



Yeniden Örnekleme Yöntemleri: Kavram ve R Uygulamaları

Resampling Methods: Concept and R Applications

C. Deha DOĞAN¹

Özet

Parametrik testler evren dağılımına ilişkin bir takım varsayımların karşılanmasını gerektirir. Bu varsayımların karşılanmadığı durumlarda araştırmacılar parametrik olmayan istatistiksel yöntemler kullanılır. Geleneksel parametrik olmayan yöntemlerde sıra puanları ve sıra ortalamalarına dayalı işlemler gerçekleştirirken yeniden örnekleme yöntemleri bu sürece farklı bir bakış açısı getirmiştir. En yaygın kullanılan yeniden örnekleme yöntemlerinin başında randomizasyon testleri gelir. Randomizasyon testlerinin temel mantığı orijinal örneklemden hesaplanan test istatistiğinin rastgele olarak oluşturulan örneklemlerdeki test istatistiği ile karşılaştırılmasına dayalıdır. Bu anlamda kullanımı dünyada gitgide yaygınlaşan randomizasyon testlerine ilişkin yeterli sayıda kaynak olmaması, özellikle Türkiye’de kullanımının çok sınırlı olması bu makalenin yazımına temel oluşturmuştur. Ayrıca randomizasyon testlerinin son yıllarda gündemde önemli bir yer tutan R programla dili üzerinden örneklendirilerek açıklanması çalışmanın diğer bir önemli unsuru olarak düşünülmektedir. Bu çalışmada randomizasyon testlerinin R programla dili ile örneklendirilerek açıklanması amaçlanmıştır. Bu bağlamda randomizasyon testlerinin temel kavramları açıklanmış akabinde sosyal bilimler alanında yaygın kullanıldığı düşünülen bağımsız ve tekrarlı örneklemler için t testi, bağımsız gruplar için tek yönlü varyans analizi ve korelasyon analizi özelinde R kodlarından faydalanılarak örneklendirmelere gidilmiştir. Bu çalışma ile özellikle Türkiye’de randomizasyon testlerinin ve R programlama dilinin kullanımının yaygınlaştırılması beklenmektedir.

Anahtar Kelimeler: yeniden örnekleme yöntemleri, randomizasyon restleri, permütasyon testleri, r programlama dili, parametrik olmayan testler, bootstrap

Abstract

Parametric tests such as t-test, ANOVA, etc. requires some assumptions about the distribution of the scores in the universe. If those assumptions are not met it is a good idea to compute non-parametric tests instead of parametric tests. Traditional non-parametric tests such as Wilcoxon Sum of Ranks and Kruskal Wallis tests etc. focus on the sum of ranks and mean ranks to compare the group scores. On the other hand, resampling methods present a different point of view on this process. One of the mostly used resampling methods is the randomization test. The basic principles of randomization tests are comparing the original test statistic (t values, F values, r coefficient, etc.) to the test statistics derived from randomly generated samples. Although usage of randomization tests in the world is pervading day by day in Turkey it is very rarely used. This may be because of insufficient written source published in Turkey. Moreover, the R programming language has become very popular recently. So in this study, it is aimed to explain the computation process of randomization tests using R codes. In this study, at first, some basic concepts about randomization tests were presented. Then randomization tests were exemplified for independent samples t-test, repeated sample t-test, one-way analysis of variance (one way ANOVA) using R codes. It is hoped that this study guide and motivate researchers to use randomizations tests and r programming language in their research.

Keywords: resampling methods, resampling tests, randomization tests, r programming language, non-parametric tests, bootstrap,

1. Ankara Üniversitesi Eğitim Bilimleri Fakültesi, Ölçme ve Değerlendirme Anabilim Dalı, Ankara, Türkiye; <https://orcid.org/0000-0003-0683-1334>

Atf / Citation: Doğan, C. D. (2019). Yeniden örnekleme yöntemleri: Kavram ve R uygulamaları. *Kastamonu Education Journal*, 27(6), 2747-2766. doi:10.24106/kefdergi.3756

Extended Abstract

Introduction

Parametric tests such as t-test, ANOVA, etc. requires some assumptions about the distribution of the scores in the universe. If those assumptions are not met it is a good idea to compute non-parametric tests instead of parametric tests. Traditional non-parametric tests such as Wilcoxon Sum of Ranks, Wilcoxon Signed Rank, and Kruskal Wallis tests, etc. focus on the sum of ranks and mean ranks to compare the group scores. On the other hand, resampling methods present a different point of view on this process. One of the mostly used resampling methods is the randomization test. Randomization tests can be thought of as another way to analyze data which don't require some restrictive assumptions about the populations. The basic principles of randomization tests are comparing the original test statistic (t values, F values, r coefficient, etc.) to the test statistics derived from randomly generated samples. Computation of randomization tests require the steps given below:

Step 1: Computing the required test statistic based on the original sample: In this step t-test statistic (t value, F value) is computed from the original sample. It is called observed test statistic (observed t value or observed F value etc.

Step 2: Generating new samples: In this step using the resampling method new samples are generated. In other words, the original sample is shuffled in order to get a new randomized sample. Then this process is repeated as the times the researcher planned. This is called iteration or replication numbers. For randomization tests, it is recommended to have more than 1000 replications.

Step 3: Computing test statistics for resampled (new) samples: In this step required test statistic is computed for each randomized (resampled) sample. So at the end of this step, we will have a statistic for each randomized sample. Those tests statistics will have a distribution.

Step 4: Comparison of test statistics: In this step, the original tests statistics and randomized test statistics are compared. The result gives us a randomized p-value. The computed p-value indicates the number of randomized test statistics which is equal or more than the originally observed test statistics.

Although usage of randomization tests in the world is pervading day by day in Turkey it is very rarely used. This may be because of insufficient written source published in Turkey. Moreover, the R programming language has become very popularly recently. So in this study, it is aimed to explain the computation process of randomization tests using R codes. In this study, at first, some basic concepts about randomization tests were presented. Then randomization tests were exemplified for independent samples t-test, repeated sample t-test, one-way analysis of variance (one way ANOVA) using R codes. It is hoped that this study guide and motivate researchers to use randomizations tests and r programming language in their research.

Below computation of randomization tests for independent and repeated samples t-tests, correlation and One Way ANOVA is explained with R codes respectively. Users can use the functions below to compute randomization tests. Besides the codes, some explanations were made with the "#" sign.

Codes for independent samples t tests to compute randomization tests

```

rand.t.test0 <-function(group,score,rep=2000) {
  observed.t<- round(t.test(score~group)$statistic,2) # computing observed t value
  resample <- function(){# function to resapmle data and computed randomized t value
    t.test(score ~ sample(group))$statistic
  }
  rept <- replicate(rep, resample()) # repeat resample data 2000 times as default
  abs_obst<-abs(observed.t) # take the absolute of t value
  repeat_abs<-abs(rept) # take the absolute of repeated resample t values
  p.value <- length(repeat_abs[repeat_abs >= abs_obst]) / rep # compute p value

  hist(rept,breaks = 50,plot = TRUE) # draw graphic of distribution of resample t va-
lues

  abline(v=observed.t,col="red",lwd=2)
  legend("topright", legend = c("obs.t",paste(round(abs_obst,2))),bty="n",cex=1.2)
  res<-data.frame(observed_t=observed.t,randomization.p.value=p.value)# create data
frame
  library(knitr)
# write possible interpretation of results
  if(p.value<0.05) {cat( paste("After", rep,
  " iterations mean differences between two groups is STATISTICALLY SIGNIFICANT
(p<0.05)") )} else
  {cat( paste("After", rep,
  " iterations mean differences between two groups is NOT STATISTICALLY SIGNIFICANT
(p>0.05)") )}
  res<-kable(res) #create a table with kable function in knitr package
  return(res)}

```

Codes for repeated samples t tests to compute randomization tests

```

rand.t.test01<-function(pre,post,rep=1000) {
  t.test.obs <- t.test(pre, post, paired = TRUE, conf.level = .95)
  t.obs <- t.test.obs$statistic # compute the observed t value
  t.obs<-abs(t.obs) # take the absolute of t value
  n<-length(post)
  t.random<-numeric (rep) # create an empty vector
  differ <- post-pre
  for (i in 1:rep){# For loop to resapmle data "i" times and computed randomized t values
    sign <- sample(c(1,-1),size = n, replace = TRUE)
    differ <- differ*sign
    t.random[i] <- t.test(differ)$statistic
  }
  up <- length(t.random[t.random >= t.obs]) # compute p value
  down<- length(t.random[t.random <= -t.obs])
  prob <- (up + down)/rep
  # draw graphic of distribution of resample t values
  hist(t.random,breaks = 50,plot = TRUE)
  abline(v=t.obs,col="red",lwd=2)
  legend("right", legend = c("t.obs",paste(round(t.obs,2))),bty="n",cex=1.2)
  res<-data.frame(observed_t=t.obs,randomized_p_value=prob) # create data frame
  library(knitr)
  # write possible interpretation of results
  if(prob<0.05) {cat( paste("After", rep,
"mean difference between pre and post test is STATISTICALLY SIGNIFICANT (p<0.05)"))}
  else
  {cat( paste("After", rep,
"mean difference between pre and post test is NOT STATISTICALLY SIGNIFICANT (p>0.05)"))}
  res<-kable(res) #create a table with kable function in knitr package
  return(res)
}

```

Codes for pearson correlation to compute randomization tests

```

rand.cor2<-function(x,y,rep=2000) {
  r.obs <- cor(x,y) # compute observed correlation coefficient
  r.random <- numeric(rep) # create an empty vector
  for (i in 1:rep) { #for loops to generate "i" times new samples
    Y <- y
    X <- sample(x, length(x), replace = FALSE)
    r.random[i] <- cor(X,Y)
  }
  prob <- length(r.random[r.random >= r.obs])/rep # compute p value
  draw graphic of distribution of resample r coefficients
  hist(r.random, breaks = 50,
    xlab = "r coefficients for randomized samples")
  r.obs <- round(r.obs, digits = 2)
  abline(v=r.obs,col="red",lwd=2)
  legend("right", legend = c("r",paste(round(r.obs,2))),bty="n",cex=1.2)

  # write possible interpretation of results
  if(prob<0.05) {cat( paste("After", rep,
"correlation between two variableis STATISTICALLY SIGNIFICANT (p<0.05)"))}
  else
  {cat( paste("After", rep,
"correlation between two variable is NOT STATISTICALLY SIGNIFICANT (p>0.05)"))}
  library(knitr)
  #create a table with kable function in knitr package
  return(kable(data.frame(randomized_p_value=prob,observed_correlation=r.obs) ))
}

