**IST3110 R İLE İSTATİSTİK UYGULAMALARI**

**2.VİZE PROJE ÖDEVİ**

**GRUP 8**

**Asst. Prof. Erhan ÇENE**

**daire, amblem, ticari marka, logo içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**GRUP BİLGİLERİ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Student Number** | **Name - Surname** | **Group\_ID** | **Group Representative** |
| **17** | **18023603** | **CEREN ACAR** | **8** | **LEYLA TUNÇ** |
| **12** | **18023049** | **AYLİN ÖZTÜRK** | **8** |
| **7** | **18023018** | **FATMA BENGİSU TEMİR** | **8** |
| **45** | **20023036** | **LEYLA TUNÇ** | **8** |
| **28** | **19023050** | **TUĞÇE YAZICI** | **8** |  |

**PART A**

**1. DATA**

Veri setimizde 8 farklı restorana ait menülerin kalori miktarı ,yağdan gelen kalori miktarı,yağ miktarı,doymuş yağ miktarı,trans yağ miktarı, kolesterol, sodyum, karbonhidrat ,lif , şeker ,protein, a vitamini, c vitamini,kalsiyum miktarı ve menü içeriğindeki salata türü değişkenleri bulunmaktadır.

**Table 1.1. Description of variables**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Değişkenler** | **Tanım** | **Veri Tipi** |
| **1** | **restaurant** | **8 farklı restoran** | **chr** |
| **2** | **item** | **Menü isimleri** | **chr** |
| **3** | **calories** | **Menülerdeki kalori miktarı** | **int** |
| **4** | **cal\_fat** | **Menülerdeki yağdan gelen kalori miktarı** | **int** |
| **5** | **total\_fat** | **Menülerdeki yağ miktarı** | **int** |
| **6** | **sat\_fat** | **Menülerdeki doymuş yağ miktarı** | **num** |
| **7** | **trans\_fat** | **Menülerdeki trans yağ miktarı** | **num** |
| **8** | **cholesterol** | **Menülerdeki kolestrol miktarı** | **int** |
| **9** | **sodium** | **Menülerdeki sodyum miktarı** | **int** |
| **10** | **total\_carb** | **Menülerdeki karbonhidrat miktarı** | **int** |
| **11** | **fiber** | **Menülerdeki lif miktarı** | **int** |
| **12** | **sugar** | **Menülerdeki şeker miktarı** | **int** |
| **13** | **protein** | **Menülerdeki protein miktarı** | **int** |
| **14** | **vit\_a** | **Menülerdeki a vitamini miktarı** | **int** |
| **15** | **vit\_c** | **Menülerdeki c vitamini miktarı** | **int** |
| **16** | **calcium** | **Menülerdeki kalsiyum miktarı** | **int** |
| **17** | **salad** | **Menülerde salata olup olmadığı** | **chr** |

**VERİ MANİPULASYONU**

Öncelikle verimizi anlamak ve nasıl bir yol izleyeceğimizi belirlemek adına birkaç adımdan oluşan veri analizi araştırmasını uygulayacağız. Verimizi R programında okutup verilerin karakter özelliklerini inceleyebiliriz.

**df <- read.csv("fast\_food.csv", sep=";")**

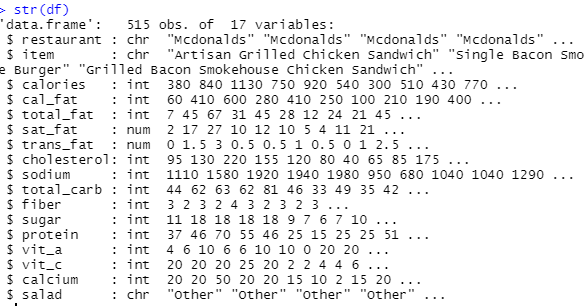
**df**

**head(df)**

**str()**

#Bu komutta str() fonksiyonuyla değişkenlerin karakter yapısını anlayıp değiştirilmesi gereken veri tipini anlarız.

#Output



Anlaşılacağı üzere verimizde numeric, integer ve char olan veriler bulunmakta ve bu veriler için bir değişikliğe ihtiyaç yok gibi görünüyor.

Daha sonra verideki değişkenlerde boş değer olup olmadığına bakalım.

**is.na(df)**

#Bu kodun çıktısı TRUE- FALSE şeklinde gelir ve kalabalık veride okunması ve anlaşılması güçtür. Onun yerine colSum() fonksiyonu kullanarak 3-16 sütunları arasındaki numeric ve integer değişkenlerin hangilerinde kaç adet null değer var rahatlıkla görebiliriz.

