobValueText\_Change()

RprobValueText.ForeColor = RGB(0,0,0)

RprobValueText.Font.Italic = False

End Sub

‘Kullanıcı RprobValueText isimli TextBox nesnesine sayısal bir değer girmez ise nesneden çıkamaz.

Private Sub RprobValueText\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

If IsNumeric(RprobValueText.Value) Then

Cancel = False

‘Kullanıcı RprobValueText isimli TextBox nesnesine 0 ile 1 arasında bir olasılık değeri girmez ise uyarı alır ve varsayılan değer 0,5 olur.

If RprobValueText.Value < 0 Or RprobValueText.Value > 1 Then

Cancel = True

MsgBox (“Lütfen 0 ile 1 arası bir olasılık değeri giriniz!”)

RprobValueText.Text = “0,5”

End If

End If

End Sub

###### EK 6

Binom testinde test olasılığı değişkeni için oluşturulan ‘*CommandButton*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı infButtonRprob butonuna tıkladığında bir MessageBox kutusu açılır ve information değişkeni ile mesaj aktarılır.

Private Sub infButtonRprob\_Click()

Dim infromation As String, title As String

information = "Test edilmek istenen oran. Bu değer ilgilenilen durum için test edilmek istenen oran ve/veya kabul edilen gerçek anakütle oranıdır. Varsayılan değeri 0,5'dir.."

title = "Bilgilendirme (Test Olasılığı)"

MsgBox information, VbOkOnly, title

End Sub

###### EK 7

Binom testinde veri adresi (sütun) değişkeni için oluşturulan ‘*TextBox*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı ColumnAdressText isimli TextBox nesnesini seçtiğinde TextBox nesnesinin içi temizlenir.

Private Sub ColumnAdressText\_Enter()

ColumnAdressText.Text = ""

End Sub

‘Kullanıcı ColumnAdressText isimli TextBox nesnesine değer girdiğinde yazı rengi siyah olur ve yazı tipinin italik özelliği kapatılır.

Private Sub ColumnAdressText\_Change()

ColumnAdressText.ForeColor = RGB(0,0,0)

ColumnAdressText.Font.Italic = False

End Sub

‘Kullanıcı ColumnAdressText isimli TextBox nesnesine sayısal bir değer girmez ise nesneden çıkamaz ve sütun adresini sayısal olarak girmesi gerektiğini aktaran bir bilgilendirme mesajı alır.

Private Sub ColumnAdressText\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

If IsNumeric(ColumnAdressText.Value) Then

Cancel = False

Else

Cancel = True

MsgBox ("Lütfen yalnızca sayı giriniz!" & vbCrLf & "'A' için 1" & vbCrLf & "'B' için 2" & vbCrLf & "'C' için 3" & vbCrLf & "." & vbCrLf & "." & vbCrLf & ".")

ColumnAdressText.Text = “1”

End If

End Sub

###### EK 8

Binom testinde veri adresi (sütun) değişkeni için oluşturulan ‘*CommandButton*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı infButton\_ColAdress butonuna tıkladığında bir MessageBox kutusu açılır ve information değişkeni ile mesaj aktarılır.

Private Sub infButton\_ColAdress\_Click()

Dim infromation As String, title As String

information = " Buraya verilerinizin bulunduğu sütun sırasını giriniz." & vbCrLf & "'A' için 1" & vbCrLf & "'B' için 2" & vbCrLf & "'C' için 3" & vbCrLf & "." & vbCrLf & "." & vbCrLf & "."

title = "Bilgilendirme (Veri Adresi Sütun)"

MsgBox information, VbOkOnly, title

End Sub

###### EK 9

Binom testinde başlangıç satırı değişkeni için oluşturulan ‘*TextBox*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı FirstRowText isimli TextBox nesnesini seçtiğinde TextBox nesnesinin içi temizlenir.

Private Sub FirstRowText\_Enter()

FirstRowText.Text = ""

End Sub

‘Kullanıcı FirstRowText isimli TextBox nesnesine değer girdiğinde yazı rengi siyah olur ve yazı tipinin italik özelliği kapatılır.

Private Sub FirstRowText\_Change()

FirstRowText.ForeColor = RGB(0,0,0)

FirstRowText.Font.Italic = False

End Sub

‘Kullanıcı FirstRowText isimli TextBox nesnesine sayısal bir değer girmez ise nesneden çıkamaz. Varsayılan değer 1 olur ve ilk satır adresini sayısal olarak girmesi gerektiğini aktaran bir bilgilendirme mesajı alır.

Private Sub FirstRowText\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

If IsNumeric(FirstRowText.Value) Then

Cancel = False

Else

Cancel = True

MsgBox ("Lütfen yalnızca sayı giriniz!")

FirstRowText.Text = “1”

End If

End Sub

###### EK 10

Binom testinde başlangıç satırı değişkeni için oluşturulan ‘*CommandButton*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı infButton\_firstRow butonuna tıkladığında bir MessageBox kutusu açılır ve information değişkeni ile mesaj aktarılır.

Private Sub infButton\_firstRow\_Click()

Dim infromation As String, title As String

information = “Buraya verilerinizin başladığı satır numarasını giriniz." & vbCrLf & "Buraya herhangi bir değer girilmez ise başlangıç satırı 1 olarak alınır.”

title = "Bilgilendirme (Başlangıç Satırı)"

MsgBox information, VbOkOnly, title

End Sub

###### EK 11

Binom testinde anlamlılık düzeyi değişkeni için oluşturulan ‘*TextBox*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı alfaText isimli TextBox nesnesini seçtiğinde TextBox nesnesinin içi temizlenir.

Private Sub alfaText\_Enter()

alfaText.Text = ""

End Sub

‘Kullanıcı alfaText isimli TextBox nesnesine değer girdiğinde yazı rengi siyah olur ve yazı tipinin italik özelliği kapatılır.

Private Sub alfaText\_Change()

alfaText.ForeColor = RGB(0,0,0)

alfaText.Font.Italic = False

End Sub

‘Kullanıcı alfaText isimli TextBox nesnesine sayısal bir değer girmez ise nesneden çıkamaz. Varsayılan değer 0,05 olur ve 𝛼 anlamlılık düzeyini sayısal olarak girmesi gerektiğini aktaran bir bilgilendirme mesajı alır.

Private Sub alfaText\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

If IsNumeric(alfaText.Value) Then

Cancel = False

Else

Cancel = True

MsgBox ("Lütfen yalnızca sayı giriniz!")

alfaText.Text = “0,05”

End If

End Sub

###### EK 12

Binom testinde ‘*General / Declarations*’ bölümünde tanımlanan değişkenlere ilişkin kodlar aşağıdaki verilmiştir.

Dim n() As Variant

Dim first\_row As Integer, column\_adress As Integer

Dim p\_value As String, alfa\_value As Single

Dim amount\_x As Integer, amount\_y As Integer

Dim sumProbability As Double

Dim testProb As Single

###### EK 13

Binom testinde ‘*CalcButton*’ nesnesinin ‘*Click*’ olay yordamına ilişkin kodlar aşağıda verilmiştir.

Public Sub CalcButton\_Click()

Dim index As Integer, arrayCount As Integer

amount\_x = 0

amount\_y = 0

index = 0

arrayCount = 0

‘Form ara yüzündeki TextBox nesnelerinden değerler alınır ve ilgili değişkene aktarılır.

p\_value = ValueText.Text

testProb = RprobValueText.Text

alfa\_value = alfaText.Text

first\_row = FirstRowText.Text

column\_adress = ColumnAdressText.Text

‘İlgilenilen durumun boş bırakılıp bırakılmadığı kontrol edilir. Eğer boş bırakıldıysa kullanıcı bilgilendirilir ve ilgilenilen durum veri setindeki ilk veri olur.

If IsNull(p\_value) Or p\_value = "" Or p\_value = " " Then

p\_value = ActiveSheet.Cells(first\_row, column\_adress).Value

MsgBox "İlgilenilen durum bölümünü boş bıraktığınız için ilgilenilen durum, veri setinizdeki ilk değer olan '" & p\_value & "' değeri olarak kabul edildi."

End If

‘Girilen veri sütunundaki satırları inceleyerek boş bir hücreye denk gelene kadar döngü devam eder ve örneklem hacmi belirlenir.

Do

arrayCount = arrayCount + 1

Loop Until IsEmpty(ActiveSheet.Cells(first\_row + (arrayCount - 1), column\_adress))

‘Örneklem hacmi artık bilindiğinden dolayı n dizisi boyutlandırılır.

ReDim n(arrayCount)

‘İlgilenilen ve ilgilenilmeyen değişkenlerin miktarları belirlenir.

Do

n(index) = ActiveSheet.Cells(first\_row + index, column\_adress)

If n(index) = p\_value Then

amount\_x = amount\_x + 1

Else

amount\_y = amount\_y + 1

End If

index = index + 1

Loop Until IsEmpty(ActiveSheet.Cells(first\_row + index, column\_adress))

‘Kullanıcının yaptığı alternatif hipotez seçimi kontrol edilir.

If twoTailed.Value = True Then

Dim amount\_interested As Integer, amount\_notInterested As Integer

If amount\_x < amount\_y Then

amount\_interested = amount\_x

amount\_notInterested = amount\_y

Else

amount\_interested = amount\_y

amount\_notInterested = amount\_x

End If

‘Eğer çift taraflı (eşit değildir) hipotezi seçildiyse pCalcNullHyp fonksiyonu çağırılır ve parametreleri olarak ilgilenilen durum miktarını tutan değişken, ilgilenilmeyen durum miktarını tutan değişken ve test edilmek istenen gerçek anakütle olasılık değerini tutan değişken girilir. Fonksiyonun döndürdüğü olasılık değeri sumProbability isimli değişkene atanır.

sumProbality = pCalcNullHyp(amount\_x, amount\_y, testProb)

If sumProbality <= (alfa\_value / 2) Then

‘Hesaplanan olasılık değeri 𝛼 anlamlılık düzeyinin yarısından küçükse Yokluk Hipotezi Reddedilir ve kullanıcıya MessageBox ile örneklem hacmi, ilgilenilen değişkenin örneklemdeki miktarı, hesaplanan olasılık değeri ve Yokluk Hipotezinin reddine ilişkin yorum aktarılır.

OutText.Text = "Yokluk Hipotezi Reddedildi."

MsgBox "n : " & (amount\_interested + amount\_notInterested) & vbTab & "x : " & amount\_interested & vbTab & "(n-x) : " & amount\_notInterested & vbCrLf & "P(X=x) = " & VBA.Round(sumProbality, 4) & " < alfa = " & (alfa\_value / 2) & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde ilgilenilen durumun gerçekleşme oranı ile test edilen oran (varsayılan anakütle oranı) arasında istatiksel olarak anlamlı bir fark vardır.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

Else

‘Hesaplanan olasılık değeri 𝛼 anlamlılık düzeyinin yarısından büyükse Alternatif Hipotezi Reddedilir ve kullanıcıya MessageBox ile örneklem hacmi, ilgilenilen değişkenin örneklemdeki miktarı, hesaplanan olasılık değeri ve Alternatif Hipotezin reddine ilişkin yorum aktarılır.