```

Codes for one way ANOVA to compute randomization tests

```

rand.anova2<-function(group,score,rep=2000) {

  model<-aov(score~group)
  obs.F<-anova(model)[[4]][1] # compute observed F value
  empty.F <- numeric(rep) # create an empty vector
  # For loop to resapmle data "i" times and computed randomized F values
  for (i in 1:rep) {
    shuffle <- sample(score)
    newF <- anova(lm(shuffle~group))
    empty.F[i] <- newF$"F value"[1]
  }
  up <- length(empty.F[empty.F >= obs.F]) # compute p value
  down<- length(empty.F[empty.F <= -obs.F])
  prob <- (up + down)/rep

  # draw graphic of distribution of resample F values
  hist(empty.F,breaks = 50,plot = TRUE)
  abline(v=obs.F,col="red",lwd=2)
  legend("top",legend= c("Observed F value",paste(round(obs.F,2))),bty="n",cex=1.2)
  library(knitr)
  res<-data.frame(randomized_p_value=prob,observed_F.value=obs.F) # create data frame
  res2<-kable(res) create a table with kable function in knitr package
  # write possible interpretation of results
  if(prob<0.05) {cat( paste("After", rep,

  " replications mean difference between groups is STATISTICALLY SIGNIFICANT
(p<0.05)"))}
  else
  {cat( paste("After", rep,
  "replications mean difference between groups is NOT STATISTICALLY SIGNIFICANT
(p>0.05)"))}

  return(res2)
}

```

Result

The codes and functions given above will guide the researcher to compute randomization tests. Those codes and functions are written mostly for pedagogical purpose. The results of those functions weren't compared to the results of some other software. Moreover, the findings of those functions weren't tested for different conditions. So it is recommended readers to use those codes and function for the pedagogical purpose to understand the basic logic of randomization tests. It is hoped that this study guide and motivate the researcher to compute randomization tests using the R programming language.

1. Giriş

Yaygın olarak kullanılan pek çok istatistiksel yöntem bir örneklemden yola çıkarak puanların evrendeki dağılımına ilişkin kestirimler yaparken, dağılımın şekline ilişkin varsayımlarda bulunur. Örneğin t testi evren varyansını tahmin etmek için örneklem varyansını kullanır ancak bunu yaparken örneklemin seçildiği evrenin normal dağılım gösterdiği gibi bir varsayıma sahiptir. Parametreler veya onlara ilişkin tahminlere yönelik bir takım varsayımlara sahip testler parametrik testler olarak adlandırılır (Howel,2007). Başka bir ifade ile parametrik testler örneklemin seçildiği evren dağılımına ait parametreler ile ilgili bir takım varsayımlara sahiptir. İlgili varsayımlar sağlandığı takdirde oldukça güçlü olan parametrik testler aksi durumda yanlış sonuçlar üretebilmektedir. Özellikle sosyal bilimler ve eğitim bilimleri alanlarında bu varsayımların karşılanmadığı durumlara sıklıkla karşılanmaktadır. Bu durumun neticesi olarak evren dağılımına ilişkin varsayımları gerektirmeyen istatistiksel yöntemler geliştirilmiştir.

Evren dağılımına ilişkin varsayımları gerektirmeyen bu testler "parametrik olmayan testler" olarak adlandırılır. Parametrik olmayan testler, ilgili varsayımların karşılanmadığı durumlarda parametrik testlere kıyasla daha güvenilir sonuçlar üretir (Bradley 1968). Ancak her parametrik testin parametrik olmayan bir alternatifi bulunamamakta veya mevcut yazı-

lımlar vasıtası ile pratik bir şekilde hesaplanamamaktadır. Yaygın olarak bilinen parametrik olmayan testler (Wilcoxon Sıra Toplamları Testi, Wilcoxon İşaretli Sıralar Testi, Kruskal-Wallis testi vb.) ham puanların sıralanması ve sıra puanları atanması temeli üzerine kurgulanmaktadır. Bu sayede uçdeğerlere ilişkin yaşanabilecek sorunlar minimize edilmektedir.

Bunların dışında, bilgisayar ve yazılım teknolojisinin gelişimi ile son yıllarda gündeme gelen diğer bir yaklaşım ise yeniden örnekleme yöntemleridir. Yeniden örnekleme yöntemleri ağırlıklı olarak parametrik olmayan bir yapıya sahip oldukları için parametrik olmayan test yaklaşımının bir türü olarak da görülebilmektedir. Parametrik testlerdeki gibi dağılıma ilişkin bir takım varsayımları gerektirmeyen bu testler bootstrap yöntemleri ve randomizasyon (permutasyon) testleri olmak üzere iki boyutta ele alınabilir (Howel, 2007).

Bootstrap yöntemi orijinal örneklem üzerinden yerine koyarak örnekleme yolu ile alt örneklem oluşturulmasına odaklıdır. Bu yolla orijinal örneklem kullanılarak binlerce yeni alt örneklem oluşturulabilir. Bootstrap yöntemi özellikle standart hatanın analitik yollarla hesaplanmadığı testlerde standart hata ve güven aralıklarının hesaplanmasında yaygın olarak kullanılır. Temel amaç evrenden seçilen bir örneklem üzerinden yerine koyarak örnekleme yöntemi ile yeni örneklem oluşturmaya dayanır. Elde edilen yeni örneklemle ilişkin istatistikler hesaplanarak standart hata ve güven aralığı kestirimi yapılabilir. Bu sayede geleneksel olarak standart hata ve güven aralığı tahmini yapılamayan pek çok istatistik için güven aralığının hesaplanmasına olanak tanır. Yerine koyarak örnekleme yöntemi kullanıldığı için orijinal örneklemde daha fazla gözlem sayısını içeren örneklem oluşturulabilir (Banjanovic, Osborne & Jason, 2016, Doğan, 2017).

Randomizasyon/ permutasyon testleri ise örneklem üzerinde yerine koyarak örnekleme yapmak yerine veriye ilişkin tüm olası permutasyonları veya verinin çok sayıda yeniden düzenlenmesini içerir. Başka bir ifade ile randomizasyon testleri yerine koymadan örnekleme yaklaşımını kullanarak rastgele oluşturulmuş yeni alt veri setleri oluşturur. Bu sayede hesaplanacak parametrik testin varsayımları karşılanmadığı veya geleneksel parametrik olmayan alternatiflerinin bulunmadığı durumlarda bile pek çok istatistiksel test hesaplanabilmektedir. Bu çalışmanın odak noktasını randomizasyon testleri oluşturmaktadır ve ileriki kısımlarda randomizasyon testlerine ilişkin kavramsal bilgiler ve hesaplanmasına yönelik örnekler sunulmuştur. Okuyucular, bu çalışmada detaylı olarak ele alınmayan bootstrap yöntemleri ile ilgili daha ayrıntılı bilgiler için Davison & Hinkley (1997); Chernick & Labudde, (2011); Banjanovic, Osborne & Jason, (2016); Doğan, (2017) kaynaklarını inceleyebilirler.

Parametrik testlerin güçlü varsayımları gerektirmesi, bazı parametrik testlerin parametrik olmayan alternatifinin bulunmaması veya pratik bir şekilde hesaplanamaması randomizasyon testlerinin önemini artırmaktadır. Randomizasyon testlerinin kullanımı dünyada gitgide yaygınlaşırken ülkemizde oldukça sınırlı bir kullanıma sahiptir. Önemli avantajlara sahip olmasına rağmen, bu testlerinin kullanıldığı akademik yayınların diğer parametrik ve parametrik olmayan yaklaşımların kullanıldığı yayınlara kıyasla oldukça az olması dikkat çekici bir durumdur. Bu durumun gerekçesi olarak bu testleri kavramsal ve uygulama boyutunda açıklayan Türkçe yayınların az olması gösterilebilir. Bu bağlamda randomizasyon testlerini hem kavramsal hem de uygulama boyutunda ele alan bu çalışmanın uygulayıcılara yol gösterici olması ve yeniden örnekleme yöntemlerinin kullanımını yaygınlaştırması beklenmektedir. Çalışmanın uygulama boyutu sosyal bilimler ve eğitim bilimleri alanlarında sıklıkla kullanılan bağımsız ve tekrarlı ölçümler için t testleri, bağımsız ve tekrarlı gruplar için tek yönlü varyans analizi ve korelasyon analizleri üzerinden örneklendirilecektir. Bu konudaki Türkçe kaynakların az olması nedeni ile çalışma özellikle Türkçe olarak kaleme alınmıştır. Çalışmada istatistiksel detaylardan ve R programlama diline ilişkin ayrıntılardan mümkün olduğunca kaçınılmış ve örnekler üzerinden uygulamaya dönük açıklamalara yer verilmiştir.

İstatistiksel veri analizi ve programlama sürecinde kullanılan R programlama dili sahip olduğu avantajlar sayesinde son yıllarda büyük popülerite kazanmıştır. Parasız temin edilmesi, açık kaynaklı olması, kullanıcılara kendi fonksiyonlarını yazmalarına olanak tanınması, dinamik ve sürekli gelişen bir yapıya sahip olması bu avantajların başını çekmektedir (Beaujean 2013). Yeniden örnekleme yöntemlerinin gerçekleştirilmesi sürecine de olanak tanıyan fonksiyonların yazılabildiği R programlama dili basit yapısı sayesinde istatistiksel programlama sürecine ilişkin bilgi ve deneyimi az olan araştırmacılar için oldukça uygundur. Bu çalışmada randomizasyon testlerinin uygulama süreci R programlama dili ile yazılan kodlar ve fonksiyonlar üzerinden örneklendirilmiştir.

Bu bağlamda bu çalışmanın amacı yeniden örnekleme yaklaşımlarından randomizasyon testlerini kavramsal ve uygulama boyutunda R programlama dili ile örnekler üzerinden açıklamaktır.