**colSums(is.na(df[3:16]))**

#Output

calories , cal\_fat ,total\_fat , sat\_fat , trans\_fat cholesterol sodium total\_carb fiber

0 0 0 0 0 0 0 0 12

sugar protein vit\_a vit\_c calcium

0 1 214 210 210

Fiber , protein, vit\_a, vic, calcium değişkenlerinde eksik gözlem vardır. Öyleyse uygun verilerle dolduralım.Boş değerleri sütun ortalamaları ile doldurursak;

**numeric\_cols <- sapply(df, is.numeric)**

**df[numeric\_cols] <- lapply(df[numeric\_cols], function(x) {**

**ifelse(is.na(x), mean(x, na.rm = TRUE), x)**

**})**

**colSums(is.na(df[3:16]))**

#Kodun açıklaması şöyledir: sapply() fonksiyonu ile df verisinden her sütunun numerikliğini test eder ve numerik değişkenleri numeric\_cols olarak atar. lapply() fonksiyonu, seçilen numeric değişkenler üzerinde belirli bir işlemi uygular. İşlem, her sütun için ayrı ayrı gerçekleştirilir. ifelse() fonksiyonu, bir koşulu kontrol eder ve koşul doğru ise bir değer döndürür, aksi takdirde başka bir değeri döndürür. Bu durumda, x değişkeninin boş değerleri (is.na(x)) kontrol edilir. Boş değer varsa, mean(x, na.rm = TRUE) kullanılarak bu sütundaki diğer değerlerin ortalaması alınır. Eğer boş değer yoksa, x değeri korunur.Çıktı sonucu tüm boş değerler doldurulur.

**df %>% filter(calories >2000)**

Daha sonra kalori bakımından 2000 cal dan yüksek olan menü var mı diye baktığımızda 1 tane menü karşımıza çıkıyor. Bu da Mcdonalds restoranında ki 2430 cal değerinde bir menüdür.

Veride integer ve numeric değişkenlerin adlarını bulan bir fonksiyon oluşturarak dilediğimiz zaman değişkenleri bize getiren bir fonksiyon oluşturalım.

**int\_num\_col\_names <- function(df) {**

**cols <- names(df)**

**int\_num\_cols <- c()**

**for (col in cols) {**

**if (is.integer(df[[col]]) || is.numeric(df[[col]])) {**

**int\_num\_cols <- c(int\_num\_cols, col)**

**}**

**}**

**return(int\_num\_cols)**

**}**

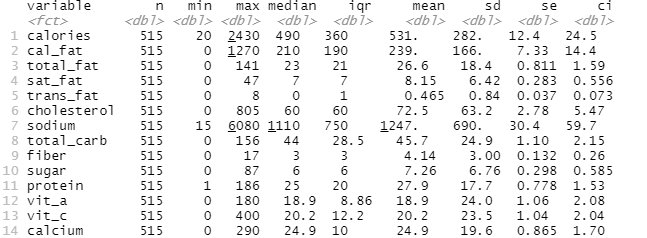
**int\_num\_col\_names(df)**

Kodu açıklayacak olursak ; ilk başta int\_num\_cols diye fonksiyon adı oluştururuz. Ardından tüm sütunların adlarını alırız.İnt ve num değişkenleri koyacağımız boş bir vektör oluştururuz.For döngüsüyle her sütunu dolaşıp num ya da int olan değişkenleri if yapısında kullandık.Daha sonra return ile num ile int sütun adlarını döndürürüz.Bu fonksiyon genellikle veri analizi uygulamalarında sıkça kullanılan bir fonksiyondur.

Veriye bir de istatiksel olarak bakmak istersek:

**df %>% get\_summary\_stats(type='common')**

# Output:



Bazı değişkenlerin standart sapmaları ortalamadan büyük çıkmıştır. Bu durum genellikle verilerin dağılımının geniş olduğunu ve değişkenliğin yüksek olduğunu gösterir. Bu durumda, verilerin ortalamadan daha uzak noktalara sahip olduğu ve geniş bir aralığa yayıldığı söylenebilir.

Frekanslarına da bir göz atarsak;

**df %>% select(where(is.numeric)) %>%**

**apply(2, freq\_table)**

Bu kod ile numeric olan değişkenleri select ile seçer ve apply fonksiyonu ile frekans tablosu oluşturulur.

**1.NORMALLİK TESTİ**

Verinin normal dağılıp dağılmadığını fikir olması adına görsel olarak hızlıca bakmak istersek histogram grafiği ile bir bakış yapabiliriz. Kendi oluşturduğumuz fonksiyonu çağırıp numeric ve integer değerleri tekrar hatırladıktan sonra aşağıdaki gibi görselleri inceleyebiliriz.