OutText.Text = "Yokluk Hipotezi Reddedilemez."

MsgBox "n : " & (amount\_interested + amount\_notInterested) & vbTab & "x : " & amount\_interested & vbTab & "(n-x) : " & amount\_notInterested & vbCrLf & "P(X=x) = " & VBA.Round(sumProbality, 4) & " > alfa = " & (alfa\_value / 2) & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde ilgilenilen durumun gerçekleşme oranı ile test edilen oran (varsayılan anakütle oranı) arasında istatiksel olarak anlamlı bir fark yoktur.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

ElseIf oneTailed\_neg.Value = True Then

‘Eğer tek taraflı küçüktür hipotezi seçildiyse pCalcIsSmall fonksiyonu çağırılır ve parametreleri olarak ilgilenilen durum miktarını tutan değişken, ilgilenilmeyen durum miktarını tutan değişken ve test edilmek istenen gerçek anakütle olasılık değerini tutan değişken girilir. Fonksiyonun döndürdüğü olasılık değeri sumProbability isimli değişkene atanır.

sumProbality = pCalcIsSmall(amount\_x, amount\_y, testProb)

If sumProbality < alfa\_value Then

‘Hesaplanan olasılık değeri 𝛼 anlamlılık düzeyinden küçükse Yokluk Hipotezi Reddedilir ve kullanıcıya MessageBox ile örneklem hacmi, ilgilenilen değişkenin örneklemdeki miktarı, hesaplanan olasılık değeri ve Yokluk Hipotezinin reddine ilişkin yorum aktarılır.

OutText.Text = "Yokluk Hipotezi Reddedildi."

MsgBox "n : " & (amount\_x + amount\_y) & vbTab & "x : " & amount\_x & vbTab & "(n-x) : " & amount\_y & vbCrLf & "P(X<=" & amount\_x & ") = " & VBA.Round(sumProbality, 4) & " < alfa = " & alfa\_value & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde ilgilenilen durumun gerçekleşme oranı, test edilen orandan (varsayılan anakütle oranından) istatiksel olarak anlamlı bir şekilde daha küçüktür.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

Else

‘Hesaplanan olasılık değeri 𝛼 anlamlılık düzeyinden büyükse Alternatif Hipotezi Reddedilir ve kullanıcıya MessageBox ile örneklem hacmi, ilgilenilen değişkenin örneklemdeki miktarı, hesaplanan olasılık değeri ve Alternatif Hipotezin reddine ilişkin yorum aktarılır.

OutText.Text = "Yokluk Hipotezi Reddedilemez."

MsgBox "n : " & (amount\_x + amount\_y) & vbTab & "x : " & amount\_x & vbTab & "(n-x) : " & amount\_y & vbCrLf & "P(X<=" & amount\_x & ") = " & VBA.Round(sumProbality, 4) & " > alfa = " & alfa\_value & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde ilgilenilen durumun gerçekleşme oranı ile test edilen oran (varsayılan anakütle oranı) arasında istatiksel olarak anlamlı bir fark yoktur.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

Else

‘Eğer tek taraflı küçüktür hipotezi ve çift taraflı (eşit değildir) hipotezi seçilmediyse tek taraflı büyüktür hipotezi seçilmiştir. Bu durumda pCalcIsBig fonksiyonu çağırılır ve parametreleri olarak ilgilenilen durum miktarını tutan değişken, ilgilenilmeyen durum miktarını tutan değişken ve test edilmek istenen gerçek anakütle olasılık değerini tutan değişken girilir. Fonksiyonun döndürdüğü olasılık değeri sumProbability isimli değişkene atanır.

sumProbality = pCalcIsBig(amount\_x, amount\_y, testProb)

If sumProbality < alfa\_value Then

‘Hesaplanan olasılık değeri 𝛼 anlamlılık düzeyinden küçükse Yokluk Hipotezi Reddedilir ve kullanıcıya MessageBox ile örneklem hacmi, ilgilenilen değişkenin örneklemdeki miktarı, hesaplanan olasılık değeri ve Yokluk Hipotezinin reddine ilişkin yorum aktarılır.

OutText.Text = "Yokluk Hipotezi Reddedildi."

MsgBox "n : " & (amount\_x + amount\_y) & vbTab & "x : " & amount\_x & vbTab & "(n-x) : " & amount\_y & vbCrLf & "P(X>" & amount\_x & ") = " & VBA.Round(sumProbality, 4) & " < alfa = " & alfa\_value & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde ilgilenilen durumun gerçekleşme oranı test edilen orandan (varsayılan anakütle oranından) istatiksel olarak anlamlı bir şekilde daha büyüktür.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

Else

‘Hesaplanan olasılık değeri 𝛼 anlamlılık düzeyinden büyükse Alternatif Hipotezi Reddedilir ve kullanıcıya MessageBox ile örneklem hacmi, ilgilenilen değişkenin örneklemdeki miktarı, hesaplanan olasılık değeri ve Alternatif Hipotezin reddine ilişkin yorum aktarılır.

OutText.Text = "Yokluk Hipotezi Reddedilemez."

MsgBox "n : " & (amount\_x + amount\_y) & vbTab & "x : " & amount\_x & vbTab & "(n-x) : " & amount\_y & vbCrLf & "P(X>" & amount\_x & ") = " & VBA.Round(sumProbality, 4) & " > alfa = " & alfa\_value & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde ilgilenilen durumun gerçekleşme oranı ile test edilen oran (varsayılan anakütle oranı) arasında istatiksel olarak anlamlı bir fark yoktur.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

End If

End Sub

###### EK 14

Binom testinde alternatif hipotezin eşit olmadığı savının olasılığını “” hesaplayan fonksiyon (*pCalcNullHyp*’) için kodlar aşağıda verilmiştir.

Public Function pCalcNullHyp(x As Integer, y As Integer, testProbability As Single) As Double

Dim Combination As Double, nFac As Variant, xFac As Variant, diffFac As Variant

Dim q\_prob As Double, p\_prob As Double, Calc As Double, probCollec As Double

Dim i As Integer, j As Integer, k As Integer, npq As Single, z\_calc As Double, b As Integer

Dim sample\_size As Integer, instan\_x As Integer, b As Integer, instant\_y As Double

‘Fonksiyona parametreleri ile aktarılan x ve y değişkenlerinin değerleri toplanarak örneklem hacmi belirlenir. Bu toplam sample\_size değişkenine atanır.

sample\_size = x + y

‘Dağılımın normale yaklaşıp yaklaşmadığını sınamak için npq isimli değişkene ilgilenilen durumun örneklem oranı, ilgilenilmeyen durumun örneklem oranı ve örneklem hacmi çarpımları aktarılır.

npq = sample\_size \* (x / sample\_size) \* (y / sample\_size)

‘Dağılımın normale yaklaşıp yaklaşmadığı sınanır.

If npq >= 9 Then

'Dağılım normale yaklaşıyor ise süreklilik düzeltmesi uygulanır ve bir VBA fonksiyonu ile normal dağılım için ilgili olasılık hesaplanır.

If x > (testProbability \* sample\_size) Then

z\_calc = ((x - 0.5) - (sample\_size / 2)) / Sqr(sample\_size / 4)

Else

z\_calc = ((x + 0.5) - (sample\_size / 2)) / Sqr(sample\_size / 4)

End If

pCalcNullHyp = WorksheetFunction.Norm\_S\_Dist(z\_calc, True)

Else

‘Dağılım normale yaklaşmıyor ise olasılık değeri binom dağılımı ile hesaplanır.

p\_prob = testProbability

q\_prob = (1 - testProbability)

nFac = 1

xFac = 1

diffFac = 1

For i = 1 To (x + y)

nFac = nFac \* i

Next i

For b = x To 0 Step -1

instant\_x = b

instant\_y = ((x + y) -b)

diffFac = 1

xFac = 1

For j = 1 To x

xFac = xFac \* j

Next j

For k = 1 To y

diffFac = diffFac \* k

Next k

Combination = (nFac / (diffFac \* xFac))

Calc = (p\_prob ^ instant\_x) \* (q\_prob ^ instant\_y)

probCollec = (Combination \* Calc)

Next b

pCalcNullHyp = probCollec

End If

End Function

###### EK 15

Binom testinde ilgilenilen değişkenin küçük olması “” olasılığını hesaplayan fonksiyon (*pCalcIsSmall*’) için kodlar aşağıda verilmiştir.

Public Function pCalcIsSmall(x As Integer, y As Integer, testProbability As Single) As Double

Dim Combination As Double, nFac As Variant, xFac As Variant, diffFac As Variant

Dim q\_prob As Double, p\_prob As Double, Calc As Double, probCollec As Double

Dim instant\_x As Integer, instant\_y As Integer,

Dim i As Integer, j As Integer, k As Integer, b As Integer

Dim npq As Single, sample\_size As Integer, z\_calc As Double

‘Fonksiyona parametreleri ile aktarılan x ve y değişkenlerinin değerleri toplanarak örneklem hacmi belirlenir. Bu toplam sample\_size değişkenine atanır.

sample\_size = x + y

‘Dağılımın normale yaklaşıp yaklaşmadığını sınamak için npq isimli değişkene ilgilenilen durumun örneklem oranı, ilgilenilmeyen durumun örneklem oranı ve örneklem hacmi çarpımları aktarılır.

npq = sample\_size \* (x / sample\_size) \* (y / sample\_size)

If npq >= 9 Then

'Dağılım normale yaklaşıyor ise süreklilik düzeltmesi uygulanır ve bir VBA fonksiyonu ile normal dağılım için ilgili olasılık hesaplanır.

If x > (testProbability \* sample\_size) Then

z\_calc = ((x - 0.5) - (sample\_size / 2)) / Sqr(sample\_size / 4)

Else

z\_calc = ((x + 0.5) - (sample\_size / 2)) / Sqr(sample\_size / 4)

End If

pCalcIsSmall = WorksheetFunction.Norm\_S\_Dist(z\_calc, True)

Else

‘Dağılım normale yaklaşmıyor ise olasılık değeri binom dağılımı ile hesaplanır.

p\_prob = testProbability

q\_prob = (1 - testProbability)

nFac = 1

xFac = 1

diffFac = 1

For i = 1 To (x + y)

nFac = nFac \* i

Next i

For b = x To 0 Step -1

instant\_x = b

instant\_y = ((x + y) - b)

xFac = 1

diffFac = 1

For j = 1 To instant\_x

xFac = xFac \* j

Next j

For k = 1 To instant\_y

diffFac = diffFac \* k

Next k

Combination = (nFac / (diffFac \* xFac))

Calc = (p\_prob ^ instant\_x) \* (q\_prob ^ instant\_y)

probCollec = probCollec + (Combination \* Calc)

Next b

pCalcIsSmall = probCollec

End If

End Function

###### EK 16

Binom testinde ilgilenilen değişkenin büyük olması “” olasılığını hesaplayan fonksiyon (*pCalcIsBig*’) için kodlar aşağıda verilmiştir.