Randomizasyon/ Permutasyon Testleri

Randomizasyon testlerinde rastgele (seçkisiz) atama önemli bir işlemdir. Aslında geleneksel parametrik testler için de rastgelelik önemli bir kavramdır. Geleneksel parametrik testlerde evren parametrelerine ilişkin doğru kestirim yapabilmek için örneklemin rastgele (seçkisiz) seçilmesi gerekmektedir. Randomizasyon testlerinde ise veri setindeki değer-

lerin rastgele olarak karıştırılması veya rastgele bir şekilde yeni gruplara atanması söz konusudur. Rastgele atama yapılarak pek çok yeni örneklem oluşturulabilir. Randomizasyon testlerinin temel mantığı orijinal örneklemden hesaplanan test istatistiğinin rastgele olarak oluşturulan örneklemdeki test istatistiği ile karşılaştırılmasına dayalıdır.

Bu durumu bir örnek üzerinden açıklamak süreci daha somutlaştıracaktır. Bir araştırmacının deneysel bir araştırma gerçekleştirdiğini, deney ve kontrol grubunda yer alan öğrencilerin bilişötesi düşünme becerilerini karşılaştırdığını varsayalım. Bu durumda araştırmacı deney ve kontrol gruplarında yer alan öğrencilerin bilişötesi düşünme becerileri puan ortalamalarını bağımsız örneklem için t testi hesaplayarak karşılaştırabilir. Ancak puanların dağılımının normallik varsayımını karşılamadığını ve araştırmacının bu nedenle parametrik bir test olan t testini hesaplayamadığını düşünelim. Bu durumda randomizasyon testini kullanmak isteyen araştırmacının aşağıdaki adımları gerçekleştirmesi gerekecektir.

Adım 1 - Orijinal örneklem üzerinden test istatistiğinin hesaplanması: Araştırmacının öncelikli olarak orijinal örnekleme dikkate alarak test istatistiğini hesaplaması gerekmektedir. Örneğimiz için araştırmacının orijinal örneklem üzerinden iki grup ortalamasını karşılaştırması ve t değerini hesaplaması gerekmektedir. Asıl hesaplanmak istenen test istatistiğinin (t değeri) yanı sıra deney ve kontrol grubunun ortalama puanları arasındaki fark da test istatistiği olarak alınabilmektedir.

Adım2 - Yeni örneklem oluşturulması: Bu aşamada araştırmacı orijinal örneklem üzerinden deneklerin (öğrencilerin) deney veya kontrol grubunda olma durumlarını dikkate almadan rastgele (seçkisiz) bir şekilde iki örneklem seçer ve bunların birisini deney diğerini kontrol grubuna atayarak yeni bir örneklem oluşturur. Başka bir ifade ile deneklerin deney ve kontrol grubunda olma durumlarını rastgele bir şekilde karıştırılır. Daha sonra araştırmacı bu işlemi belirlediği bir sayı kadar tekrar ettirir. Bu sayı "tekrar sayısı" veya "replikasyon sayısı" olarak adlandırılır. Bilgisayar yazılımlarının kullanılması ile bir işlem birçok defa ettirilebilir. Bu sayının belirlenmesinde hesaplanacak olan test istatistiğinin karmaşıklığı ve örneklemin büyüklüğü de dikkate alınabilir. Örneğin farklı koşulların (örneklem büyüklüğü, madde sayısı, dağılımın yapısı vb.) simüle edildiği ve her koşul için pek çok istatistiğin hesaplandığı bazı durumlarda 40 replikasyon yeterli olabilirken daha basit durumlar için replikasyon sayısının çok daha fazla olması beklenmektedir. Farklı istatistikler ve araştırma desenleri için değişik replikasyon sayıları önerilebilmektedir. Howel, (2007) randomizasyon testlerinde güvenilir sonuçların alınabilmesi için 1000'in üzerinde replikasyon yapılmasını önermektedir. Bu işlem sonucunda replikasyon sayısı kadar yeni örneklem elde edilmiş olunur.

Adım 3 - Yeni örneklem için test istatistiklerinin hesaplanması: Bu aşamada araştırmacının replikasyonlar ile oluşturduğu her örneklem için ilgili test istatistiğini hesaplaması gerekmektedir. Bu işlem neticesinde her örneklem için hesaplanan test istatistiklerinin bir dağılımı oluşacaktır.

Adım 4- Test istatistiklerinin karşılaştırılması: Bu son aşamada orijinal örneklemden elde edilen test istatistikleri ile replikasyonlar sonucu oluşturulan yeni örneklemde hesaplanan test istatistikleri karşılaştırılır. Orijinal test istatistiğinin, yeni örneklemde elde edilen test istatistiklerinin %5'inden küçük olması (ya da %95'inden büyük olması) sonucun 0.05 alfa düzeyinde manidar olduğu şeklinde yorumlanır. Başka bir ifade ile orijinal test istatistiğinin yeni örneklemde test istatistiklerinin yer aldığı dağılımda üst %5'lik dilimde bulunması (tek yönlü bir hipotez için) sonucun 0.05 alfa düzeyinde manidar olduğunu gösterir. Bu değer araştırma başında belirlenen alfa değerine veya hipotezin tek yönlü ya da iki yönlü kurulmuş olmasına göre de farklılaşabilmektedir.

Yukarıda belirtilen dört aşama bütün randomizasyon testlerinin temel adımları olarak düşünülebilir. Ancak bağımsız ve/veya bağımlı değişken sayısı 1'den fazla olan istatistikler için (örneğin faktoriyel ANOVA vb.) bu süreç biraz daha karmaşık bir yapıya dönüşmektedir.

Randomizasyon testi ve permutasyon testi kavramları alanyazında sıklıkla birbirlerinin yerine kullanılmakla beraber bu iki kavramın farklı ele alınması gerektiği de düşünülmektedir. Randomizasyon testlerinde araştırmacı yeniden örnekleme işlemini belirlediği bir sayı kadar tekrar ettirmektedir. Temel olarak permutasyon testlerinde ise verilerin rastgele karıştırılması sürecinde olası tüm permutasyonları içerecek sayıda yeniden örneklem üretilmesi gerekmektedir. Bu durumun örneklem sayıları ve grup sayılarının çok az olmadığı koşullarda gerçekleştirilmesi mümkün olamamaktadır. Örneğin gözlem sayısı 35 olan bir örneklem için olası tüm permutasyonlar 35! yani 1.03 x 10⁴⁰ kadardır ve bilgisayar yazılımları ile bile hesaplanması haftalar alabilecektir (Albert & Rizzo, 2012). Bu durum bir korelasyon veya t testi hesaplamak için pek pratik gözükmemektedir.

Onghena (2018), bu iki kavramın hem tarihsel hem de kavramsal açıdan farklılaştığını düşünmektedir. Yazara göre kavramsal bir bakış açısı ile randomizasyon testleri rastgele (seçkisiz) atamaya dayalı iken permutasyon testlerinde rastgele örnekleme söz konusudur. Ancak alanyazında bu iki kavramın kullanımına ilişkin tam bir uzlaşmanın olduğu

da söylenemez. Bazen aynı anlama gelen kavramlar uygulamada farklı süreçler içerirken, bazen de farklı anlama gelen kavramlar benzer uygulama süreçlerini içerebilmektedir. Randomizasyon ve permutasyon testleri kavramlarının kullanımında yaşanan karışıklığın nedeni olarak bu gösterilebilir. Örneğin Gibbons (1986)'a göre permutasyon testleri randomizasyon testlerinin bir türü iken Edgington (1986) ise randomizasyon testlerinin radtegele atama içeren permutasyon testi olduğunu düşünmektedir. Bu çalışmada, belirtilen kavram kargaşasına ve bu kargaşanın çözümüne ilişkin detaylı açıklamalara değinilmeyecektir. Randomizasyon ve permutasyon testlerinin tarihsel gelişimi, kavramsal farklılıkları, Monte Carlo testleri ile ilişkisi gibi konularda detaylı bilgi edinmek isteyen okuyucuların (Onghena, 2018) kaynağına ulaşmaları önerilir. Ayrıca ilgili okuyuculara randomizasyon testlerinde p değeri hesaplanması ve yorumlanması sürecinde dikkat edilmesi gereken boyutlara odaklanan (Onghena & May, 1995) kaynağını incelenmeleri önerilir.

Makalenin ileriki kısımlarında çalışmanın amacında belirtilen istatistiksel yöntemleri R kod ve fonksiyonları ile hesaplama süreci açıklanmıştır. Okuyucuların R programlama diline ilişkin detaylı bilgiye sahip olmalarına gerek yoktur. Okuyucuların makalede sunulan kod ve fonksiyonları kopyalayarak kendi bilgisayarlarında kullanmaları yeterli olacaktır. Ayrıca ele alınan istatistiksel yöntemlerin ayrıntılı açıklamaları da bu çalışmanın kapsamı dışındadır. Bu konularda ayrıntılı bilgi edinmek isteyen araştırmacıların (Field, 2018) kaynağına ulaşmaları önerilir.

Bağımsız Örneklemeler için t Testi ve Randomizasyon Testleri

İki ortalama puanın karşılaştırılması sürecinde Sosyal Bilimler ve Eğitim Bilimleri alanlarında sıklıkla kullanılan bağımsız gruplar için t testi normallik varsayımı ve varyansların homojenliği gibi önemli varsayımları içeren parametrik bir testtir. Bağımsız örneklemeler için t testini deneysel bir araştırma üzerinden örneklendirilmiştir.