**int\_num\_col\_names(df)** #Fonksiyonu çağırdık.

**hist(df$calories)**

**hist(df$cal\_fat)**

**hist(df$sodium)**

**hist(df$total\_fat)**

**hist(df$sat\_fat)**

**hist(df$trans\_fat)**

**hist(df$cholesterol)**

**hist(df$total\_carb)**

**hist(df$fiber)**

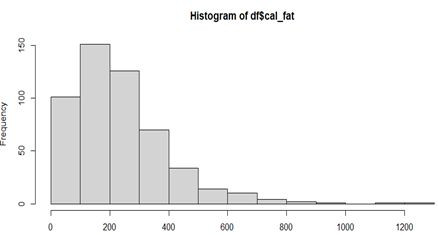
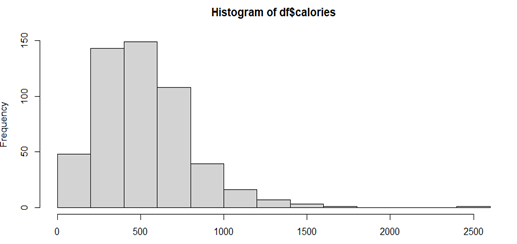
**hist(df$sugar)**

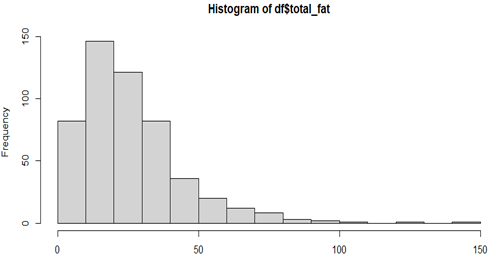
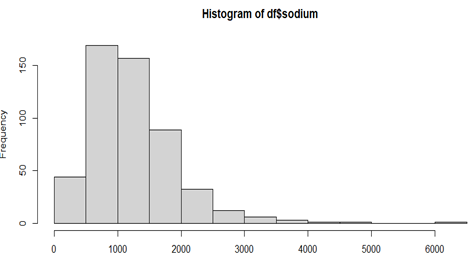
**hist(df$protein)**

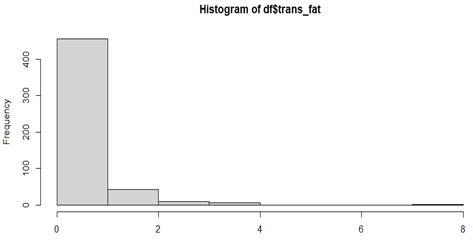
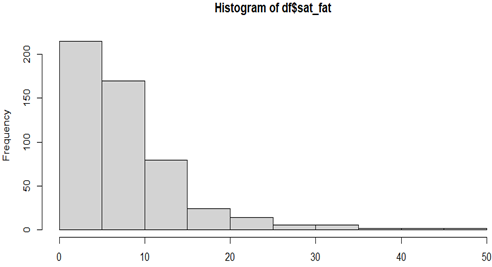
**hist(df$vit\_a)**

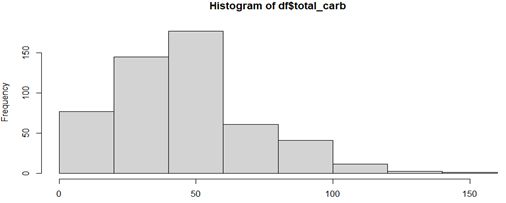
**hist(df$vit\_c)**

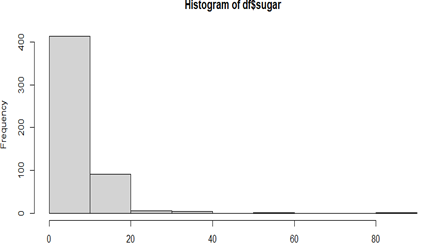
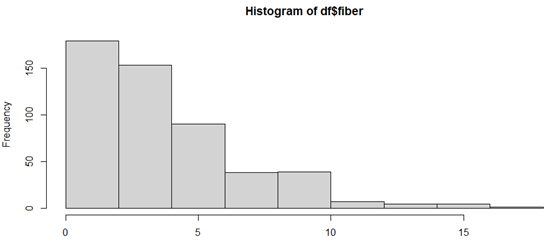
**hist(df$calcium)**

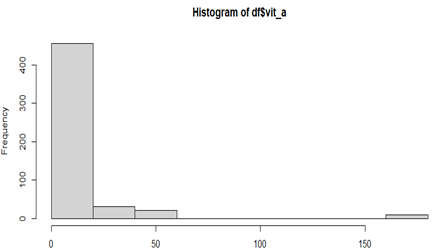
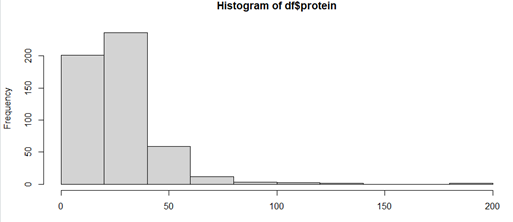


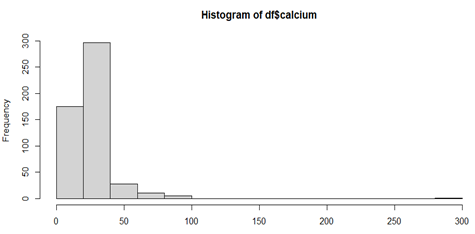
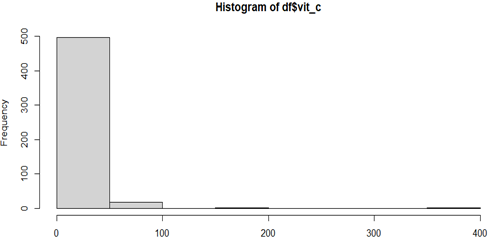












**SHAPIRO WILKS TESTI**

Modeli kurmaya gelecek olursak verinin normalliği Shapiro Wilks testi ile incelenebilir. Her bir değişkenin normal olup olmadığına bakabiliriz. Hipotezimizi kurduktan sonra ;

H0:Veri normal dağılmıştır.