Public Function pCalcIsBig(x As Integer, y As Integer, testProbability As Single) As Double

Dim Combination As Double, nFac As Variant, xFac As Variant, diffFac As Variant

Dim q\_prob As Double, p\_prob As Double, Calc As Double, probCollec As Double

Dim instant\_x As Integer, instant\_y As Integer

Dim i As Integer, j As Integer, k As Integer, b As Integer

Dim npq As Single, sample\_size As Integer, z\_calc As Double

‘Fonksiyona parametreleri ile aktarılan x ve y değişkenlerinin değerleri toplanarak örneklem hacmi belirlenir. Bu toplam sample\_size değişkenine atanır.

sample\_size = x + y

‘Dağılımın normale yaklaşıp yaklaşmadığını sınamak için npq isimli değişkene ilgilenilen durumun örneklem oranı, ilgilenilmeyen durumun örneklem oranı ve örneklem hacmi çarpımları aktarılır.

npq = sample\_size \* (x / sample\_size) \* (y / sample\_size)

If npq >= 9 Then

'Dağılım normale yaklaşıyor ise süreklilik düzeltmesi uygulanır ve bir VBA fonksiyonu ile normal dağılım için ilgili olasılık hesaplanır.

If x > (testProbability \* sample\_size) Then

z\_calc = ((x - 0.5) - (sample\_size / 2)) / Sqr(sample\_size / 4)

Else

z\_calc = ((x + 0.5) - (sample\_size / 2)) / Sqr(sample\_size / 4)

End If

pCalcIsBig = (1 - WorksheetFunction.Norm\_S\_Dist(z\_calc, True))

Else

‘Dağılım normale yaklaşmıyor ise olasılık değeri binom dağılımı ile hesaplanır.

p\_prob = testProbability

q\_prob = (1 - testProbability)

nFac = 1

xFac = 1

diffFac = 1

For i = 1 To (x + y)

nFac = nFac \* i

Next i

For b = (x + 1) To (x + y)

instant\_x = b

instant\_y = ((x + y) - b)

xFac = 1

diffFac = 1

For j = 1 To instant\_x

xFac = xFac \* j

Next j

For k = 1 To instant\_y

diffFac = diffFac \* k

Next k

Combination = (nFac / (diffFac \* xFac))

Calc = (p\_prob ^ instant\_x) \* (q\_prob ^ instant\_y)

probCollec = probCollec + (Combination \* Calc)

Next b

pCalcIsBig = probCollec

End If

End Function

###### EK 17

Form nesnesi kullanılmadan yapılan Binom testi için fonksiyona ilişkin kodlar aşağıda verilmiştir.

Function BinomTest(data As Range, Optional x As String, Optional testProb As Single, Optional alfa\_value As Single) As Variant

Dim rowsCount As Integer, columnsCount As Integer, index As Integer

Dim k As Integer, nCount As Integer

Dim amount\_x As Integer, amount\_y As Integer, hypothesis As Integer

Dim answerP1 As String, answerP2 As String

Dim n() As Variant

Dim probability As Double

index = 1

amount\_x = 0

amount\_y = 0

‘Fonksiyonun ilk parametresi olan data değişkeni üzerinden satır ve sütun sayılarına ulaşılır ve ilgili değişkene atanır.

rowsCount = data.Rows.Count

columnsCount = data.Columns.Count

‘Eğer hem birden fazla satır hem de birden fazla sütun seçildiyse kullanıcı bilgilendirilir.

If rowsCount > 1 And columnsCount > 1 Then

answerP1 = MsgBox("Verilerinizi seçerken birden fazla satır ve sütun belirttiniz. Bir yanlışlık olduğunu düşünüyorsanız lütfen yeniden seçim yapınız. Birden fazla satır ve sütun seçimi yapılarak devam edecekseniz tüm veriler tek br seriye ait olarak değerlendirilecektir. Verilerinizi yeniden seçmek istiyor musunuz?", vbYesNo + vbQuestion, "Birden Fazla Satır ve Sütun Seçimi")

‘Kullanıcı verileri yeniden fonksiyona girmek isterse fonksiyon gerekli bilgiyi döndürür ve fonksiyondan çıkılır.

If answerP1 = vbYes Then

BinomTest = "Lütfen verilerinizi yeniden girin."

Exit Function

End If

‘Herhangi bir hücre seçilmezse kullanıcı bilgilendirilir ve fonksiyondan çıkılır.

ElseIf rowsCount < 1 And columnsCount < 1 Then

MsgBox "Lütfen birinci parametre olarak verilerinizin adresini girin veya tablonuzdan seçin.", vbOKOnly + vbCritical, "Eksik Veri Girişi"

BinomTest = "Lütfen verilerinizi yeniden girin."

Exit Function

End If

‘Fonksiyonun ikinci parametresi olan ilgilenilen değer girilmez ise kullanıcı bilgilendirilir ve bir InputBox nesnesi açılarak ilgilenilen değerin girilmesi istenir.

If IsEmpty(x) Or x = "" Or x = " " Then

MsgBox "İlgilenilen durumu (x) boş bıraktınız veya girmediniz. Lütfen geçerli bir değer giriniz.", vbOKOnly + vbCritical, "İlgilenilen Durum (İkinci Parametre)"

x = InputBox("İlgilenilen durumu giriniz. Verileriniz sayısal değil ise lütfen büyük/küçük harfe dikkat ediniz.", "İlgilenilen Durum (Yeniden Giriş)", data.Cells(1, 1).Value)

End If

‘Fonksiyonun üçüncü parametresi olan test edilmek istenen gerçek anakütle olasılık değeri girilmez veya 0 ile 1 arasında bir değer girilmez ise kullanıcı bilgilendirilir ve test olasılığını girmek isteyip istemediğini soran bir MessageBox nesnesi açılır. Kullanıcı burada hayır cevabını verirse test olasılığı varsayılan olarak 0,5 kabul edilir ve işlemlere devam edilir.

Do Until testProb > 0 And testProb < 1

answerP2 = MsgBox("Test edilmek istenen (varsayılan anakütle) olasılık değerini 'üçüncü parametre' boş bıraktınız veya yanlış girdiniz. Varsayılan olarak test olasılığı 0,5 olarak alınacaktır. Farklı bir olasılık değeri girmek ister misiniz?", vbYesNo + vbQuestion, "Test Olasılığı")

If answerP2 = vbYes Then

testProb = InputBox("Test edilmek istenen (varsayılan anakütle) olasılık değerini giriniz.", "Test Olasılığı (Yeniden Giriş)", 0.5) 'Type:=1

Else

testProb = 0.5

End If

Loop

‘Fonksiyonun dördüncü parametresi olan 𝛼 anlamlılık düzeyi değeri girilmez veya 0 ile 1 arasında bir değer girilmez ise varsayılan olarak 0,05 kabul edilir ve işlemlere devam edilir.

If alfa\_value <= 0 Or alfa\_value >= 1 Then

alfa\_value = 0.05

End If

‘Örneklem hacmi satır ve sütun sayılarının çarpımı ile hesaplanır.

nCount = (rowsCount \* columnsCount)

‘Örneklem hacmi artık bilindiğinden nCount dizisi yeniden boyutlandırılır.

ReDim n(1 To nCount)

‘Veriler bir sütun ve birden fazla satırda ise ilgilenilen değişken ve ilgilenilmeyen değişken miktarı burada hesaplanır.

If rowsCount > 1 And columnsCount = 1 Then

Do

n(index) = data.Rows.Cells((index), 1).Value

If n(index) = x Then

amount\_x = amount\_x + 1

Else

amount\_y = amount\_y + 1

End If

index = index + 1

Loop Until rowsCount < index

‘Veriler bir satır ve birden fazla sütunda ise ilgilenilen değişken ve ilgilenilmeyen değişken miktarı burada hesaplanır.

ElseIf columnsCount > 1 And rowsCount = 1 Then

Do

n(index) = data.Columns.Cells(1, (index)).Value

If n(index) = x Then

amount\_x = amount\_x + 1

Else

amount\_y = amount\_y + 1

End If

index = index + 1

Loop Until columnsCount < index

‘Veriler birden fazla sürun ve birden fazla satırda ise ilgilenilen değişken ve ilgilenilmeyen değişken miktarı burada hesaplanır.

ElseIf columnsCount > 1 And rowsCount > 1 Then

For k = 1 To columnsCount

Do

n(index) = data.Rows.Cells((index), k).Value

If n(index) = x Then

amount\_x = amount\_x + 1

Else

amount\_y = amount\_y + 1

End If

index = index + 1

Loop Until rowsCount < index

index = 1

Next k

Else

MsgBox "Lütfen birinci parametre olarak verilerinizin adresini girin veya tablonuzdan seçin.", vbOKOnly + vbCritical, "Veri Girişinde Hata"

Exit Function

End If

‘Bir InputBox nesnesi açılarak kullanıcıdan Alternatif Hipotezi seçmesi istenir ve seçim hypothesis isimli değişkene aktarılır.

hypothesis = InputBox("Lütfen hipotez seçimi yapınız." & vcCrLf & "İlgilenilen durumun gerçekleşme olasılığı ile test edilmek istenen (varsayılan anakütle) gerçekleşme olasılığına ilişkin alternatif hipotezi seçiniz. " & vbCrLf & "Küçük olduğunu sınamak için 1 değerini giriniz." & vbCrLf & "Büyük olduğunu sınamak için 2 değerini giriniz. " & vbCrLf & "Eşit olmadığını sınamak için 0 değerini giriniz.", "Hipotez Seçimi", 0) 'Type:=1

‘Eğer çift taraflı (eşit değildir) hipotezi seçildiyse BinomForm isimli modülden pCalcNullHyp fonksiyonu çağırılır ve parametreleri olarak ilgilenilen durum miktarını tutan değişken, ilgilenilmeyen durum miktarını tutan değişken ve test edilmek istenen gerçek anakütle olasılık değerini tutan değişken girilir. Fonksiyonun döndürdüğü olasılık değeri probability isimli değişkene atanır.

If hypothesis = 0 Then

probability = BinomForm.pCalcNullHyp(amount\_x, amount\_y, testProb)

If probability <= (alfa\_value / 2) Then

‘Hesaplanan olasılık değeri 𝛼 anlamlılık düzeyinin yarısından küçükse Yokluk Hipotezi Reddedilir ve kullanıcıya MessageBox ile örneklem hacmi, ilgilenilen değişkenin örneklemdeki miktarı, hesaplanan olasılık değeri ve Yokluk Hipotezinin reddine ilişkin yorum aktarılır.

MsgBox "n : " & (amount\_x + amount\_y) & vbTab & "p : " & amount\_x & vbTab & "q : " & amount\_y & vbCrLf & "P(X=x) = " & VBA.Round(probability, 4) & " < alfa = " & (alfa\_value / 2) & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde ilgilenilen durumun gerçekleşme oranı ile test edilen oran (varsayılan anakütle oranı) arasında istatiksel olarak anlamlı bir fark vardır.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

Else

‘Hesaplanan olasılık değeri 𝛼 anlamlılık düzeyinin yarısından büyükse Alternatif Hipotezi Reddedilir ve kullanıcıya MessageBox ile örneklem hacmi, ilgilenilen değişkenin örneklemdeki miktarı, hesaplanan olasılık değeri ve Alternatif Hipotezin reddine ilişkin yorum aktarılır.