Örnek 1: Araştırmada Deney ve Kontrol olmak üzere iki grup mevcuttur. Bu iki grubun deneysel işlem sonrasında işbirlikli problem çözme ölçeği son test puanlarının karşılaştırılması ile ilgilenilmektedir. Deney ve kontrol grubunda yer alan 30'ar kişinin son test puanları aşağıda sunulmuştur.

```
> deney
60 66 46 68 43 48 74 54 84 63 49 46 55 63 76 65 70 51 72 63 59 64 46 56 63 32 56 60 60 28

> kontrol
66 53 45 39 50 61 49 44 57 53 38 46 78 45 27 43 42 58 38 57 45 54 37 41 37 38 51 36 46 60
```

Okuyucuların yukarıdaki veri setini kendi bilgisayarlarına aktarmak için aşağıdaki kodları sırasıyla yazıp akabinde ilgili değerleri kopyalayıp R Console ekranına yapıştırmaları yeterli olacaktır.

```
deney <- scan()
kontrol <-scan()
```

Akabinde analizlerin gerçekleştirilmesi için belirtilen vektörler aşağıda verilen kodlar ile bir veri çerçevesine dönüştürülerek "örnek1" ismi verilmiştir.

```
örnek1<- as.data.frame(cbind(deney,kontrol)) # vektörlerin veri çerçevesine dönüştürülmesi
örnek1<-stack(örnek1) # veri çerçevesinin iki sütun içerecek şekilde düzenlenmesi
colnames(örnek1)<-c("sontest","grup") # sütun isimlerinin verilmesi
head(örnek1) # veri çerçevesinin ilk 6 değeri
  puan grup
1 60 deney
2 66 deney
3 46 deney
4 68 deney
5 43 deney
6 48 deney
```

Randomizasyon testine dayalı olarak iki grubun ortalama puanlarını karşılaştırmak için çalışmanın başlarında belirtilen adımları gerçekleştirmek gerekmektedir. Öncelikli olarak mevcut veri setini kullanarak bir test istatistiği elde edilir. Mevcut örnek için t değerinin hesaplanması gerekmektedir. Bunun için kullanıcılar kendi kod ve fonksiyonlarını yazabilmekle beraber R ortamında yer alan "t.test" fonksiyonunu kullanmak pratiklik sağlayacaktır. Gerçekleştirilen hesaplama neticesinde analiz çıktısı içerisinde t değeri alınmıştır. Bu değere "gözlünen t değeri" nin kısaltması olan "göz.t" ismi verilmiştir.

t testinin hesaplanması ve t değerinin analiz çıktısı içerisinde çekilmesi

```
göz.t<-t.test(örnek1$sonetest ~ örnek1$grup)$statistic
göz.t
t
3.42396      # gözlenen t değeri
```

Gözlenen t değerini hesapladıktan sonra bir sonraki adım olarak deneklerin deney ve kontrol grubunda olma durumlarını rastgele bir şekilde karıştırarak yeni örneklem oluşturacak ve o örneklem için t testini hesaplayıp t değerini bize sunan kısa bir fonksiyon yazılmış ve bu fonksiyona “yeniden.örnekle” ismi verilmiştir.

```
yeniden.örnekle <- function(){
t.test(örnek1$sonetest ~ sample(örnek1$grup))$statistic
}
yeniden.örnekle() # fonksiyonun çalıştırılması
```

Yukarıdaki fonksiyon her çalıştırıldığında mevcut veri seti için deney ve kontrol grupları rastgele karıştırılarak yeni bir veri seti oluşturulacak, o veri seti üzerinden t testi hesaplanacak ve ihtiyaç duyulan test istatistiği olan t değeri çıktı olarak sunulacaktır. Üçüncü adımda ise “yeniden.örnekle” isimli fonksiyon ile yapılan işlemi bir çok defa tekrar ettirecek kodların yazılması gerekmektedir. Bunun için “for” döngüleri içeren kodlar yazılabilmekle beraber mevcut örnek için “replicate” fonksiyonundan faydalanmak daha pratik olacaktır. Yapılan bu işlem “tekrar” isimli bir obje içerisine kaydedilmiş ve işlemi 5000 defa tekrar ettirilmiştir. Başka bir ifade ile deney ve kontrol gruplarını rastgele karıştırarak 5000 yeni örneklem oluşturulmuştur.

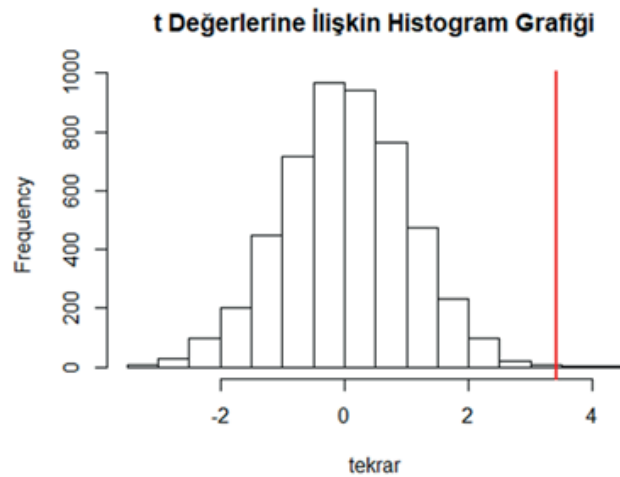
```
tekrar<- replicate(5000, yeniden.örnekle()) # süreci 5000 defa tekrar ettirme
```

Yukarıdaki komut ile 5000 yeni örneklem oluşturulacak ve bu yeni örneklemeler üzerinden hesaplanan t değerleri “tekrar” isimli bir obje içerisinde kaydedilecektir. Dördüncü ve son aşamada ise orijinal veriden elde edilen gözlenen t değeri ile 5000 replikasyon (tekrar) sonucunda elde edilen t değerlerinin karşılaştırılması gereklidir. Bunun için aşağıdaki kodlardan faydalanılabilir.

```
p.değeri<- mean(tekrar > göz.t)*2
p.değeri
[1] 0.0016
```

Yukarıda verilen kod kısaca randomizasyon ile üretilen 5000 örneklem için hesaplanan t değerlerinin yüzde kaçının orijinal veri üzerinde hesaplanan t değerinden yüksek olduğunu verir. Elde edilen değer yaygın olarak kullanılan 0.05 ve 0.01 gibi alfa düzeylerinden düşük olması sonucun istatistiksel olarak manidar olduğu anlamına gelmektedir. Diğer bir ifade ile elde edilen sonuç (0.0016) deney ve kontrol gruplarının ortalama puanları arasında gözlemlenen farklılıkların şansa ortaya çıkma olasılığının çok düşük olduğu şeklinde yorumlanabilir. Aşağıdaki kısa kod ile de bu belirtilen durumun histogram grafiğini çizebilir ve orijinal veriden elde edilen gözlenen t değeri grafik üzerinde gösterilebilir (Şekil 1).

```
hist(tekrar, main="t Değerlerine İlişkin Histogram Grafiği")
abline(v=göz.t,col="red",lwd=2)
```



Şekil 1. Randomizasyon testi ile elde edilen t değerlerinin dağılımı

Grafikte yer alan kırmızı çizgi orijinal veriden elde edilen t değerini göstermektedir. Grafikte de açıkça görülmektedir ki 5000 tekrar sonucunda elde edilen t değerlerinin çok az bir kısmı orijinal veriden elde edilen t değerinden büyüktür ve bu da gözlenen farkın şansla ortaya çıkma olasılığının düşükliğini göstermektedir.

Yukarıda belirtilen işlemleri daha kolay ve hızlı bir şekilde gerçekleştirmek için aşağıda verilen fonksiyon kullanılabilir. Bu fonksiyonu bir kez tanımladıktan sonra daha pratik bir şekilde bağımsız örneklem için t testi için randomizasyon testi hesaplanabilir. Yalnızca bu fonksiyonu çalıştırmadan önce “knitr” paketini indirmeniz fonksiyon çıktısının bir çizelge şeklinde sunulması açısından gereklidir.

```
rand.t.test1 <-function(grup,puan,rep=2000) {
  gözlenen.t<- round(t.test(puan~grup)$statistic,2)
  yeniden.örnekle <- function(){
    t.test(puan ~ sample(grup))$statistic
  }
  tekrar <- replicate(rep, yeniden.örnekle())
  abs.goz<-abs(gözlenen.t)
  tekrar.abs<-abs(tekarar)
  p.deger<- length(tekarar.abs[tekarar.abs >= abs.goz]) / rep
  hist(tekarar,breaks = 50,plot = TRUE)
  abline(v=gözlenen.t,col="red",lwd=2)
  legend("right", legend = c("göz.t",paste(round(gözlenen.t,2))),bty="n",cex=1.2)
  res<-data.frame(gözlenen.t,randomizasyon_testi_p.degeri=p.deger)
  library(knitr)
  if(p.deger<0.05) {cat( paste("Gerçekleştirilen", rep,
    " tekrar sonrasında karşılaştırılan iki grup arasındaki farkın
    istatistiksel olarak MANİDAR OLDUĞU belirlenmiştir.(p<0.05)")) }
  else
  {cat(paste("Gerçekleştirilen",rep,
    " tekrar sonrasında karşılaştırılan iki grup arasındaki farkın
    istatistiksel olarak MANİDAR OLMADIĞI belirlenmiştir.(p>0.05)")) }
  res<-kable(res)
  return(res)
}
```

Yukarıda verilen “rand.t.test1” isimli fonksiyon girdi olarak üç argüman içermektedir. Bunlardan birincisi karşılaştırılacak gruplara ait değişken iken diğeri grupların karşılaştırılacağı sürekli puanları ifade etmektedir. Üçüncüsü ise yeniden üretilmek istenen örneklem sayısı, başka bir ifade ile tekrar sayısıdır. Bu sayı ön tanımlı (default) olarak 2000’e sabitlenmiştir. Bu argüman girilmezse fonksiyon 2000 replikasyon gerçekleştirecektir.

Yazılan bu fonksiyonda verilen örnekten farklı olarak, kullanıcıların grup ve sürekli puan değişkenlerinin ayrı sütunlarda yer aldığı bir veri çerçevesine (data frame) sahip olabileceği düşünülmüş ve fonksiyonun girdileri buna göre düzenlenmiştir. Bu fonksiyon kopyalanıp R Console’a aktarılıp çalıştırdıktan sonra aşağıdaki kod yazılarak 5000 tekrar içerecek şekilde bağımsız gruplar t testi için randomizasyon testi hesaplanabilir.

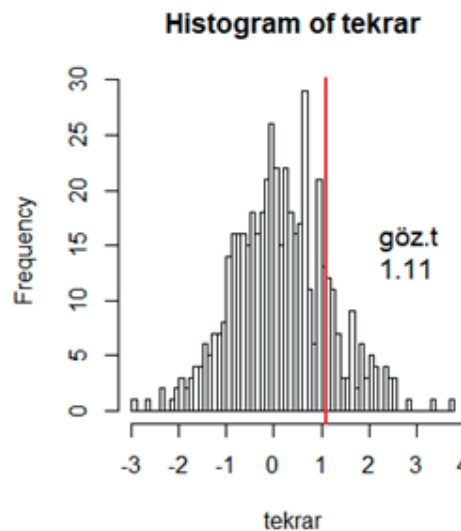
```
rand. t. test1 (grup değişkeninin ismi , sürekli değişkeninin ismi,5000)
```

“rand.t.test1” fonksiyonunun örnek bir çıktısı Şekil 2’de sunulmuştur.