H1:Veri normal dağılmamıştır.

Alpha Değeri:%95

R kodu ile;

**df %>% select(where(is.numeric)) %>% apply(2, shapiro\_test)**

|  |  |  |  |
| --- | --- | --- | --- |
| calories:  p-value:9.59e-16 | cal\_fat:  p-value:6.33e-20 | total\_fat:  p-value:5.45e-20 | sat\_fat:  p-value:9.55e-23 |
| trans\_fat:  p-value:1.76e-32 | cholesterol:  p-value:5.10e-29 | sodium:  p-value:5.36e-19 | total\_carb:  p-value:2.56e-11 |
| fiber:  p-value:2.81e-20 | sugar:  p-value:3.61e-29 | protein:  ,p-value:3.46e-24 | vit\_a:  p-value:4.38e-37 |
| vit\_c:  p-value:3.71e-36 | calcium:  p-value:1.64e-31 |  |  |

Bu çıktı sonucu tüm değişkenlerin p value değeri 0.05 den oldukça düşük.H0 hipotezi reddetmek ile yapacağımız hata her değişken için çok küçük.Böylelikle reddederiz. Veri normal dağılıma uygun değildir.

Veri normal dağılmıyorsa non-parametrik testlerden olan Mann Whitney testi ile aynı dağılımdan gelip gelmemesini test edebiliriz.

**2.MANN WHITHEY-U**

**H0:** Burger King ve McDonalds restoranlarının kalori değerlerinin medyanları arasında fark yoktur

**H1:** Burger King ve McDonalds restoranlarının kalori değerlerinin medyanları arasında fark vardır.

**Tablo**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Restoran** | **n** | **Ortalama Kalori**  **(ortalama ± std sapma)** | **W** | **p** |
| Burger King | 70 | 608.571 ± 290 | 2019.5 | 0.9074 |
| McDonald’s | 57 | 640 .350 ± 411 |

Mann-Whitney U testi,bağımsız iki grup arasındaki medyan farkını değerlendirmek için kullanılan bir non-parametrik testtir.Bu test gruplardaki verilerin sıralamalarını kullanarak istatistiksel olarak anlamlı bir fark olup olmadığını belirler.W değeri test istatistiğini temsil eder ve iki grup arasındaki sıralamaların toplamını ifade eder.Burger King ve McDonalds restoranları arasında kalorin medyanı arasında anlamlı bir fark yoktur. W değeri 2019.5 ve p-değeri 0.9074'tür.P değeri 0.05 anlamlılık düzeyinden daha büyük olduğu için,iki restoranın kalori seviyeleri arasında anlamlı bir fark yoktur.

**Kod:**

# Burger King ve McDonald's restoranlarini filtreleme

**selected\_data <- df %>%**

**filter(restaurant %in% c("Burger King", "Mcdonalds"))**

**selected\_data <- selected\_data %>%**

**select(restaurant, calories) %>%**

**mutate(calori\_avg = as.numeric(calories))**

# Mann-Whitney U testi

**mannwhitney\_test <- wilcox.test(calori\_avg ~ restaurant, data = selected\_data)**

#Sonuc

**print(mannwhitney\_test)**

**summary(selected\_data)**

# Değiskenler sıralandı

**sorted\_data <- df %>%**

**arrange(calories)**

**sorted\_data**

#Restoranlara göre gruplandi ve özetlendi

**grouped\_summary <- df %>%**

**group\_by(restaurant) %>%**

**summarise(**

**cal\_fat\_avg = mean(cal\_fat),**

**total\_fat\_avg = mean(total\_fat),**

**sat\_fat\_avg = mean(sat\_fat),**

**trans\_fat\_avg = mean(trans\_fat),**

**cholesterol\_avg = mean(cholesterol),**

**sodium\_avg = mean(sodium),**

**total\_carb\_avg = mean(total\_carb),**

**fiber\_avg = mean(fiber)**

**)**

**grouped\_summary**

**library(ggplot2)**

**# GGplot kullanarak grafik olusturma**

**ggplot(data = df, aes(x = restaurant, y = calories, fill = restaurant)) +**

**geom\_boxplot() +**

**labs(x = "Restaurant", y = "Calories", title = "Calories by Restaurant") +**

**theme\_minimal()**

**Restoranları filtreleme:**

İlk olarak, data1 veri setinde bulunan restaurant sütununu kontrol eder.

**filter()** fonksiyonu, sadece "Burger King" ve "Mcdonalds" restoranlarına ait satırları seçer.

selected\_data adlı yeni bir veri seti oluşturulur.