MsgBox "n : " & (amount\_x + amount\_y) & vbTab & "p : " & amount\_x & vbTab & "q : " & amount\_y & vbCrLf & "P(X=x) = " & VBA.Round(probability, 4) & " > alfa = " & (alfa\_value / 2) & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde ilgilenilen durumun gerçekleşme oranı ile test edilen oran (varsayılan anakütle oranı) arasında istatiksel olarak anlamlı bir fark yoktur.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

‘Eğer tek taraflı küçüktür hipotezi seçildiyse BinomForm isimli modülden pCalcIsSmall fonksiyonu çağırılır ve parametreleri olarak ilgilenilen durum miktarını tutan değişken, ilgilenilmeyen durum miktarını tutan değişken ve test edilmek istenen gerçek anakütle olasılık değerini tutan değişken girilir. Fonksiyonun döndürdüğü olasılık değeri probability isimli değişkene atanır.

ElseIf hypothesis = 1 Then

probability = BinomForm.pCalcIsSmall(amount\_x, amount\_y, testProb)

If probability < alfa\_value Then

Hesaplanan olasılık değeri 𝛼 anlamlılık düzeyinden küçükse Yokluk Hipotezi Reddedilir ve kullanıcıya MessageBox ile örneklem hacmi, ilgilenilen değişkenin örneklemdeki miktarı, hesaplanan olasılık değeri ve Yokluk Hipotezinin reddine ilişkin yorum aktarılır.

MsgBox "n : " & (amount\_x + amount\_y) & vbTab & "p : " & amount\_x & vbTab & "q : " & amount\_y & vbCrLf & "P(X<" & amount\_x & ") = " & VBA.Round(probability, 4) & " < alfa = " & alfa\_value & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde ilgilenilen durumun gerçekleşme oranı, test edilen orandan (varsayılan anakütle oranından) istatiksel olarak anlamlı bir şekilde daha küçüktür.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

Else

‘Hesaplanan olasılık değeri 𝛼 anlamlılık düzeyinden büyükse Alternatif Hipotezi Reddedilir ve kullanıcıya MessageBox ile örneklem hacmi, ilgilenilen değişkenin örneklemdeki miktarı, hesaplanan olasılık değeri ve Alternatif Hipotezin reddine ilişkin yorum aktarılır.

MsgBox "n : " & (amount\_x + amount\_y) & vbTab & "p : " & amount\_x & vbTab & "q : " & amount\_y & vbCrLf & "P(X<" & amount\_x & ") = " & VBA.Round(probability, 4) & " > alfa = " & alfa\_value & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde ilgilenilen durumun gerçekleşme oranı ile test edilen oran (varsayılan anakütle oranı) arasında istatiksel olarak anlamlı bir fark yoktur.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

‘Eğer tek taraflı büyüktür hipotezi seçildiyse BinomForm isimli modülden pCalcIsBig fonksiyonu çağırılır ve parametreleri olarak ilgilenilen durum miktarını tutan değişken, ilgilenilmeyen durum miktarını tutan değişken ve test edilmek istenen gerçek anakütle olasılık değerini tutan değişken girilir. Fonksiyonun döndürdüğü olasılık değeri probability isimli değişkene atanır.

ElseIf hypothesis = 2 Then

probability = BinomForm.pCalcIsBig(amount\_x, amount\_y, testProb)

If probability < alfa\_value Then

‘Hesaplanan olasılık değeri 𝛼 anlamlılık düzeyinden küçükse Yokluk Hipotezi Reddedilir ve kullanıcıya MessageBox ile örneklem hacmi, ilgilenilen değişkenin örneklemdeki miktarı, hesaplanan olasılık değeri ve Yokluk Hipotezinin reddine ilişkin yorum aktarılır.

MsgBox "n : " & (amount\_x + amount\_y) & vbTab & "p : " & amount\_x & vbTab & "q : " & amount\_y & vbCrLf & "P(X>" & amount\_x & ") = " & VBA.Round(probability, 4) & " < alfa = " & alfa\_value & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde ilgilenilen durumun gerçekleşme oranı test edilen orandan (varsayılan anakütle oranından) istatiksel olarak anlamlı bir şekilde daha büyüktür.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

Else

‘Hesaplanan olasılık değeri 𝛼 anlamlılık düzeyinden büyükse Alternatif Hipotezi Reddedilir ve kullanıcıya MessageBox ile örneklem hacmi, ilgilenilen değişkenin örneklemdeki miktarı, hesaplanan olasılık değeri ve Alternatif Hipotezin reddine ilişkin yorum aktarılır.

MsgBox "n : " & (amount\_x + amount\_y) & vbTab & "p : " & amount\_x & vbTab & "q : " & amount\_y & vbCrLf & "P(X>" & amount\_x & ") = " & VBA.Round(probability, 4) & " > alfa = " & alfa\_value & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde ilgilenilen durumun gerçekleşme oranı ile test edilen oran (varsayılan anakütle oranı) arasında istatiksel olarak anlamlı bir fark yoktur.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

Else

‘Hipotez seçimi yapılmadıysa fonksiyondan çıkılır.

Exit Function

End If

‘Fonksiyon ayrıca Excel’de yazıldığı hücreye, hesaplanan olasılık değerini döndürür.

BinomTest = VBA.Round(probability, 4)

End Function

###### EK 18

İşaret testinde form arayüzünü açan fonksiyon aşağıdaki gibidir.

‘Form Nesnesini Açan Fonksiyon

Function IsaretTestiForm()

IsaretForm.Show

End Function

‘Form Nesnesini Açan Fonksiyon

Function SignTestForm()

IsaretForm.Show

End Function

###### EK 19

İşaret testinde veri adresi (sütun) değişkeni için oluşturulan ‘*TextBox*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı ColumnAdressText isimli TextBox nesnesini seçtiğinde TextBox nesnesinin içi temizlenir.

Private Sub ColumnAdressText\_Enter()

ColumnAdressText.Text = ""

End Sub

‘Kullanıcı ColumnAdressText isimli TextBox nesnesine değer girdiğinde yazı rengi siyah olur ve yazı tipinin italik özelliği kapatılır.

Private Sub ColumnAdressText\_Change()

ColumnAdressText.ForeColor = RGB(0,0,0)

ColumnAdressText.Font.Italic = False

End Sub

‘Kullanıcı ColumnAdressText isimli TextBox nesnesine sayısal bir değer girmez ise nesneden çıkamaz ve sütun adresini sayısal olarak girmesi gerektiğini aktaran bir bilgilendirme mesajı alır.

Private Sub ColumnAdressText\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

If IsNumeric(ColumnAdressText.Value) Then

Cancel = False

Else

Cancel = True

MsgBox ("Lütfen yalnızca sayı giriniz!" & vbCrLf & "'A' için 1" & vbCrLf & "'B' için 2" & vbCrLf & "'C' için 3" & vbCrLf & "." & vbCrLf & "." & vbCrLf & ".")

ColumnAdressText.Text = “1”

End If

End Sub

###### EK 20

İşaret testinde veri adresi (sütun) değişkeni için oluşturulan ‘*CommandButton*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı infButton\_ColAdress butonuna tıkladığında bir MessageBox kutusu açılır ve information değişkeni ile mesaj aktarılır.

Private Sub infButton\_ColAdress\_Click()

Dim infromation As String, title As String

information = "Verilerinizin bulunduğu sütunu girin. A için 1, B için 2, C için 3... değerlerini giriniz."

title = "Bilgilendirme (Veri Adresi)"

MsgBox information, VbOkOnly, title

End Sub

###### EK 21

İşaret testinde başlangıç satırı değişkeni için oluşturulan ‘*TextBox*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı FirstRowText isimli TextBox nesnesini seçtiğinde TextBox nesnesinin içi temizlenir.

Private Sub FirstRowText\_Enter()

FirstRowText.Text = ""

End Sub

‘Kullanıcı FirstRowText isimli TextBox nesnesine değer girdiğinde yazı rengi siyah olur ve yazı tipinin italik özelliği kapatılır.

Private Sub FirstRowText\_Change()

FirstRowText.ForeColor = RGB(0,0,0)

FirstRowText.Font.Italic = False

End Sub

‘Kullanıcı FirstRowText isimli TextBox nesnesine sayısal bir değer girmez ise nesneden çıkamaz. Varsayılan değer 1 olur ve ilk satır adresini sayısal olarak girmesi gerektiğini aktaran bir bilgilendirme mesajı alır.

Private Sub FirstRowText\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

If IsNumeric(FirstRowText.Value) Then

Cancel = False

Else

Cancel = True

MsgBox ("Lütfen yalnızca sayısal bir değer giriniz!

FirstRowText.Text = “1”

End If

End Sub

###### EK 22

İşaret testinde başlangıç satırı değişkeni için oluşturulan ‘*CommandButton*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı infButton\_firstRow butonuna tıkladığında bir MessageBox kutusu açılır ve information değişkeni ile mesaj aktarılır.

Private Sub infButton\_firstRow\_Click()

Dim infromation As String, title As String

information = “Verilerinizin bulunduğu sütunda, verilerinizin kaçıncı satırda başladığını giriniz."

title = "Bilgilendirme (Başlangıç Satırı)"

MsgBox information, VbOkOnly, title

End Sub

###### EK 23

İşaret testinde anakütle medyanı değişkeni için oluşturulan ‘*TextBox*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı AMText isimli TextBox nesnesini seçtiğinde TextBox nesnesinin içi temizlenir.

Private Sub AMText\_Enter()

AMText.Text = ""

End Sub

‘Kullanıcı AMText isimli TextBox nesnesine değer girdiğinde yazı rengi siyah olur ve yazı tipinin italik özelliği kapatılır.

Private Sub AMText\_Change()

AMText.ForeColor = RGB(0,0,0)

AMText.Font.Italic = False

End Sub

‘Kullanıcı AMText isimli TextBox nesnesine sayısal bir değer girmez ise nesneden çıkamaz.

Private Sub AMText\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

If IsNumeric(AMText.Value) Then

Cancel = False

Else

Cancel = True

MsgBox ("Lütfen yalnızca sayısal bir değer giriniz!

AMText.Text = “1”

End If

End Sub

###### EK 24

İşaret testinde ana kütle değişkeni için oluşturulan ‘*CommandButton*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı infButton\_Median butonuna tıkladığında bir MessageBox kutusu açılır ve information değişkeni ile mesaj aktarılır.

Private Sub infButton\_Median\_Click()

Dim infromation As String, title As String

information = "Çekilen örneklemin temsil ettiği anakütleinin medyan değerini giriniz."

title = "Bilgilendirme (Anakütle Medyanı)"

MsgBox information, VbOkOnly, title

End Sub

###### EK 25

İşaret testinde anlamlılık düzeyi değişkeni için oluşturulan ‘*TextBox*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı alfaText isimli TextBox nesnesini seçtiğinde TextBox nesnesinin içi temizlenir.