Gerçekleştirilen 5000 tekrar sonrasında karşılaştırılan iki grup arasındaki farkın istatistiksel olarak MANİDAR OLMADIĞI belirlenmiştir.

($p > 0.05$)

	gözlenen.t	randomizasyon_testi_p.degeri
t	1.11	0.296



Şekil 2. “rand.t.test1” fonksiyonunun örnek bir çıktısı

Tekrarlı ölçümler için t Testi ve Randomizasyon Testleri

Tekrarlı ölçümler için t testi de aynı grup üzerinde farklı zamanlarda elde edilen puanların ortalamalarını karşılaştırmak için kullanılan parametrik bir testtir ve fark puanlarının evrende normal dağıldığı varsayımı gerektirir. Bağımsız örneklem için t testinde olduğu gibi burada da deneysel bir araştırma üzerinden örneklendirmeye gidilmiştir.

Örnek 2: Bu örnekte ise deney grubunda bulunan öğrencilerin “öntest” ve “sontest” işbirlikli problem çözme ölçeği puan ortalamaları arasındaki farkının anlamlılığı randomizasyon testleri ile sınınmıştır Deney grubunun ön test ve sontest puanları aşağıda verilmiştir.

```
> öntest
54 56 68 47 56 70 65 56 57 71 45 59 64 58 55 54 75 66 51 69 57 75 43 80 65 59 51 76 66 60

> sontest
67 45 56 53 49 66 88 65 73 73 49 74 65 58 65 63 77 63 59 67 82 57 69 51 69 54 49 76 72 56
```

Mevcut veriler üzerinden alıştırma yapmak isteyen okuyucular daha önce belirtildiği gibi “scan” fonksiyonunu kullanarak verileri R ortamına aktarabilirler. Daha sonra randomizasyon testlerinin adımlarına bağlı olarak öncelikle orijinal veri seti için tekrarlı ölçümler için t testi hesaplanmış ve bu test sonucunda elde edilen gözlenen t değeri çıktı içerisinde alınmıştır.

```
# tekrarlı ölçümler için t testinin hesaplanması

t.test.göz <- t.test(öntest, sontest, paired = TRUE, conf.level = .95)

t.göz <- abs(t.test.göz$statistic) # t değerinin çıktı içerisinden çekilmesi ve mutlak değerinin alınması
```

Yukarıdaki kodlar sonucunda orijinal veriden elde edilen gözlenen t değerini “t.göz” isimli bir obje içerisine kaydedilmiştir. İkinci adım olarak değerler rastgele (seçkisiz) bir şekilde karıştırılarak yeni örneklem oluşturulması ve akabinde yeni oluşturulan her bir örneklem için tekrarlı ölçümler için t testinin ve t değerinin hesaplanması gerekmektedir. Örnek 1’de benzer bir işlem için “yeniden örnekle” isimli bir fonksiyon yazılmış ve bu fonksiyon “replicate” fonksiyonu içerisinde kullanılmıştı. Örnek 2’de ise “replicate” fonksiyonu kullanılmadan “for” döngüleri kullanılarak benzer bir işlem gerçekleştirilmiştir.

```

tekrar.sayı<-5000 # yeniden üretilecek örneklem sayısı

n<-length(öntest)

t.rastgele<-numeric(tekrar.sayı) # yeni t değerlerinin kaydedileceği boş vektör

fark <- sontest-öntest

# tekrar işleminin gerçekleştirilmesi ve t değerlerinin hesaplanması

for (i in 1:tekrar.sayı) {
  işaretler <- sample(c(1,-1),size = n, replace = TRUE)
  fark <- fark*işaretler
  t.rastgele[i] <- t.test(fark)$statistic
}

```

Yukarıda verilen kodlarda öntest ve sontest puanlarının karıştırılma işlemi, her seferinde rastgele olarak denek sayısı kadar üretilen (-1, 1) değerlerini içeren bir vektör ile fark puanlarının çarpımı ile gerçekleştirilmiştir. Son aşama olarak orijinal veriden elde edilen gözlenen t değeri ile 5000 replikasyon (tekrar) sonucunda elde edilen t değerlerinin karşılaştırılması gereklidir. Bu amaçla aşağıdaki kodlardan faydalanılabilir.

```

üst <- length(t.rastgele[t.rastgele >= t.göz])
alt<- length(t.rastgele[t.rastgele <= -t.göz])
olasılık <- (üst + alt)/tekrar.sayı
> olasılık
[1] 0.2472

```

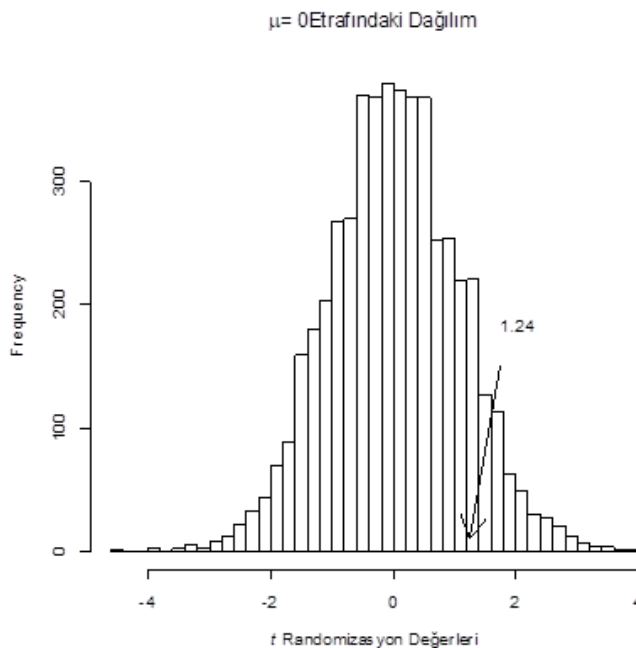
Elde edilen sonuç deney grubu öğrencilerinin öntest ve sontest puan ortalamaları arasındaki farkın istatistiksel olarak manidar olmadığını göstermektedir ($p > 0.05$). Aşağıdaki kod ile bu durumun grafiği örnek 1'de olduğu gibi çizilebilir ve grafikte gözlenen t değerinin yeri ok ile belirtilebilir (Şekil 3).

```

hist(t.rastgele, breaks = 50, main = expression(paste(mu, "=", 0, "Etrafındaki Dağılım ")),
  xlab = expression(paste(italic(t), " Randomizasyon Değerleri")))
t.göz <- round(t.göz, digits = 2)
legend(t.göz, 200, t.göz, bty = "n")
arrows(t.göz +.5,150,t.göz, 10)

```

Grafikte de açıkça görülmektedir ki 5000 tekrar sonucunda elde edilen t değerlerinin %5'inden büyük bir kısmı orijinal veriden elde edilen t değerinden büyüktür. Bu durum gözlenen farkın şansla ortaya çıkma olasılığının yüksek olduğu ve ortalama puanlar arasındaki farkın istatistiksel olarak manidar olmadığı şeklinde yorumlanır.



Şekil 3. Randomizasyon testi ile elde edilen t değerlerinin dağılımı

```

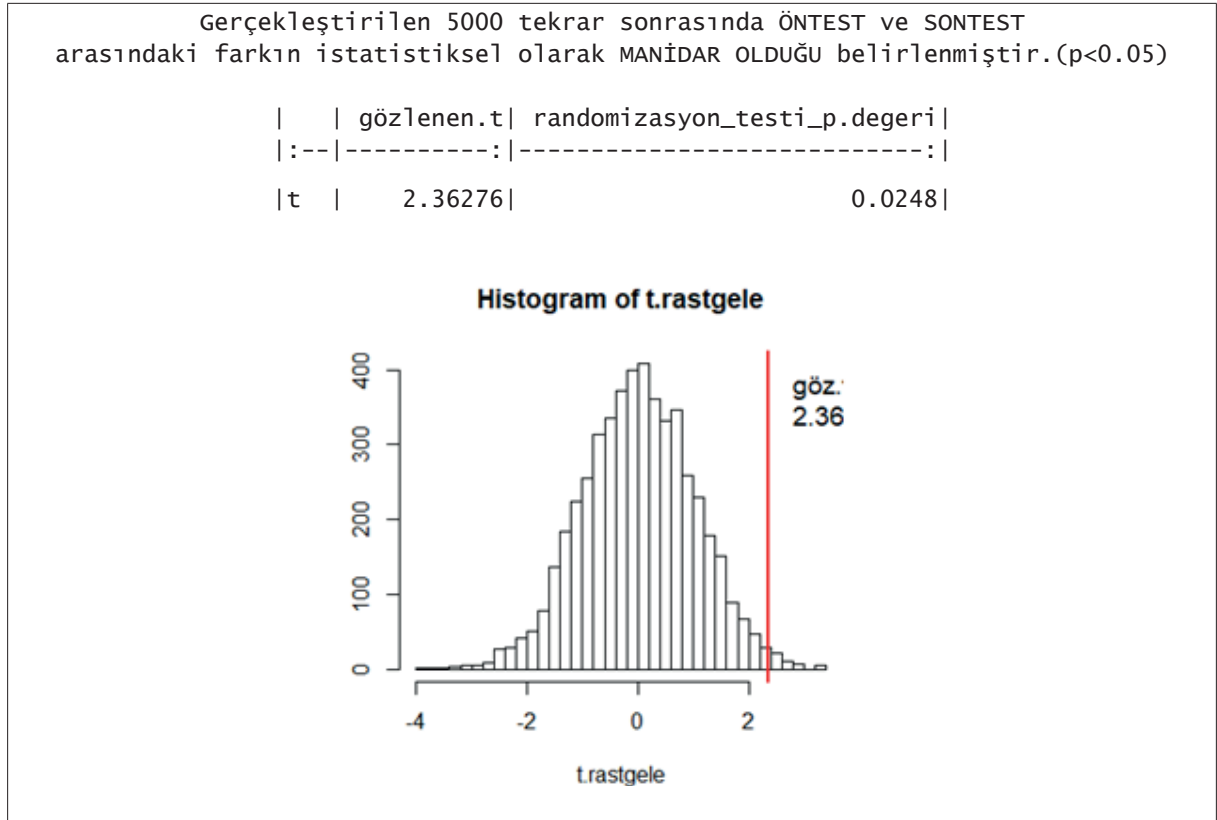
rand.t.test2<-function(önce,sonra,tekrar=1000) {
  t.test.göz <- t.test(önce, sonra, paired = TRUE, conf.level = .95)
  t.göz <- t.test.göz$statistic
  t.göz<-abs(t.göz)
  n<-length(sonra)
  t.rastgele<-numeric (tekrar)
  fark <- sonra-önce
  for (i in 1:tekrar) {
    işaretler <- sample(c(1,-1),size = n, replace = TRUE)
    fark <- fark*işaretler
    t.rastgele[i] <- t.test(fark)$statistic
  }
  üst <- length(t.rastgele[t.rastgele >= t.göz])
  alt<- length(t.rastgele[t.rastgele <= -t.göz])
  olasılık <- (üst + alt)/tekrar
  hist(t.rastgele,breaks = 50,plot = TRUE)
  abline(v=t.göz,col="red",lwd=2)
  legend("topright", legend = c("göz.t",paste(round(t.göz,2))),bty="n",cex=1.2)
  res<-data.frame(gözlenen.t=t.göz,randomizasyon_testi_p.degeri=olasılık)
  library(knitr)
  if(olasılık<0.05) {cat( paste("Gerçekleştirilen", tekrar,
    "tekrar sonrasında ÖNTEST ve SONTEST arasındaki farkın
    istatistiksel olarak MANİDAR OLDUĞU belirlenmiştir.(p<0.05)") ) }
  else
  {cat(paste("Gerçekleştirilen",tekrar,
    "tekrar sonrasında ÖNTEST ve SONTEST arasındaki farkın
    istatistiksel olarak MANİDAR OLMADIĞI belirlenmiştir.(p>0.05)") ) }
  res<-kable(res)
  return(res)}