**Sütunları seçme ve dönüşüm:**

selected\_data veri setindeki sütunları seçmek için **select()** fonksiyonu kullanılır.

Yalnızca "restaurant" ve "calories" sütunları seçilir.

**mutate()** fonksiyonu ve **as.numeric()** işlevi, "calories" sütununu sayısal bir formata dönüştürür.

Dönüştürülen sütun, "calori\_avg" adıyla selected\_data veri setine eklenir.

**Mann-Whitney U testi:**

**wilcox.test()** fonksiyonu, "calori\_avg" değişkeninin "restaurant" değişkenine göre gruplar arasında istatistiksel bir farklılık olup olmadığını test etmek için Mann-Whitney U testini uygular.

mannwhitney\_test adlı bir nesne oluşturulur ve test sonuçlarını içerir.

print() işleviyle test sonuçları ekrana yazdırılır.

**Veri özetleme:**

**summary()** işlevi, selected\_data veri setinin istatistiksel özetini sağlar.

**Değişkenleri sıralama:**

**arrange()** fonksiyonu, "calories" sütununa göre data1 veri setini sıralar.

Sıralanmış veri, sorted\_data olarak adlandırılan yeni bir veri setine atanır.

**Restoranlara göre gruplama ve özetleme:**

**group\_by()** fonksiyonu, "restaurant" sütununa göre data1 veri setini gruplandırır.

**summarise()** fonksiyonu, her restoran için belirli değişkenlerin ortalamalarını hesaplar.

grouped\_summary adlı yeni bir veri seti oluşturulur ve özetleme sonuçlarını içerir.

**GGplot ile grafik oluşturma:**

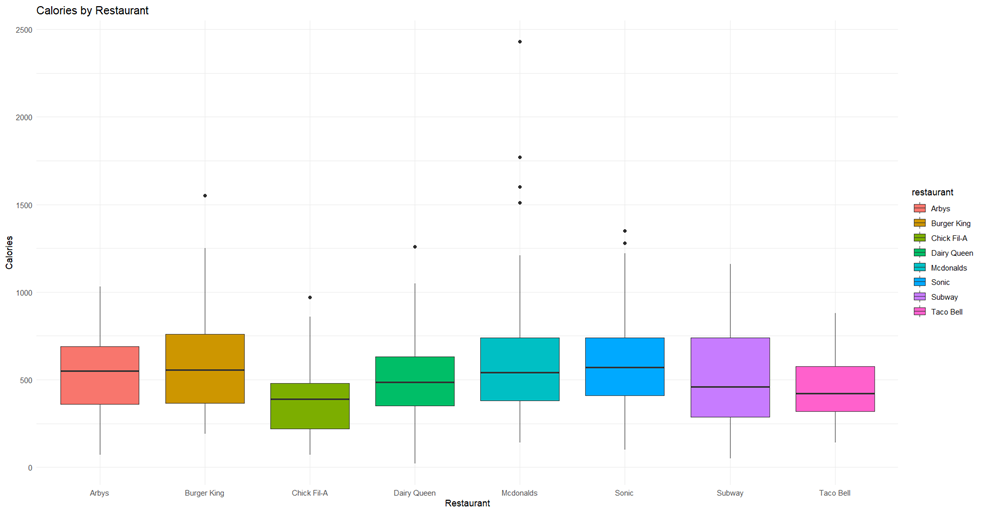
**ggplot()** fonksiyonu, data1 veri setini temel alarak bir grafik nesnesi oluşturur.

**aes()** işlevi, x-ekseni olarak "restaurant" sütununu, y-ekseni olarak "calories" sütununu ve renklendirme için "restaurant" sütununu belirtir.

**labs()** fonksiyonu, grafik etiketlerini belirtir.

**theme\_minimal()** fonksiyonu, minimal bir tema kullanarak grafik görünümünü ayarlar.

Bu şekilde, verilerin Burger King ve McDonalds restoranları arasındaki kalori dağılımını gösteren bir kutu grafiği oluşturulur.



Her bir restoran için kalori değerlerine kutu grafiği ile baktığımızda da McDonald’s ve Burger king restoranlarının menülerinin kalori miktarları arasında bir fark olmadığını görebiliriz. Burger King Chick- Fill-A, Dairy Queen McDonald’s ve Sonic restoranlarında aykırı değerler olduğunu görebiliyoruz.

**3.Kİ-KARE**

Bağımsızlık Ki-Kare Testi, iki kategorik değişken arasındaki ilişkiyi belirlemek için kullanılır. Gözlenen frekansların beklenen frekanslardan önemli ölçüde farklı olup olmadığını tespit eder. Test, iki değişkenin bağımsız mı yoksa ilişkili mi olduğunu belirlemeye yardımcı olur.

Bağımsızlık Chi-Kare Testi'nin formülü aşağıdaki gibi ifade edilir:

X^2 = Σ((O - E)^2 / E)

Burada:

X^2: Ki-Kare test istatistiğini temsil eder.

O: Gözlenen frekansları temsil eder.