Private Sub alfaText\_Enter()

alfaText.Text = ""

End Sub

‘Kullanıcı alfaText isimli TextBox nesnesine değer girdiğinde yazı rengi siyah olur ve yazı tipinin italik özelliği kapatılır.

Private Sub alfaText\_Change()

alfaText.ForeColor = RGB(0,0,0)

alfaText.Font.Italic = False

End Sub

‘Kullanıcı alfaText isimli TextBox nesnesine sayısal bir değer girmez ise nesneden çıkamaz. Varsayılan değer 0,05 olur ve 𝛼 anlamlılık düzeyini sayısal olarak girmesi gerektiğini aktaran bir bilgilendirme mesajı alır.

Private Sub alfaText\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

If IsNumeric(alfaText.Value) Then

Cancel = False

If alfaText.Value < 0 Or alfaText.Value > 1 Then

Cansel = True

MsgBox ("Lütfen 0 ile 1 arası bir değer giriniz!")

alfaText.Text = "0,05"

End If

Else

Cancel = True

MsgBox ("Lütfen 0 ile 1 arası bir değer giriniz!")

alfaText.Text = “0,05”

End If

End Sub

###### EK 26

İşaret testinde ‘*CalcButton*’ nesnesinin ‘*Click*’ olay yordamına ilişkin kodlar aşağıda verilmiştir.

Private Sub CalcButton\_Click()

'Callculates...

Dim index As Integer, arrayCount As Integer, amount\_neg As Integer, amount\_poz As Integer

Dim alfa\_value As Single, first\_row As Integer, column\_adress As Integer

Dim n() As Variant, temp As Single, sample\_median As Double, pop\_median As Double

amount\_neg = 0

amount\_poz = 0

index = 0

arrayCount = 0

alfa\_value = alfaText.Text

first\_row = FirstRowText.Text

column\_adress = ColumnAdressText.Text

pop\_median = AMText.Text

‘Kullanıcının belirlediği sütunda belirlediği satırdan başlayarak boş hücreye denk gelene kadar hücrelerin içine bakılır ve her dolu hücre için arrayCount değişkeni arttırılır. Bu işlem örneklem hacmi için tanımlanacak olan “n()” dizisinin boyutunu belirlemek adına yapılır.

Do

arrayCount = arrayCount + 1

Loop Until IsEmpty(ActiveSheet.Cells(first\_row + (arrayCount - 1), column\_adress))

arrayCount = arrayCount - 1

ReDim n(arrayCount)

‘Veriler “n()” dizisine atanır.

Do

n(index) = ActiveSheet.Cells(first\_row + index, column\_adress)

index = index + 1

Loop Until IsEmpty(ActiveSheet.Cells(first\_row + index, column\_adress))

‘Örneklemi temsil eden “n()” dizisinin değerleri burada küçükten büyüğe sıralanır.

For i = 1 To (arrayCount - 1)

For j = 0 To (arrayCount - 2)

If n(i) < n(j) Then

temp = n(i)

n(i) = n(j)

n(j) = temp

End If

Next j

Next i

‘Medyan belirlenmek üzere önce örneklem hacminin tek mi yoksa çift mi olduğuna bakılır. Ardından örneklem medyanı hesaplanır.

If arrayCount Mod 2 = 0 Then

sample\_median = (n(arrayCount / 2) + n((arrayCount + 2) / 2)) / 2

Else

sample\_median = n((arrayCount + 1) / 2)

End If

‘Burada anakütle meydanından küçük olan değerlerin sayısını “amount\_neg” isimli değişken tutar. Anakütle medyanından büyük olan değerlerin sayısını ise “amount\_poz” isimli değişken tutar. Anakütle medyanına eşit olan değerler işaret testinin kuralı gereği hesaba katılmaz.

For i = 0 To (arrayCount - 1)

If n(i) < pop\_median Then

amount\_neg = amount\_neg + 1

ElseIf n(i) = pop\_median Then

Else

amount\_poz = amount\_poz + 1

End If

Next i

‘Kullanıcı tarafından seçilen alternatif hipotez örneklem medyanının anakütle madyanına eşit olmadığını savunan hipotez ise buradaki “If” bloğu çalışır.

If Option\_NotEqual.Value = True Then

‘Anakütle medyanından büyük olan değerlerin sayısı küçük olanlara göre daha az ise buradaki “If” bloğu çalışır.

If amount\_poz < amount\_neg Then

'S+

sumProbality = pCalcIsSmall((amount\_poz), (amount\_neg), 0.5)

‘Hesaplanan olasılık değeri alfa değerinden küçük veya eşit ise buradaki “If” bloğu çalıştırılır ve H0’ın reddine karar verilir.

If sumProbality <= (alfa\_value / 2) Then

'H0 Red

SMText.Text = "Örneklem Medyanı: " & sample\_median

OutText.Text = "Yokluk Hipotezi Reddedildi."

MsgBox "n : " & (amount\_poz + amount\_neg) & vbTab & "x : " & amount\_poz & vbTab & "(n-x) : " & amount\_neg & vbCrLf & " " & vbCrLf & "P(X<=" & amount\_poz & ") = " & VBA.Round(sumProbality, 4) & " < alfa = " & (alfa\_value / 2) & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde örneklem medyanı ile ana kütle medyanı arasında istatiksel olarak anlamlı bir fark vardır." & vbCrLf & " " & vbCrLf & "Bu örneklemin medyanı anlamlı bir biçimde varsayılan anakütle medyanından daha küçüktür.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

‘Hesaplanan olasılık değeri alfa değerinden büyük ise buradaki “If” bloğu çalıştırılır ve H0’ın kabul edildiğine karar verilir.

Else

'H0 Kabul

SMText.Text = "Örneklem Medyanı: " & sample\_median

OutText.Text = "Yokluk Hipotezi Reddedilemez."

MsgBox "n : " & (amount\_poz + amount\_neg) & vbTab & "x : " & amount\_poz & vbTab & "(n-x) : " & amount\_neg & vbCrLf & " " & vbCrLf & "P(X<=" & amount\_poz & ") = " & VBA.Round(sumProbality, 4) & " > alfa = " & (alfa\_value / 2) & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde örneklem medyanı ile ana kütle medyanı arasında istatiksel olarak anlamlı bir fark yoktur." & vbCrLf & " " & vbCrLf & "Bu örneklemin medyanı anlamlı bir biçimde varsayılan anakütle medyanından daha büyüktür.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

‘Anakütle medyanından küçük olan değerlerin sayısı büyük olanlara göre daha az ise buradaki “If” bloğu çalışır.

Else

'S-

sumProbality = pCalcIsSmall((amount\_neg), (amount\_poz), 0.5)

‘Hesaplanan olasılık değeri alfa değerinden küçük veya eşit ise buradaki “If” bloğu çalıştırılır ve H0’ın reddine karar verilir.

If sumProbality <= (alfa\_value / 2) Then

'H0 Red

SMText.Text = "Örneklem Medyanı: " & sample\_median

OutText.Text = "Yokluk Hipotezi Reddedildi."

MsgBox "n : " & (amount\_neg + amount\_poz) & vbTab & "x : " & amount\_neg & vbTab & "(n-x) : " & amount\_poz & vbCrLf & " " & vbCrLf & "P(X<=" & amount\_neg & ") = " & VBA.Round(sumProbality, 4) & " < alfa = " & (alfa\_value / 2) & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde örneklem medyanı ile ana kütle medyanı arasında istatiksel olarak anlamlı bir fark vardır." & vbCrLf & " " & vbCrLf & "Bu örneklemin medyanı anlamlı bir biçimde varsayılan anakütle medyanından daha büyüktür.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

‘Hesaplanan olasılık değeri alfa değerinden büyük ise buradaki “If” bloğu çalıştırılır ve H0’ın kabul edildiğine karar verilir.

Else

'H0 Kabul

SMText.Text = "Örneklem Medyanı: " & sample\_median

OutText.Text = "Yokluk Hipotezi Reddedilemez."

MsgBox "n : " & (amount\_neg + amount\_poz) & vbTab & "x : " & amount\_neg & vbTab & "(n-x) : " & amount\_poz & vbCrLf & " " & vbCrLf & "P(X<=" & amount\_neg & ") = " & VBA.Round(sumProbality, 4) & " > alfa = " & (alfa\_value / 2) & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde örneklem medyanı ile ana kütle medyanı arasında istatiksel olarak anlamlı bir fark yoktur." & vbCrLf & " " & vbCrLf & "Bu örneklemin medyanı anlamlı bir biçimde varsayılan anakütle medyanından daha küçüktür.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

End If

‘Kullanıcı tarafından seçilen alternatif hipotez örneklem medyanının anakütle medyanından küçük olduğunu savunan hipotez ise buradaki “If” bloğu çalışır.

ElseIf Option\_LessThen.Value = True Then

sumProbality = pCalcIsSmall(amount\_poz, amount\_neg, 0.5)

‘Hesaplanan olasılık değeri alfa değerinden küçük veya eşit ise buradaki “If” bloğu çalıştırılır ve H0’ın reddine karar verilir.

If sumProbality <= alfa\_value Then

'H0 Red

SMText.Text = "Örneklem Medyanı: " & sample\_median

OutText.Text = "Yokluk Hipotezi Reddedildi."

MsgBox "n : " & (amount\_poz + amount\_neg) & vbTab & "x : " & amount\_poz & vbTab & "(n-x) : " & amount\_neg & vbCrLf & " " & vbCrLf & "P(X=<" & amount\_poz & ") = " & VBA.Round(sumProbality, 4) & " < alfa = " & alfa\_value & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde örneklem medyanı, ana kütle medyanından istatiksel olarak anlamlı bir şekilde daha küçüktür.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

‘Hesaplanan olasılık değeri alfa değerinden büyük ise buradaki “If” bloğu çalıştırılır ve H0’ın kabul edildiğine karar verilir.

Else

'H0 Kabul

SMText.Text = "Örneklem Medyanı: " & sample\_median

OutText.Text = "Yokluk Hipotezi Reddedilemez."

MsgBox "n : " & (amount\_poz + amount\_neg) & vbTab & "x : " & amount\_poz & vbTab & "(n-x) : " & amount\_neg & vbCrLf & " " & vbCrLf & "P(X=<" & amount\_poz & ") = " & VBA.Round(sumProbality, 4) & " > alfa = " & alfa\_value & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde örneklem medyanı ile ana kütle medyanı arasında istatiksel olarak anlamlı bir fark yoktur.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

‘Kullanıcı tarafından seçilen alternatif hipotez örneklem medyanının anakütle medyanından büyük olduğunu savunan hipotez ise buradaki “If” bloğu çalışır.

Else

sumProbality = pCalcIsBig(amount\_poz, amount\_neg, 0.5)

‘Hesaplanan olasılık değeri alfa değerinden küçük veya eşit ise buradaki “If” bloğu çalıştırılır ve H0’ın reddine karar verilir.

If sumProbality <= alfa\_value Then

'H0 Red

SMText.Text = "Örneklem Medyanı: " & sample\_median

OutText.Text = "Yokluk Hipotezi Reddedildi."