```

Yukarıda tanımlanan "rand.t.test2" fonksiyonu girdi olarak üç argüman içermektedir. İlk iki argüman grupların birinci (önce) ve ikinci (sonra) ölçümlerini, son argüman ise tekrar sayısını içermektedir. Tekrar sayısı "rand.t.test1" fonksiyonunda olduğu gibi 1000 değerine sabitlenmiştir. Bu fonksiyon R ortamında tanımlandıktan sonra aşağıdaki kod yazılarak tekrarlı ölçümler t testi için randomizasyon testi hesaplanabilir. Yalnızca bu fonksiyonu çalıştırmadan önce "knitr" paketini indirmeniz fonksiyon çıktısının bir çizelge şeklinde sunulması açısından gereklidir.

```
rand.t.test2(öntest,sontest,5000)
```

"rand.t.test2" fonksiyonunun örnek bir çıktısı Şekil 4'te sunulmuştur.



Şekil 4. "rand.t.test2" fonksiyonunun örnek bir çıktısı

Pearson Korelasyon Katsayısı ve Randomizasyon Testleri

Pearson Korelasyon katsayısı iki sürekli değişken arasındaki doğrusal ilişkinin yönünün ve düzeyinin belirlenmesinde sıklıkla kullanılan parametrik bir testtir. Pearson korelasyon katsayısı ilişkileri incelenecek değişkenlerin evrende normal dağıldığı varsayımına sahiptir.

Örnek 3: Bu bölümde aşağıda sunulan iki değişken arasındaki ilişki ve bu ilişkinin anlamlılığı randomizasyon testleri ile hesaplanarak örneklendirilmiştir. Arasındaki ilişkinin hesaplanacağı sürekli değişkenlerden birincisi öğrencilerin okula yönelik tutumu (tutum), diğeri ise dönem sonu başarı puanlarıdır (başarı).

```
> tutum
 154 123 146 139 122 140 145 141 142 143 135 153 147 150 132 133 134 144 138 159 139 132 135
 130 151 156 136 122 137 140

> başarı
 84 80 74 69 75 66 66 69 81 78 100 67 77 76 25 63 74 78 69 79 74 64 73 59 77 90 66 88 45 78
```

Önceki örneklerde olduğu gibi yapılması gereken ilk aşama orijinal veriler üzerinden gözlenen korelasyon katsayısını hesaplamaktır. Bu katsayı "r.goz" isimli bir obje içerisinde kaydedilmiştir.

```
r.goz <- cor(tutum, başarı)
r.goz
[1] 0.21
```

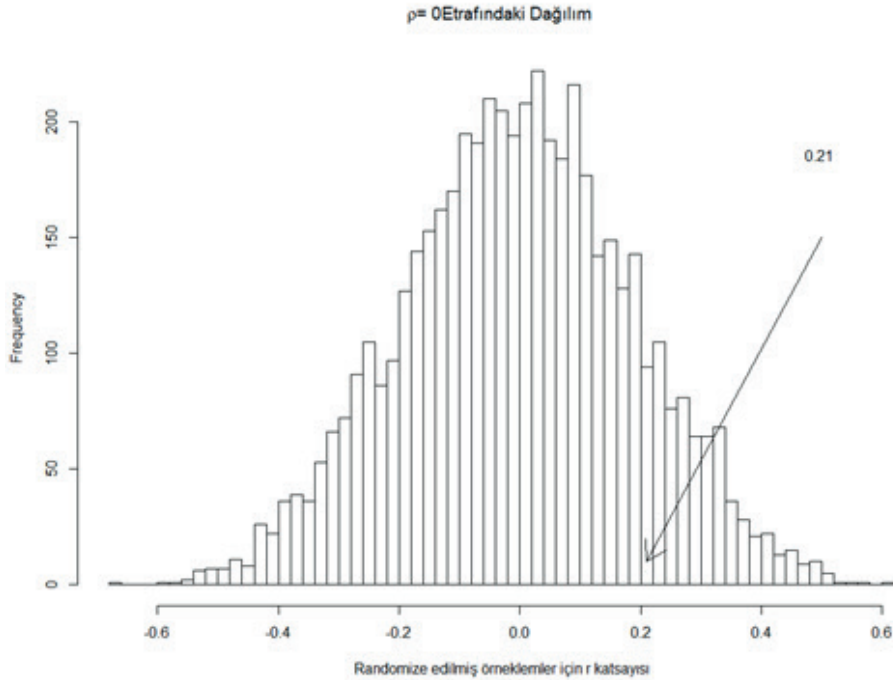
Gözlenen korelasyon katsayısına dayalı olarak iki değişken arasında pozitif ancak zayıf bir ilişki olduğu belirtilebilir. Şimdi bu ilişkinin anlamlılığının randomizasyon testi ile sınanması için aşağıdaki kodlar kullanılabilir.

```
nreps <- 5000
r.random <- numeric(nreps)
for (i in 1:nreps) {
  Y <- başarı
  X <- sample(tutum, 28, replace = FALSE)
  r.random[i] <- cor(X,Y)
}
ols <- length(r.random[r.random >= r.göz])/nreps
> ols
[1] 0.1312
```

Bu kodlarda öncelikli olarak tekrar sayısı 5000 olarak belirlenmiştir. Akabinde "r.random" isimli ve 5000 gözlem içeren boş bir vektör oluşturulmuştur. Sonrasında yer alan "for" döngüsü ile "tutum" değişkeninde yer alan 28 gözlem rastgele bir şekilde karıştırılarak 5000 yeni örneklem elde edilir. Bu süreçte arasındaki ilişki incelenen iki değişkenden birisi sabit tutulup diğeri üzerinden yeni örneklem oluşturulur. Bu örnekte "başarı" değişkeni sabit tutulmuş ve "tutum" değişkeni üzerinden 5000 yeni örneklem oluşturulmuştur. Daha sonra tutum değişkenine ilişkin yeni oluşturulan 5000 yeni veri seti ile başarı değişkeni arasındaki korelasyon katsayıları hesaplanmış ve bu hesaplanan katsayılar "r.random" isimli boş vektör içerisinde kaydedilmiştir.

Son aşama olarak "r.random" vektörü içerisinde yer alan korelasyon katsayılarının ne kadarının gözlenen verilerden elde ettiğimiz korelasyon katsayısına eşit veya daha fazla olduğunu belirlemiş ve bu değer "ols" isimli obje içerisine kaydedilmiştir. Elde edilen sonuç iki değişken arasında ilişkinin anlamı olmadığı, randomize edilen değişkenler ile elde edilen korelasyon katsayılarının yaklaşık %13'ünün gözlenen korelasyondan yüksek olduğu şeklinde yorumlanabilir. Diğer bir ifade ile gözlenen korelasyon katsayısının şansla ortaya çıkma olasılığı %5'ten fazladır ve bu bulgu elde edilen sonucun 0.05 düzeyinde manidar olmadığı anlamına gelir. Elde edilen sonucu grafik ile görselleştirmek için aşağıdaki kodlardan faydalanılabilir (Şekil 5).

```
hist(r.random, breaks = 50, main = expression(paste(,rho, "= 0", "Etrafındaki Dağılım ")),
     xlab = "Randomize edilmiş örneklem için r katsayısı")
r.göz <- round(r.göz, digits = 2)
legend(.40, 200, r.göz, bty = "n")
arrows(.5, 150, r.göz, 10)
```



Şekil 5. Randomizasyon testi ile elde edilen korelasyon katsayılarının dağılımı

Grafikte de görüldüğü gibi 5000 tekrar neticesinde elde edilen korelasyon katsayılarının %5'ten büyük bir kısmı gözlenen korelasyon katsayısından fazladır ve bu bulgu sonucun istatistiksel olarak manidar olmadığı şeklinde yorumlanabilir.

Yukarıda belirtilen işlemleri içeren bir fonksiyon aşağıda sunulmuştur. Korelasyon analizi için randomizasyon testi hesaplamada kullanılabilecek "rand.cor" isimli bu fonksiyon girdi olarak üç argüman içermektedir. İlk iki argüman arasındaki korelasyon hesaplanacak sürekli değişkenler iken son argüman gerçekleştirilecek tekrar sayısı başka bir ifade ile yeniden üretilecek örneklem sayısıdır. Bu fonksiyonu çalıştırmadan önce "knitr" fonksiyonunu indirmeniz gerekmektedir. Daha önce de belirtildiği gibi bu fonksiyon çıktısının çizelge şeklinde sunulması açısından önemlidir.