E: Beklenen frekansları temsil eder.

Bu testin örnek uygulamaları arasında cinsiyet ve siyasi parti tercihi arasındaki ilişkinin incelenmesi, sigara alışkanlığı ile akciğer kanseri görülme arasındaki ilişkinin analizi veya eğitim düzeyi ile iş memnuniyeti arasındaki ilişkinin araştırılması yer alabilir.

Test, bir ki-kare test istatistiği, serbestlik derecesi ve bir p-değeri sağlayacaktır. P-değeri, değişkenler arasındaki ilişkinin anlamlılığını gösterir. Eğer p-değeri seçilen anlamlılık düzeyinin (örneğin, 0.05) altındaysa, bu değişkenler arasında bir ilişki olduğuna dair kanıt sağlar.

**H0:** Fast food restoranlarındaki kalorilerin kolesterol seviyesiyle ilişkisi yoktur.

**H1:** Fast food restoranlarındaki kaloriler, kolesterol seviyesiyle ilişkilidir.

**kalori\_durum<-ifelse(df$cholesterol>800, "high",**

**ifelse(df$calories>=150 & df$calories<=800,"low","medium"))**

**chisq.test(df$restaurant,kalori\_durum)**

kalori\_durum adında bir değişken tanımlıyoruz. Bu değişkene atanan ifade, "df" veri setindeki "calories" sütununda yer alan değerlere dayanarak "high", "medium" veya "low" değerlerinden birini döndürür. Bu değişkenin değeri, fastfood veri çerçevesindeki calories sütununun 150'den büyük veya eşit ve 800'den küçük veya eşit olması durumunda "low" olarak atanıyor. Eğer fastfood veri çerçevesindeki cholesterol sütunu 800'den büyükse "high" atanıyor. Son olarak, chisq\_test(fastfood$restaurant, kalori\_durum) komutunu kullanarak df veri çerçevesindeki "restaurant" sütunu ile kalori\_durum değişkeni arasında bir ki-kare testi yapıyoruz.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | statistic | p | df | method |
| 515 | 34.33277 | 0.00184 | 14 | Chi-Square Test |

Tabloda df (serbestlik derecesi) değeri 14, p değeri 0.00184, statistic değeri 34.33277 ve n (gözlem sayısı) değeri 515 olarak belirtilmiş.

Ki-kare test sonuçlarını yorumlamak için öncelikle p değerine odaklanabiliriz. P değeri, verideki gözlemlenen ilişkinin tesadüfi mi yoksa istatistiksel olarak anlamlı mı olduğunu belirler. Belirtilen p değeri (0.00184) < 0.05 olduğundan dolayı H0 hipotezi reddedilir. Null hipotezini reddedebilir ve iki değişken arasında anlamlı bir ilişki olduğunu söyleyebiliriz.

Yani,df$restaurant ve kalori\_durum değişkenleri arasında istatistiksel olarak anlamlı bir ilişki olduğunu söyleyebiliriz.

Df değeri, testin istatistiksel dağılımının hesaplanmasında kullanılan serbestlik derecesini ifade eder. Bu durumda, 14 serbestlik derecesine sahip olduğu belirtilmiştir.

İstatistik değeri (statistic) 34.33277 olarak belirtilmiştir. Bu değer, ki-kare test istatistiğinin gözlemlenen değerini temsil eder. Test istatistiği, iki değişken arasındaki farkı ölçer ve ilişkinin gücünü gösterir. Gözlem sayısı (n), analizde kullanılan toplam gözlem sayısını ifade eder. Bu durumda, 515 gözlem bulunmaktadır.

Sonuç olarak, verilerinizdeki df$restaurant değişkeni ile kalori\_durum değişkeni arasında anlamlı bir ilişki olduğunu söyleyebiliriz, çünkü düşük p değeri null hipotezi reddeder.