MsgBox "n : " & (amount\_poz + amount\_neg) & vbTab & "x : " & amount\_poz & vbTab & "(n-x) : " & amount\_neg & vbCrLf & " " & vbCrLf & "P(X=>" & amount\_poz & ") = " & VBA.Round(sumProbality, 4) & " < alfa = " & alfa\_value & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde örneklem medyanı, ana kütle medyanından istatiksel olarak anlamlı bir şekilde daha büyüktür.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

‘Hesaplanan olasılık değeri alfa değerinden büyük ise buradaki “If” bloğu çalıştırılır ve H0’ın kabul edildiğine karar verilir.

Else

'H0 Kabul

SMText.Text = "Örneklem Medyanı: " & sample\_median

OutText.Text = "Yokluk Hipotezi Reddedilemez."

MsgBox "n : " & (amount\_poz + amount\_neg) & vbTab & "x : " & amount\_poz & vbTab & "(n-x) : " & amount\_neg & vbCrLf & " " & vbCrLf & "P(X=>" & amount\_poz & ") = " & VBA.Round(sumProbality, 4) & " > alfa = " & alfa\_value & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa\_value) \* 100) & " güven düzeyinde örneklem medyanı ile ana kütle medyanı arasında istatiksel olarak anlamlı bir fark yoktur.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

End If

End Sub

###### EK 27

İşaret testinde alternatif hipotezin eşit olmadığı savının olasılığını “” hesaplayan fonksiyon (*pCalcNullHyp*’) için kodlar aşağıda verilmiştir.

Public Function pCalcNullHyp(ByRef x As Integer, ByRef y As Integer, testProbability As Single) As Double

Dim Combination As Double, nFac As Variant, xFac As Variant, diffFac As Variant

Dim q\_prob As Double, p\_prob As Double, Calc As Double, probCollec As Double

Dim i As Integer, j As Integer, k As Integer, npq As Single, sample\_size As Integer, z\_calc As Double

Dim b As Integer

sample\_size = x + y

npq = sample\_size \* (x / sample\_size) \* (y / sample\_size)

‘Örneklem hacmi, p olasılık değeri ve q olasılık değeri çarpımları 9’a eşit veya 9’dan büyük ise süreklilik düzeltmesi yapılır ve normal dağılım üzerinden hesap yapılır.

If npq >= 9 Then

'Süreklilik

If x > (testProbability \* sample\_size) Then

z\_calc = ((x - 0.5) - (sample\_size / 2)) / Sqr(sample\_size / 4)

Else

z\_calc = ((x + 0.5) - (sample\_size / 2)) / Sqr(sample\_size / 4)

End If

pCalcNullHyp = WorksheetFunction.Norm\_S\_Dist(z\_calc, True)

‘Örneklem hacmi, p olasılık değeri ve q olasılık değeri çarpımları 9’dan küçük ise binom dağılımı olasılık yoğunluk fonksiyonu ile hesaplama yapılır.

Else

p\_prob = testProbability

q\_prob = (1 - testProbability)

nFac = 1

xFac = 1

diffFac = 1

For i = 1 To sample\_size

nFac = nFac \* i

Next i

For b = x To 0 Step -1

xFac = 1

diffFac = 1

For j = 1 To x

xFac = xFac \* j

Next j

For k = 1 To y

diffFac = diffFac \* k

Next k

Combination = (nFac / (diffFac \* xFac))

Calc = (p\_prob ^ x) \* (q\_prob ^ y)

probCollec = probCollec + (Combination \* Calc)

Next b

pCalcNullHyp = probCollec

End If

End Function

###### EK 28

Public Function pCalcIsSmall(ByRef x As Integer, ByRef y As Integer, ByRef testProbability As Single) As Double

Dim Combination As Double, nFac As Variant, xFac As Variant, diffFac As Variant

Dim q\_prob As Double, p\_prob As Double, Calc As Double, probCollec As Double

Dim instant\_x As Integer, instant\_y As Integer, i As Integer, j As Integer, k As Integer

Dim npq As Single, sample\_size As Integer, z\_calc As Double, b As Integer

sample\_size = x + y

npq = sample\_size \* (x / sample\_size) \* (y / sample\_size)

‘Örneklem hacmi, p olasılık değeri ve q olasılık değeri çarpımları 9’a eşit veya 9’dan büyük ise süreklilik düzeltmesi yapılır ve normal dağılım üzerinden hesap yapılır.

If npq >= 9 Then

'Süreklilik

If x > (testProbability \* sample\_size) Then

z\_calc = ((x - 0.5) - (sample\_size / 2)) / Sqr(sample\_size / 4)

Else

z\_calc = ((x + 0.5) - (sample\_size / 2)) / Sqr(sample\_size / 4)

End If

pCalcIsSmall = WorksheetFunction.Norm\_S\_Dist(z\_calc, True)

‘Örneklem hacmi, p olasılık değeri ve q olasılık değeri çarpımları 9’dan küçük ise binom dağılımı olasılık yoğunluk fonksiyonu ile hesaplama yapılır.

Else

p\_prob = testProbability

q\_prob = (1 - testProbability)

nFac = 1

xFac = 1

diffFac = 1

For i = 1 To (x + y)

nFac = nFac \* i

Next i

For b = x To 0 Step -1

instant\_x = b

instant\_y = ((x + y) - b)

xFac = 1

diffFac = 1

For j = 1 To instant\_x

xFac = xFac \* j

Next j

For k = 1 To instant\_y

diffFac = diffFac \* k

Next k

Combination = (nFac / (diffFac \* xFac))

Calc = (p\_prob ^ instant\_x) \* (q\_prob ^ instant\_y)

probCollec = probCollec + (Combination \* Calc)

Next b

pCalcIsSmall = probCollec

End If

End Function

###### EK 29

Public Function pCalcIsBig(ByRef x As Integer, ByRef y As Integer, testProbability As Single) As Double

Dim Combination As Double, nFac As Variant, xFac As Variant, diffFac As Variant

Dim q\_prob As Double, p\_prob As Double, Calc As Double, probCollec As Double

Dim instant\_x As Integer, instant\_y As Integer, i As Integer, j As Integer, k As Integer

Dim npq As Single, sample\_size As Integer, z\_calc As Double, b As Integer

sample\_size = x + y

npq = sample\_size \* (x / sample\_size) \* (y / sample\_size)

‘Örneklem hacmi, p olasılık değeri ve q olasılık değeri çarpımları 9’a eşit veya 9’dan büyük ise süreklilik düzeltmesi yapılır ve normal dağılım üzerinden hesap yapılır.

If npq >= 9 Then

'Süreklilik

If x > (testProbability \* sample\_size) Then

z\_calc = ((x - 0.5) - (sample\_size / 2)) / Sqr(sample\_size / 4)

Else

z\_calc = ((x + 0.5) - (sample\_size / 2)) / Sqr(sample\_size / 4)

End If

pCalcIsBig = (1 - WorksheetFunction.Norm\_S\_Dist(z\_calc, True))

‘Örneklem hacmi, p olasılık değeri ve q olasılık değeri çarpımları 9’dan küçük ise binom dağılımı olasılık yoğunluk fonksiyonu ile hesaplama yapılır.

Else

p\_prob = testProbability

q\_prob = (1 - testProbability)

nFac = 1

xFac = 1

diffFac = 1

For i = 1 To (x + y)

nFac = nFac \* i

Next i

For b = x To (x + y)

instant\_x = b

instant\_y = ((x + y) - b)

xFac = 1

diffFac = 1

For j = 1 To instant\_x

xFac = xFac \* j

Next j

For k = 1 To instant\_y

diffFac = diffFac \* k

Next k

Combination = (nFac / (diffFac \* xFac))

Calc = (p\_prob ^ instant\_x) \* (q\_prob ^ instant\_y)

probCollec = probCollec + (Combination \* Calc)

Next b

pCalcIsBig = probCollec

End If

End Function

###### EK 30

Form nesnesi kullanılmadan yapılan İşaret testi için fonksiyona ilişkin kodlar aşağıda verilmiştir.

Function IsaretTesti(data As Range, Optional pop\_median As String, Optional alfa\_value As String) As Variant

Dim rowCount As Integer, columnCount As Integer, alfa As Single

Dim median As Single, n() As Variant, nCount As Integer

Dim index As Integer, temp As Single, Hyphothesis As String

Dim probability As Variant, sample\_median As Single

Dim amount\_poz As Integer, amount\_neg As Integer, sumProbability As Double

index = 1

rowCount = data.Rows.Count

columnCount = data.Columns.Count

‘Burada kullanıcı tarafından seçilen hücre sayısı kontrol edilir.

If (rowCount \* columnCount) < 4 Then

MsgBox "Lütfen yeterli sayıda veri seçin.", vbOKOnly

Exit Function

End If

‘Burada kullanıcı tarafından seçilen hücrelerin sayıları bir değişkene atanarak o değişken ile örneklem verilerini tutacak olan dizi yeniden boyutlandırılır.

nCount = rowCount \* columnCount

ReDim n(1 To nCount)

‘Eğer anakütle medyanı parametresi girilmemiş ise teste başlamadan önce bu parametre değeri kullanıcıya sorulur.

If IsEmpty(pop\_median) Or pop\_median = "" Or pop\_median = " " Then

pop\_median = InputBox("Lütfen anakütle medyanını giriniz!", "Anakütle Medyanı", "-")

If IsNumeric(pop\_median) Then

Else

Do

pop\_median = InputBox("Lütfen anakütle medyanını sayısal olarak giriniz!", "Anakütle Medyanı", "-")

Loop Until IsNumeric(pop\_median)

End If

End If

‘Eğer alfa değeri girilmemişse burada kontrol edilir ve kullanıcıya sorulur.

If IsEmpty(alfa\_value) Or alfa\_value = "" Or alfa\_value = " " Then

alfa\_value = InputBox("Lütfen anlam düzeyini giriniz!", "Anlam Düzeyi", 0.05)

If IsNumeric(alfa\_value) And alfa\_value < 1 And alfa\_value > 0 Then

Else

Do

alfa\_value = InputBox("Lütfen anlam düzeyini doğru bir biçimde giriniz!", "Anlam Düzeyi", 0.05)

Loop Until IsNumeric(alfa\_value) And alfa\_value < 1 And alfa\_value > 0

End If

End If

alfa = CSng(alfa\_value)

median = CSng(pop\_median)

‘Kullanıcının seçtiği örneklem verileri “n()” dizisine atanır.

For i = 1 To rowCount

For j = 1 To columnCount

n(index) = data.Cells(i, j).Value

index = index + 1

Next j

Next i

“Burada örneklem verilerini tutan “n()” dizisindeki veriler küçükten büyüğe sıralanır.

For i = 2 To (nCount)

For j = 1 To (nCount - 1)

If n(i) < n(j) Then

temp = n(i)

n(i) = n(j)

n(j) = temp

End If

Next j

Next i

“Örneklem medyanı belirlenir.

If nCount Mod 2 = 0 Then

sample\_median = (n(nCount / 2) + n((nCount + 2) / 2)) / 2

Else

sample\_median = n((nCount + 1) / 2)

End If

“Anakütle medyanından küçük değerlerin sayısını “amount\_neg”, büyük olan değerlerin sayısını “amount\_poz” değişkeni tutar. Anakütle medyanına eşit olan değerler çıkartılır.