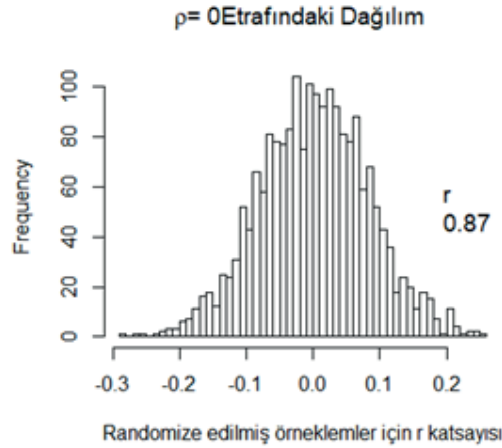
```
rand.cor<-function(x,y,rep=2000) {
  r.göz <- cor(x,y)
  r.random <- numeric(rep)
  for (i in 1:rep) {
    Y <- y
    X <- sample(x, length(x), replace = FALSE)
    r.random[i] <- cor(X,Y)
  }
  ols <- length(r.random[r.random >= r.göz])/rep
  hist(r.random, breaks = 50, main = expression(paste(,rho, "= 0", "Etrafındaki Dağılım ")),
       xlab = "Randomize edilmiş örneklem için r katsayısı")
  r.göz <- round(r.göz, digits = 2)
  abline(v=r.göz,col="red",lwd=2)
  legend("right", legend = c("r",paste(round(r.göz,2))),bty="n",cex=1.2)

  if(ols<0.05) {cat( paste("Gerçekleştirilen", rep,
    " tekrar sonucunda iki değişken arasındaki ilişkinin
    İstatistiksel olarak MANİDAR OLDUĞU belirlenmiştir.(p<0.05)") ) }
  else
  {cat(paste("Gerçekleştirilen", rep,
    " tekrar sonucunda iki değişken arasındaki ilişkinin
    istatistiksel olarak MANİDAR OLMADIĞI belirlenmiştir.(p<0.05)") ) }
  library(knitr)
  return(kable(data.frame(randomizasyon_p.değeri=ols,gözlenen_kor.değeri=r.göz) ))
}
```

Yukarıda verilen "rand.cor" fonksiyonun örnek bir çıktısı Şekil 6'da sunulmuştur.

Gerçekleştirilen 2000 tekrar sonucunda iki değişken arasındaki ilişkinin istatistiksel olarak MANİDAR OLDUĞU belirlenmiştir.
($p < 0.05$)

randomizasyon_p.değeri	gözlenen_kor.değeri
0	0.87



Şekil 6. “rand.cor” fonksiyonun örnek bir çıktısı

Bağımsız Gruplar için Tek yönlü Varyans Analizi ve Randomizasyon Testleri

Bağımsız gruplar için tek yönlü varyans analizi (Tek Yönlü ANOVA) popülasyona ait 2 veya daha fazla ortalamaların karşılaştırılmasında kullanılan bir yöntemdir. ANOVA ortalamalar arasındaki farklılıkları modelde yer alan varyans bileşenlerini analiz ederek karşılaştıran bir yöntemdir (Albert & Rizzo, 2012). Eğer gruplar arasında bir farklılık yoksa gruplar arası ve gruplar içi hata karelerinin ortalamalarının benzer olması beklenir. Diğer taraftan gruplar arasında bir farklılık varsa gruplar arası hata karelerinin ortalaması (bu değer gruplar arası varyansı içermektedir) farkın olmadığı duruma göre daha büyük olacaktır. Bir tek Yönlü ANOVA analizi neticesinde anlamlı çıkan F değeri yokluk hipotezinin reddedildiği anlamına gelir ve ortalama puanları karşılaştırılan gruplar arasında en az ikisi arasındaki farkın istatistiksel olarak manidar olduğu şeklinde yorumlanabilir. Anlamlı çıkan bir F değeri neticesinde çoklu karşılaştırma testleri ile farklılıkların hangi gruplar arasında olduğu tespit edilir.

Tek yönlü ANOVA için randomizasyon testi hesaplama süreci benzer süreçler içermektedir. Bu süreç Örnek 4’te R kodları ile açıklanmıştır.

Örnek 4: Bu örnekte farklı sosyo-ekonomik düzeye sahip bireylerin kitap okumaya yönelik tutum puanlarını içeren yapay bir veri seti üretilmiş ve bu veri seti üzerinde Tek Yönlü ANOVA için randomizasyon testlerinin aşamaları sunulmuştur.

```
SED<-sample(c("ALT","ORTA","ÜST"),45,replace=TRUE) # yapay SED değişkeni Üretme
SED
[1] "ÜST" "ORTA" "ÜST" "ALT" "ORTA" "ORTA" "ALT" "ÜST" "ALT" "ORTA"
[11] "ÜST" "ALT" "ALT" "ALT" "ÜST" "ORTA" "ÜST" "ORTA" "ORTA" "ORTA"
[21] "ALT" "ÜST" "ÜST" "ORTA" "ALT" "ALT" "ORTA" "ALT" "ÜST" "ALT"
[31] "ORTA" "ORTA" "ÜST" "ORTA" "ALT" "ÜST" "ALT" "ÜST" "ÜST" "ORTA"
[41] "ÜST" "ORTA" "ALT" "ÜST" "ORTA"
kitap<-round(rnorm(45,50,10)) # kitap okumaya yönelik tutum verisi üretme
kitap
[1] 51 40 46 57 43 44 47 50 46 45 52 47 56 39 66 53 39 54 64 21 58 68 57 47 66
[26] 46 47 50 64 25 44 58 29 51 64 55 49 60 76 70 55 38 42 47 53
```

İlk aşama olarak mevcut veriler üzerinde gözlenen F istatistiğinin hesaplanması gerekmektedir.


```

model<-aov(kitap~SED)# ANOVA modelini hesaplama
göz.F<-anova(model)[[4]][1] # Model içerisinde gözlenen F değerini çağırma

```

Yukarıda verilen komutlar ile kitap okumaya yönelik tutumun SED'e göre değişip değişmediğini belirlemeye yönelik tek yönlü ANOVA hesaplanmış ve akabinde analiz çıktıları içerisinde randomizasyon testi için gerekli olan "F" değeri çekilip "göz.F" isimli bir obje içerisinde kaydedilmiştir. Aşağıdaki kodlar ile randomizasyon testinin ileriki aşamaları açıklanmıştır.

```

tekrar<-5000
bos.F <- numeric(nreps)
counter <- 0
for (i in 1:tekrar) {
  karıştır <- sample(kitap)
  yeniF <- anova(lm(karıştır~SED))
  bos.F[i] <- yeniF$"F value"[1]
  if (bos.F[i] > göz.F) counter = counter + 1
}
üst <- length(bos.F[bos.F >= göz.F])
alt<- length(bos.F[bos.F <= -göz.F])
olasılık <- (üst + alt)/tekrar
> olasılık
[1] 0.3072

```

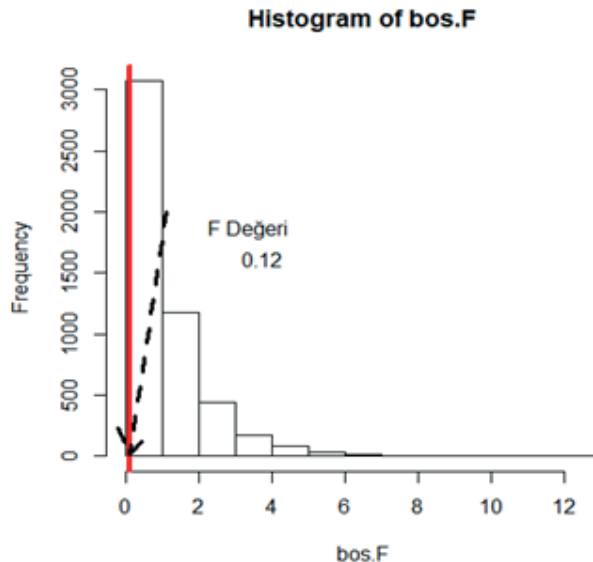
Yukarıda verilen R kodlarında öncelikle yeniden üretilecek örneklem sayısı belirlenmiş ve üretilecek yeni örneklem- lere ilişkin hesaplanan "F" değerlerinin kaydedileceği 5000 gözlemden oluşan "bos.F" adında boş bir vektör oluşturul- muştur. Akabinde "for" döngüsünden faydalanılarak yeni örneklem- ler üretilmiş her biri için "F" değeri hesaplanmış ve bu değerler "bos.F" isimli objenin içerisine kaydedilmiştir. Son olarak randomizasyon testine ilişkin p değeri (rastgele üretilen örneklem-lerden elde edilen F değerlerinin yüzde kaçının gözlenen F değerinden büyük olduğu) hesaplanmış ve olasılık isimli bir objeye kaydedilmiştir. Hesaplanan bu değer yaklaşık 0.31 olarak belirlenmiştir ve bu durum ortalama puanlar arasındaki farkın şansa ortaya çıkma olasılığının yüksek olduğu başka bir ifade ile sonucun istatistiksel olarak anlamlı olmadığı şeklinde yorumlanabilir.