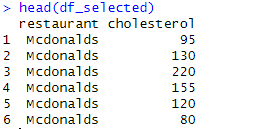
**4.KRUSKAL WALLIS**

**# select()** kullanarak belirli değişkenleri seçme

**df\_selected <- select(df, restaurant, cholesterol)**

**df\_selected**

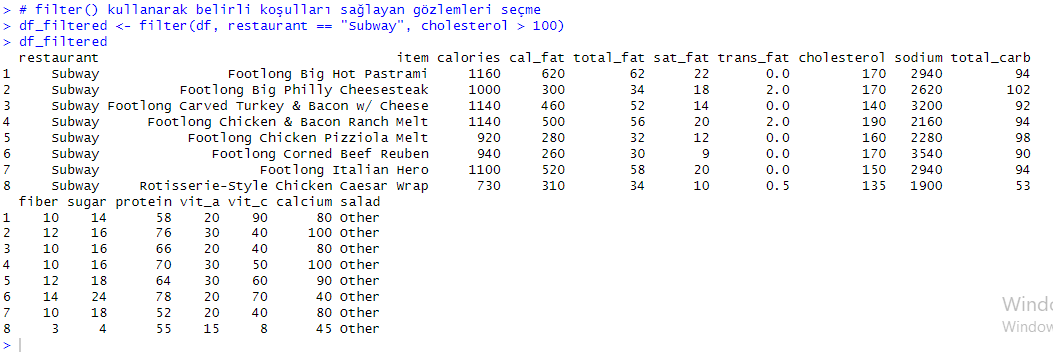
**head(df\_selected)**

****

**# filter**() kullanarak belirli koşulları sağlayan gözlemleri seçme

**df\_filtered <- filter(df, restaurant == "Subway", cholesterol > 100)**

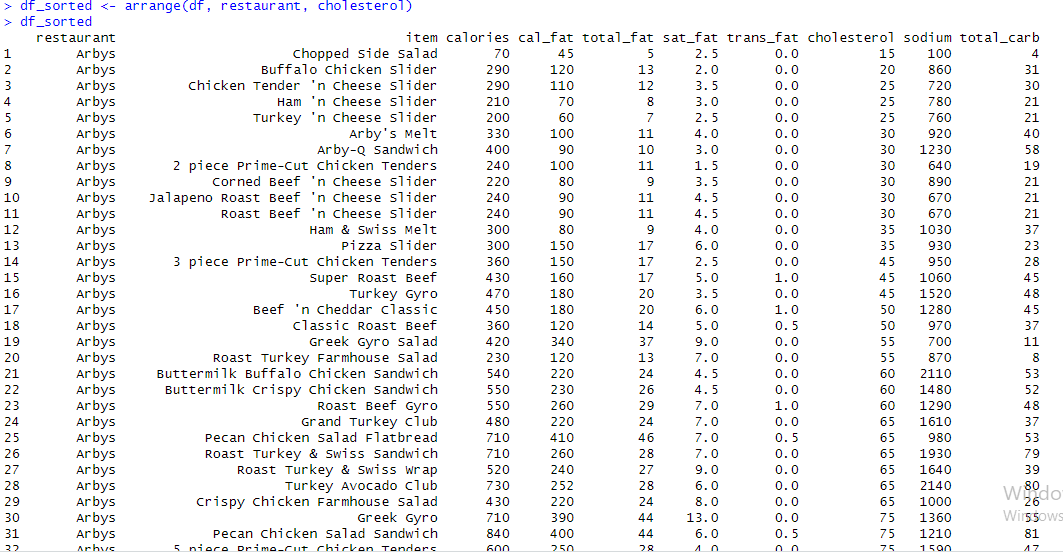
**df\_filtered**



**# arrange()** kullanarak restaurant ve cholesterol değişkenlerine göre sıralama

**df\_sorted <- arrange(df, restaurant, cholesterol)**

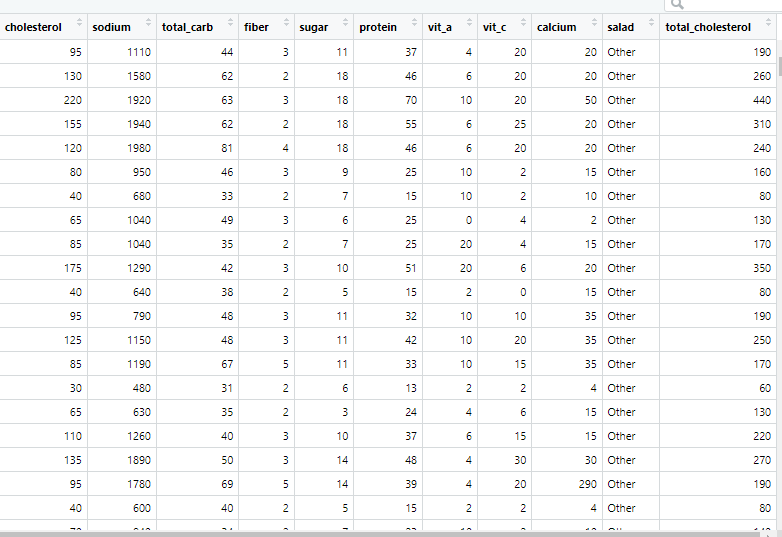
**df\_sorted**



**# mutate()** kullanarak yeni bir değişken oluşturma veya mevcut değişkenleri dönüştürme

**df\_modified <- mutate(df, total\_cholesterol = cholesterol \* 2)**

**df\_modified**



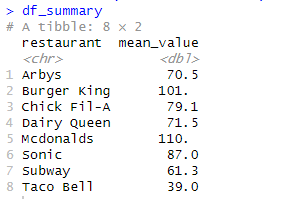
**# group\_by()** ve **summarise()** ile grup ortalamalarını hesaplama

**df\_summary <- df %>%**

**group\_by(restaurant) %>%**

**summarise(mean\_value = mean(cholesterol))**

**df\_summary**

****

Normallik Varsayımı: ANOVA, gruplardaki verilerin normal dağılım göstermesini varsayar. Ancak, gruplardaki veriler normal dağılımı sağlamadığında veya bu varsayımı karşılamadığında Kruskal-Wallis testi kullanılabilir. Kruskal-Wallis testi, verilerin sıralı toplamlarına dayandığı için gruplardaki verilerin dağılımıyla ilgili spesifik bir varsayım yapmaz.Başta yaptığımız normallik testinde verimizin normal olmadığını gözlemledik. . Veri seti normal bir dağılıma sahip değildir.