For i = 1 To (nCount)

If n(i) < median Then

amount\_neg = amount\_neg + 1

ElseIf n(i) = median Then

Else

amount\_poz = amount\_poz + 1

End If

Next i

‘Kullanıcıya bu aşamada hangi alternatif hipotezi test edeceği sorulur. Bunun için “InputBox” kullanılır.

hypothesis = InputBox("Lütfen hipotez seçimi yapınız." & vbCrLf & vbCrLf & "Alternatif hipoteziniz bu örneklemin daha küçük bir medyana sahip olduğunu savunuyorsa 1 değerini giriniz." & vbCrLf & vbCrLf & "Alternatif hipoteziniz bu örneklemin daha büyük bir medyana sahip olduğunu savunuyorsa 2 değerini giriniz." & vbCrLf & vbCrLf & "Alternatif hipoteziniz bu örneklemin medyanının bilinen anakütle medyanına eşit olmadığını savunuyorsa 3 değerini giriniz.", "Hipotez Seçimi", 3)

‘Eğer örneklem medyanının anakütle medyanına eşit olmadığını savunan alternatif hipotez seçilmişse bu “If” bloğu çalışır.

If hypothesis = 3 Then

If amount\_poz < amount\_neg Then

'S+

sumProbability = IsaretForm.pCalcIsSmall((amount\_poz), (amount\_neg), 0.5)

If sumProbability <= (alfa / 2) Then

'H0 Red

MsgBox "n : " & (amount\_poz + amount\_neg) & vbTab & "x : " & amount\_poz & vbTab & "(n-x) : " & amount\_neg & vbCrLf & " " & vbCrLf & "P(X<=" & amount\_poz & ") = " & VBA.Round(sumProbability, 4) & " < alfa = " & (alfa / 2) & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa) \* 100) & " güven düzeyinde örneklem medyanı ile ana kütle medyanı arasında istatiksel olarak anlamlı bir fark vardır." & vbCrLf & " " & vbCrLf & "Bu örneklemin medyanı anlamlı bir biçimde varsayılan anakütle medyanından daha küçüktür.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

Else

'H0 Kabul

MsgBox "n : " & (amount\_poz + amount\_neg) & vbTab & "x : " & amount\_poz & vbTab & "(n-x) : " & amount\_neg & vbCrLf & " " & vbCrLf & "P(X<=" & amount\_poz & ") = " & VBA.Round(sumProbability, 4) & " > alfa = " & (alfa / 2) & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa) \* 100) & " güven düzeyinde örneklem medyanı ile ana kütle medyanı arasında istatiksel olarak anlamlı bir fark yoktur.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

Else

'S-

sumProbability = IsaretForm.pCalcIsSmall((amount\_neg), (amount\_poz), 0.5)

If sumProbability <= (alfa / 2) Then

'H0 Red

MsgBox "n : " & (amount\_neg + amount\_poz) & vbTab & "x : " & amount\_neg & vbTab & "(n-x) : " & amount\_poz & vbCrLf & " " & vbCrLf & "P(X<=" & amount\_neg & ") = " & VBA.Round(sumProbability, 4) & " < alfa = " & (alfa / 2) & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa) \* 100) & " güven düzeyinde örneklem medyanı ile ana kütle medyanı arasında istatiksel olarak anlamlı bir fark vardır." & vbCrLf & " " & vbCrLf & "Bu örneklemin medyanı anlamlı bir biçimde varsayılan anakütle medyanından daha büyüktür.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

Else

'H0 Kabul

MsgBox "n : " & (amount\_neg + amount\_poz) & vbTab & "x : " & amount\_neg & vbTab & "(n-x) : " & amount\_poz & vbCrLf & " " & vbCrLf & "P(X<=" & amount\_neg & ") = " & VBA.Round(sumProbability, 4) & " > alfa = " & (alfa / 2) & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa) \* 100) & " güven düzeyinde örneklem medyanı ile ana kütle medyanı arasında istatiksel olarak anlamlı bir fark yoktur.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

End If

‘Eğer örneklem medyanının anakütle medyanından küçük olduğunu savunan alternatif hipotez seçilmişse bu “If” bloğu çalışır.

ElseIf hypothesis = 1 Then

sumProbability = IsaretForm.pCalcIsSmall(amount\_poz, amount\_neg, 0.5)

If sumProbability < alfa Then

'H0 Red

MsgBox "n : " & (amount\_poz + amount\_neg) & vbTab & "x : " & amount\_poz & vbTab & "(n-x) : " & amount\_neg & vbCrLf & " " & vbCrLf & "P(X=<" & amount\_poz & ") = " & VBA.Round(sumProbability, 4) & " < alfa = " & alfa & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa) \* 100) & " güven düzeyinde örneklem medyanı, ana kütle medyanından istatiksel olarak anlamlı bir şekilde daha küçüktür.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

Else

'H0 Kabul

MsgBox "n : " & (amount\_poz + amount\_neg) & vbTab & "x : " & amount\_poz & vbTab & "(n-x) : " & amount\_neg & vbCrLf & " " & vbCrLf & "P(X=<" & amount\_poz & ") = " & VBA.Round(sumProbability, 4) & " > alfa = " & alfa & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa) \* 100) & " güven düzeyinde örneklem medyanı ile ana kütle medyanı arasında istatiksel olarak anlamlı bir fark yoktur.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

‘Eğer örneklem medyanının anakütle medyanından büyük olduğunu savunan alternatif hipotez seçilmişse bu “If” bloğu çalışır.

ElseIf hypothesis = 2 Then

sumProbability = IsaretForm.pCalcIsBig(amount\_poz, amount\_neg, 0.5)

If sumProbability < alfa Then

'H0 Red

MsgBox "n : " & (amount\_poz + amount\_neg) & vbTab & "x : " & amount\_poz & vbTab & "(n-x) : " & amount\_neg & vbCrLf & " " & vbCrLf & "P(X=>" & amount\_poz & ") = " & VBA.Round(sumProbability, 4) & " < alfa = " & alfa & " olduğundan Yokluk Hipotezi Reddedilir." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa) \* 100) & " güven düzeyinde örneklem medyanı, ana kütle medyanından istatiksel olarak anlamlı bir şekilde daha büyüktür.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

Else

'H0 Kabul

MsgBox "n : " & (amount\_poz + amount\_neg) & vbTab & "x : " & amount\_poz & vbTab & "(n-x) : " & amount\_neg & vbCrLf & " " & vbCrLf & "P(X=>" & amount\_poz & ") = " & VBA.Round(sumProbability, 4) & " > alfa = " & alfa & " olduğundan Yokluk Hipotezi Reddedilemez." & vbCrLf & " " & vbCrLf & "%" & ((1 - alfa) \* 100) & " güven düzeyinde örneklem medyanı ile ana kütle medyanı arasında istatiksel olarak anlamlı bir fark yoktur.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

Else

Exit Function

End If

IsaretTesti = VBA.Round(probability, 4)

End Function

###### EK 31

Ki-kare uyum iyilği testinde form arayüzünü açan fonksiyon aşağıdaki gibidir.

‘Form Nesnesini Açan Fonksiyon

Function KiKareTestiForm()

IsaretForm.Show

End Function

###### EK 32

Ki-kare uyum iyiliği testinde veri adresi (sütun) değişkeni için oluşturulan ‘*TextBox*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı ColumnAdressText isimli TextBox nesnesini seçtiğinde TextBox nesnesinin içi temizlenir.

Private Sub ColumnAdressText\_Enter()

ColumnAdressText.Text = ""

End Sub

‘Kullanıcı ColumnAdressText isimli TextBox nesnesine değer girdiğinde yazı rengi siyah olur ve yazı tipinin italik özelliği kapatılır.

Private Sub ColumnAdressText\_Change()

ColumnAdressText.ForeColor = RGB(0,0,0)

ColumnAdressText.Font.Italic = False

End Sub

‘Kullanıcı ColumnAdressText isimli TextBox nesnesine sayısal bir değer girmez ise nesneden çıkamaz ve sütun adresini sayısal olarak girmesi gerektiğini aktaran bir bilgilendirme mesajı alır.

Private Sub ColumnAdressText\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

If IsNumeric(ColumnAdressText.Value) Then

Cancel = False

Else

Cancel = True

MsgBox ("Lütfen yalnızca sayı giriniz!" & vbCrLf & "'A' için 1" & vbCrLf & "'B' için 2" & vbCrLf & "'C' için 3" & vbCrLf & "." & vbCrLf & "." & vbCrLf & ".")

ColumnAdressText.Text = “1”

End If

End Sub

###### EK 33

Ki-kare uyum iyiliği testinde veri adresi (sütun) değişkeni için oluşturulan ‘*CommandButton*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı infButton\_ColAdress butonuna tıkladığında bir MessageBox kutusu açılır ve information değişkeni ile mesaj aktarılır.

Private Sub infButton\_ColAdress\_Click()

Dim infromation As String, title As String

information = "Verilerinizin bulunduğu sütunu girin. A için 1, B için 2, C için 3... değerlerini giriniz."

title = "Bilgilendirme (Veri Adresi)"

MsgBox information, VbOkOnly, title

End Sub

###### EK 34

Ki-kare uyum iyiliği testinde başlangıç satırı değişkeni için oluşturulan ‘*TextBox*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı FirstRowText isimli TextBox nesnesini seçtiğinde TextBox nesnesinin içi temizlenir.

Private Sub FirstRowText\_Enter()

FirstRowText.Text = ""

End Sub

‘Kullanıcı FirstRowText isimli TextBox nesnesine değer girdiğinde yazı rengi siyah olur ve yazı tipinin italik özelliği kapatılır.

Private Sub FirstRowText\_Change()

FirstRowText.ForeColor = RGB(0,0,0)

FirstRowText.Font.Italic = False

End Sub

‘Kullanıcı FirstRowText isimli TextBox nesnesine sayısal bir değer girmez ise nesneden çıkamaz. Varsayılan değer 1 olur ve ilk satır adresini sayısal olarak girmesi gerektiğini aktaran bir bilgilendirme mesajı alır.

Private Sub FirstRowText\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

If IsNumeric(FirstRowText.Value) Then

Cancel = False

Else

Cancel = True

MsgBox ("Lütfen yalnızca sayısal bir değer giriniz!

FirstRowText.Text = “1”

End If

End Sub

###### EK 35

Ki-kare uyum iyiliği testinde başlangıç satırı değişkeni için oluşturulan ‘*CommandButton*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı infButton\_firstRow butonuna tıkladığında bir MessageBox kutusu açılır ve information değişkeni ile mesaj aktarılır.

Private Sub infButton\_firstRow\_Click()

Dim infromation As String, title As String

information = “Verilerinizin bulunduğu sütunda, verilerinizin kaçıncı satırda başladığını giriniz."

title = "Bilgilendirme (Başlangıç Satırı)"

MsgBox information, VbOkOnly, title

End Sub

###### EK 36

Anlamlılık düzeyi için oluşturulan ‘*TextBox*’ nesnesine ait kodlar aşağıdaki gibidir.