Aşağıdaki kodlar ile gerçekleştirilen randomizasyon testine ilişkin histogram grafiği oluşturulmuştur (Şekil 7).

```

hist(bos.F)
abline(v=göz.F, col="red", lwd=4)
arrows(göz.F + 1, 2000, göz.F, 10, lwd=3, lty=2)
legend(göz.F+2, 2000, round(göz.F,2), bty = "n", title = "F Değeri")

```



Şekil 7. Randomizasyon testi ile elde edilen F değerlerinin dağılımı

Yukarıda belirtilen işlemleri içeren bir fonksiyon aşağıda sunulmuştur. Korelasyon analizi için randomizasyon testi hesaplamada kullanılacak "rand.cor" isimli bu fonksiyon girdi olarak üç argüman içermektedir. İlk iki argüman ara- sındaki korelasyon hesaplanacak sürekli değişkenler iken son argüman gerçekleştirilecek tekrar sayısı başka bir ifade ile yeniden üretilecek örneklem sayısıdır. Bu fonksiyonu çalıştırmadan önce "knitr" fonksiyonunu indirmeniz gerekmekte-

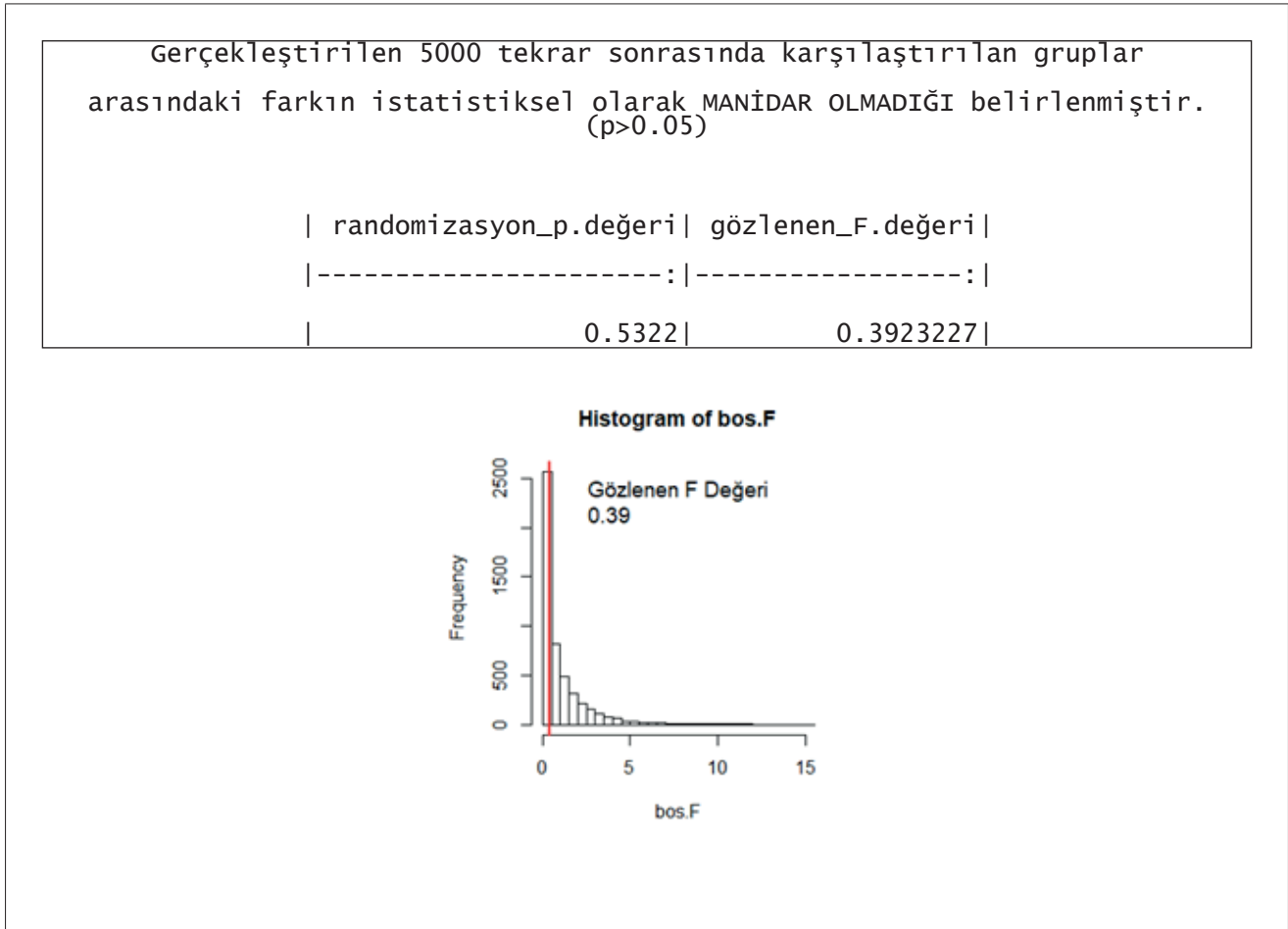
dir. Daha önce de belirtildiği gibi bu fonksiyon çıktısının çizelge şeklinde sunulması açısından önemlidir.

```

rand.anova<-function(grup,puan,tekrar=2000) {
  model<-aov(puan~grup)
  göz.F<-anova(model)[[4]][1]
  bos.F <- numeric(tekrar)
  for (i in 1:tekrar) {
    karıştır <- sample(puan)
    yeniF <- anova(lm(karıştır~grup))
    bos.F[i] <- yeniF$"F value"[1]
  }
  üst <- length(bos.F[bos.F >= göz.F])
  alt<- length(bos.F[bos.F <= -göz.F])
  olasılık <- (üst + alt)/tekrar
  hist(bos.F,breaks = 50,plot = TRUE)
  abline(v=göz.F,col="red",lwd=2)
  legend("top",legend= c("Gözlenen F Değeri",paste(round(göz.F,2))),bty="n",cex=1.2)
  library(knitr)
  res<-data.frame(randomizasyon_p.değeri=olasılık,gözlenen_F.değeri=göz.F)
  res2<-kable(res)
  if(olasılık<0.05) {cat( paste("Gerçekleştirilen", tekrar,
  " tekrar sonrasında karşılaştırılan gruplar arasındaki farkın
  istatistiksel olarak MANİDAR OLDUĞU belirlenmiştir. (p<0.05)")) }
  else
  {cat(paste("Gerçekleştirilen",tekrar,
  "tekrar sonrasında karşılaştırılan gruplar arasındaki farkın
  istatistiksel olarak MANİDAR OLMADIĞI belirlenmiştir. (p>0.05)")) }
  return(res2)
}

```

Yukarıda sunulan "rand.anova" isimli fonksiyonun örnek bir çıktısı şekil 8'de sunulmuştur.



Şekil 8. “rand.anova” fonksiyonu örnek çıktısı

2. Sonuç

İlgili varsayımların karşılanmadığı durumlarda parametrik testlere kıyasla daha güvenilir sonuçlar üreten parametrik olmayan testler evren dağılımına ilişkin bazı varsayımları gerektirmez. Geleneksel parametrik olmayan testler sıra puanları ve sıra ortalamalarını dikkate alarak gruplar arasındaki farklılıkları veya değişkenler arasındaki ilişkileri belirlemeye çalışır. Yeniden örnekleme yöntemlerinde ise evren dağılımına ilişkin varsayımların karşılanmadığı durumlarda daha farklı bir yaklaşım izlenir. Yeniden örnekleme uygulamaları arasında yaygın olarak ön plana çıkan iki yaklaşım ise randomizasyon testleri ve bootsrap yöntemidir. Bu makalede randomizasyon testlerinin temel mantığı ve R programlama dili üzerinden örnek uygulamaları sunulmuştur. Türkçe alan yazında randomizasyon testlerine ilişkin çok kısıtlı kaynakların olması bu çalışmanın önemini artıran bir boyuttur. Ayrıca randomizasyon testlerinin son yıllarda yaygınlaşan R programlama dili üzerinden ele alınmasının uygulayıcılara yol gösterici olacağı düşünülmektedir.

Yeniden örnekleme yaklaşımlarından randomizasyon testlerinin kullanılması, evren dağılımına ilişkin varsayımların karşılanmadığı veya örnekleme büyüklüğünün çok küçük olduğu durumlarda araştırmacılara etkili bir çözüm sunmaktadır. Ayrıca parametrik testlerin ilgili varsayımlarının karşılandığı durumlarda bile randomizasyon testlerinin kullanılması daha güvenilir sonuçlar elde edilmesine katkı sağlayabilir. Bu anlamda araştırmacılara veri çözümleme sürecinde yeniden örnekleme yöntemlerinden faydalanmaları önerilir.

Bu makalede yer alan R kodları ve fonksiyonları pedagojik amaçlı yazılmıştır. Başka bir ifade ile R kodları okuyuculara randomizasyon testlerinin açıklanması sürecinde kullanılmıştır. Bu anlamda okuyucular bu makalede yer alan kod ve fonksiyonlardan faydalanarak randomizasyon testlerinin hesaplanma sürecine ilişkin bilgi ve becerileri edinebilirler. Ancak bu çalışma istatistiksel bir paket veya fonksiyon geliştirme amacı taşımadığı için yazılan kod ve fonksiyonlar ile elde edilen sonuçlar geçerli ve güvenilir başka yazılımların sonuçları ile karşılaştırılmamış farklı koşul ve durumlarda kod ve fonksiyonların işlerliği test edilmemiştir. Bu bağlamda bu makalede yer alan kod ve fonksiyonların pedagojik (eğitsel) amaçlı kullanılması önerilir.

3. Kaynakça

- Albert, J., & Rizzo, M. (2012). *R by example*. Springer New York. USA
- Banjanovic, Erin S. & Osborne, Jason W. (2016). Confidence Intervals for Effect Sizes: Applying Bootstrap Resampling. *Practical Assessment, Research & Evaluation*, 21(5).
- Beaujean, A. A. (2013). Factor analysis using R. *Practical Assessment: Research & Evaluation*. 18(4) <http://pareonline.net/getvn.asp?v=18&n=4>
- Bradley, J. V. (1968). *Distribution free statistical tests*. Englewood Cliffs, NJ: Prentice-Hall.
- Chernick, M. R., & Labudde, R. A. (2011). *An introduction to bootstrap methods with applications to R*. A John Wiley & Sons, Inc. New Jersey.
- Davison, A. C., & Hinkley, D. V. (1997). *Bootstrap methods and their application*. Cambridge, United Kingdom: Cambridge University Press.
- Doğan, C.D. (2017). Applying Bootstrap Resampling to Compute Confidence Intervals for Various Statistics with R. *Eurasian Journal of Educational Research* (68), 1-18.
- Edgington, E. S. (1986). Randomization tests. In S. Kotz & N. L. Johnson (Eds.), *Encyclopedia of statistical sciences*, Vol. 7 New York, NY: Wiley. 530–538.
- Field, A. (2018). *Discovering Statistics Using SPSS*. 5th ed. London: Sage Publication.
- Gibbons, J. D. (1986). Permutation tests. In S. Kotz & N. L. Johnson (Eds.), *Encyclopedia of statistical sciences*, Vol. 6 . New York, NY: Wiley. 690
- Howel, D. C. (2007). *Statistical Method for Psychology*. Wadsworth, Cengage Learning. USA
- Onghena, P., & May, R. B. (1995). Pitfalls in computing and interpreting randomization test p values: A commentary on Chen and Dunlap. *Behavior Research Methods, Instruments, & Computers*, 27, 408–411.
- Onghena, P. (2018). Randomization tests or permutation tests? A historical and terminological clarification. In V. Berger (Ed.), *Randomization, masking, and allocatio concealment* (pp. 209-227). Boca Raton/FL: Chapman & Hall/CRC Press.