‘Kullanıcı alfaText isimli TextBox nesnesini seçtiğinde TextBox nesnesinin içi temizlenir.

Private Sub alfaText\_Enter()

alfaText.Text = ""

End Sub

‘Kullanıcı alfaText isimli TextBox nesnesine değer girdiğinde yazı rengi siyah olur ve yazı tipinin italik özelliği kapatılır.

Private Sub alfaText\_Change()

alfaText.ForeColor = RGB(0,0,0)

alfaText.Font.Italic = False

End Sub

‘Kullanıcı alfaText isimli TextBox nesnesine sayısal bir değer girmez ise nesneden çıkamaz. Varsayılan değer 0,05 olur ve 𝛼 anlamlılık düzeyini sayısal olarak girmesi gerektiğini aktaran bir bilgilendirme mesajı alır.

Private Sub alfaText\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

If IsNumeric(alfaText.Value) Then

Cancel = False

If alfaText.Value < 0 Or alfaText.Value > 1 Then

Cansel = True

MsgBox ("Lütfen 0 ile 1 arası bir değer giriniz!")

alfaText.Text = "0,05"

End If

Else

Cancel = True

MsgBox ("Lütfen 0 ile 1 arası bir değer giriniz!")

alfaText.Text = “0,05”

End If

End Sub

###### EK 37

Ki-kare uyum iyiliği testi için ‘*CalcButton*’ nesnesinin ‘*Click*’ olay yordamına ilişkin kodlar aşağıda verilmiştir.

Private Sub CalcButton\_Click()

Dim index As Integer, arrayCount As Integer, expected\_value As Integer

Dim alfa\_value As Single, first\_row As Integer, column\_adress As Integer

Dim n() As Variant, total As Double, df As Integer, calcValue As Double

Dim tableValue As Double, temp As Double

expected\_value = 0

index = 0

arrayCount = 0

alfa\_value = alfaText.Text

first\_row = FirstRowText.Text

column\_adress = ColumnAdressText.Text

‘Burada n dizisi için tüm hücreler boş hücreye denk gelene dek kontrol edilir.

Do

arrayCount = arrayCount + 1

Loop Until IsEmpty(ActiveSheet.Cells(first\_row + (arrayCount - 1), column\_adress))

arrayCount = arrayCount - 1

‘n dizisi yeniden boyutlandırılır.

ReDim n(arrayCount)

‘Her bir frekans n dizisinin bir elemanına atanır.

Do

n(index) = ActiveSheet.Cells(first\_row + index, column\_adress)

index = index + 1

Loop Until IsEmpty(ActiveSheet.Cells(first\_row + index, column\_adress))

‘Frekanslar toplamı elde edilir.

For i = 0 To (arrayCount - 1)

total = total + n(i)

Next i

‘Beklenen değer, frekanslar toplamının gözlem sayısına bölümü ile elde edilir.

expected\_value = total / arrayCount

‘Serbestlik derecesi belirlenir.

df = arrayCount – 1

‘Tablo değeri VBA fonksiyonu sayesinde hesaplanır.

tableValue = WorksheetFunction.ChiSq\_Inv\_RT(alfa\_value, df)

'Ki-Kare hesap değeri hesaplanır.

For i = 0 To arrayCount - 1

temp = (((n(i) - expected\_value) ^ 2) / expected\_value)

calcValue = calcValue + temp

Next i

‘Hesaplanan ki-kare değeri tablo değerinden büyük ise buraya girilir ve yokluk hipotezi reddedilir.

If calcValue > tableValue Then

'H0 Red

OutText.Text = "Yokluk Hipotezi Reddedildi."

MsgBox "Ki-Kare hesap değeri = " & calcValue & vbCrLf & vbCrLf & "Ki-Kare tablo değeri = " & tableValue & vbCrLf & vbCrLf & calcValue & " > " & tableValue & " olduğundan yokluk hipotezi reddedilir." & vbCrLf & vbCrLf & " Gözlenen frekanslar, beklenen frekanslardan farklıdır.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

‘Hesaplanan ki-kare değeri tablo değerinden küçük ise buraya girilir ve alternatif hipotez reddedilir.

Else

'H0 Kabul

OutText.Text = "Yokluk Hipotezi Reddedilemez."

MsgBox "Ki-Kare hesap değeri = " & calcValue & vbCrLf & vbCrLf & "Ki-Kare tablo değeri = " & tableValue & vbCrLf & vbCrLf & calcValue & " < " & tableValue & " olduğundan yokluk hipotezi reddedilemez." & vbCrLf & vbCrLf & " Gözlenen frekanslar, beklenen frekanslardan istatiksel olarak farklı değildir.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

End Sub

###### EK 38

Form nesnesi kullanılmadan yapılan Ki-kare uyum iyiliği testi için fonksiyona ilişkin kodlar aşağıda verilmiştir.

Function KiKareTesti(data As Range, Optional alfa\_value As String) As Variant

Dim rowCount As Integer, columnCount As Integer, alfa As Single, n() As Variant

Dim index As Integer, temp As Single, total As Double, expected\_value As Double

Dim calcValue As Double, tableValue As Double, df As Integer, nCount As Integer

index = 1

rowCount = data.Rows.Count

columnCount = data.Columns.Count

‘Kaç gözlem birimi olduğu belirlenir.

nCount = rowCount \* columnCount

‘n dizisi yeniden boyutlandırılır.

ReDim n(1 To nCount)

‘Anlamlılık düzeyi parametresi girilmediyse kullanıcı bilgilendirilir ve doğru bir anlam düzeyi değeri girene kadar kullanıcıdan yeni bri giriş yapması istenir.

If IsEmpty(alfa\_value) Or alfa\_value = "" Or alfa\_value = " " Then

alfa\_value = InputBox("Lütfen anlam düzeyini giriniz!", "Anlam Düzeyi", 0.05)

If IsNumeric(alfa\_value) And alfa\_value < 1 And alfa\_value > 0 Then

Else

Do

alfa\_value = InputBox("Lütfen anlam düzeyini doğru bir biçimde giriniz!", "Anlam Düzeyi", 0.05)

Loop Until IsNumeric(alfa\_value) And alfa\_value < 1 And alfa\_value > 0

End If

End If

alfa = CSng(alfa\_value)

‘Frekanslar toplamı hesaplanır.

For i = 1 To rowCount

For j = 1 To columnCount

n(index) = data.Cells(i, j).Value

total = total + n(index)

index = index + 1

Next j

Next i

‘Beklenen değer ve serbestlik derecesi hesaplanır.

expected\_value = total / nCount

df = nCount – 1

‘Hazır bir VBA fonksiyonu ile tablo değeri hesaplanır.

tableValue = WorksheetFunction.ChiSq\_Inv\_RT(alfa\_value, df)

'Ki-Kare hesap değeri hesaplanır.

For i = 1 To nCount

temp = (((n(i) - expected\_value) ^ 2) / expected\_value)

calcValue = calcValue + temp

Next i

‘Hesaplanan ki-kare değeri tablo değerinden büyük ise buraya girilir ve yokluk hipotezi reddedilir.

If calcValue > tableValue Then

'H0 Red

MsgBox "Ki-Kare hesap değeri = " & calcValue & vbCrLf & vbCrLf & "Ki-Kare tablo değeri = " & tableValue & vbCrLf & vbCrLf & calcValue & " > " & tableValue & " olduğundan yokluk hipotezi reddedilir." & vbCrLf & vbCrLf & " Gözlenen frekanslar, beklenen frekanslardan farklıdır.", vbOKOnly, "Yokluk Hipotezi Reddedildi."

‘Hesaplanan ki-kare değeri tablo değerinden küçük ise buraya girilir ve alternatif hipotez reddedilir.

Else

'H0 Kabul

MsgBox "Ki-Kare hesap değeri = " & calcValue & vbCrLf & vbCrLf & "Ki-Kare tablo değeri = " & tableValue & vbCrLf & vbCrLf & calcValue & " < " & tableValue & " olduğundan yokluk hipotezi reddedilemez." & vbCrLf & vbCrLf & " Gözlenen frekanslar, beklenen frekanslardan istatiksel olarak farklı değildir.", vbOKOnly, "Yokluk Hipotezi Reddedilemez."

End If

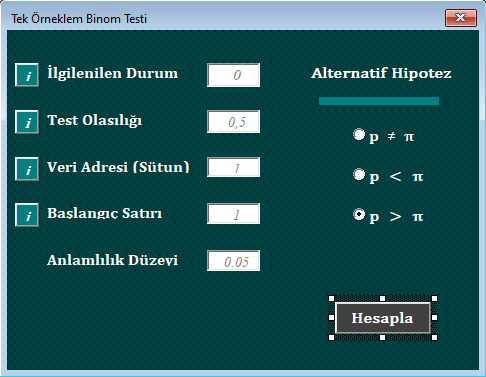
KiKareTesti = calcValue

End Function

# Şekiller

###### Şekil 1

Tek Örneklem Binom Testi Form Arayüzü



Şekil 2 (Tek Örneklem Binom Testi Form Arayüzü)

###### Şekil 2

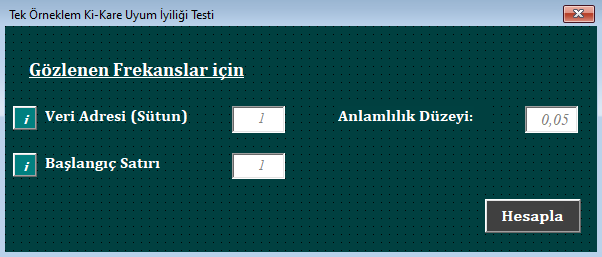
Tek Örneklem İşaret Testi Form Arayüzü



Şekil 2 (Tek Örneklem İşaret Testi Form Arayüzü)

###### Şekil 3

Tek Örneklem Ki-Kare Uyum İyiliği Testi Form Arayüzü



Şekil 3 (Tek Örneklem Ki-Kare Uyum İyiliği Testi Form Arayüzü)

# Kaynakça

Kişisel ders notları;

Parametrik Olmayan İstatistiksel Teknikler

Görsel Programlama

Hesap Tabloları ve Veri Analizi

<https://docs.microsoft.com/en-us/office/vba/api/overview/>

<https://www.techonthenet.com/excel/index.php>

<https://www.excel-easy.com/>

<https://www.excelinefendisi.com/default.aspx>

<https://excel.officetuts.net/>

<https://www.automateexcel.com/>

<https://stackoverflow.com/>

<https://wellsr.com/>

<https://www.slideshare.net/yigitcanozmeral/parametrik-olmayan-nonparametric-hipotez-testleri>

<https://docplayer.biz.tr/2628557-Parametrik-olmayan-testler-isaret-testi-the-sign-test-mann-whiney-u-testi-wilcoxon-testi-kruskal-wallis-testi.html>

<https://endustri.eskisehir.edu.tr/nerginel/ENM317/icerik/5.%20PARAMETR%C4%B0K%20OLMAYAN%20TESTLER.pdf>