Bu durumda anova kullanılmaz. Onun alternatifi olan Kruskal - Wallis testi kullanılır.

**H0:** Gruplar arasında fark yoktur.

**H1:** En az bir grup diğerlerinden farklıdır.

**df %>% kruskal\_test(cholesterol ~ restaurant)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| y | n | statistic | df | p |
| cholesterol | 515 | 102. | 7 | 3.69e-19 |

p < 0.05 olduğundan dolayı H0 hipotezi reddedilir. Gruplardan en az biri farklı dağılımdan geliyor.

Restoranlardan en az biri diğerlerinden farklı kolesterol ortalamasına sahiptir.

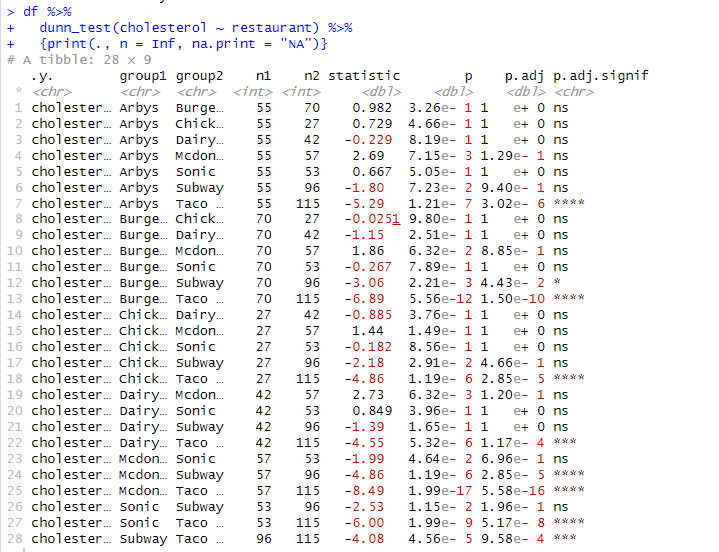
Dunn testi Kruskal-Wallis testi sonucunda gruplar arasında anlamlı bir fark olduğu belirlendiğinde, hangi grupların birbirinden farklı olduğunu belirlemek için kullanılır.

**H0:** Restoranlar ile kolesterol düzeyi arasında anlamlı bir fark yoktur.

**H1**: En az bir restoran ile kolesterol düzeyi arasında anlamlı bir fark vardır.

**df %>% dunn\_test(cholesterol ~ restaurant) %>%**

**{print(., n = Inf, na.print = "NA")}**



Burada p değerlerine bakarız. Eğer p<0.05 ise h0 hipotezi reddedilir. İkili arasında anlamlı bir farkın olduğu sonucuna varırız. Eğer p>0.05 ise h0 hipotezi reddedilemez. İkili arasında anlamlı bir farkın olmadığı sonucuna varılır. Arbys ve Taco Bell restoranları arasındaki fark çok yüksek anlamlılık düzeyindedir. Burger King ve Subway restoranları arasındaki fark anlamlıdır. Burger King ve Taco Bell restoranları arasındaki fark çok yüksek anlamlılık düzeyindedir. Chick Fil-A ve Tacı Bell restoranları arasındaki fark çok yüksek anlamlılık düzeyindedir. Dairy Queen ve Taco Bell restoranları arasındaki fark anlamlıdır. Mcdonalds ve Subway restoranları arasındaki fark çok yüksek anlamlılık düzeyindedir. Mcdonalds ve Taco Bell restoranları arasındaki fark çok yüksek anlamlılık düzeyindedir. Sonic ve Taco Bell restoranları arasındaki fark çok yüksek anlamlılık düzeyindedir. Subway ve Taco Bell restoranları arasındaki fark anlamlıdır.

ggboxplot veri setinin dağılımını görselleştirmek, gruplar arasındaki karşılaştırmaları yapmak ve aykırı değerleri belirlemek için kullanılan bir grafik yöntemidir.

**library(ggpubr)**

**ggboxplot(df, x='restaurant', y='cholesterol',**

**color = 'restaurant',**

**palette = "jco")**



McDonald’s, Chick- Fill-A, Sonic, Burger King, Subway restoranlarında aykırı değerler olduğunu görebiliyoruz.

**5.KORELASYON**

Korelasyon, iki değişken arasındaki doğrusal ilişkiyi ölçmek için kullanılır.Korelasyon katsayısı -1 ile 1 arasında değerler alır.

0'a yakın değerler ilişkinin olmadığını, -1'e yakın değerler güçlü negatif ilişkiyi ve 1'e yakın değerler güçlü pozitif ilişkiyi gösterir

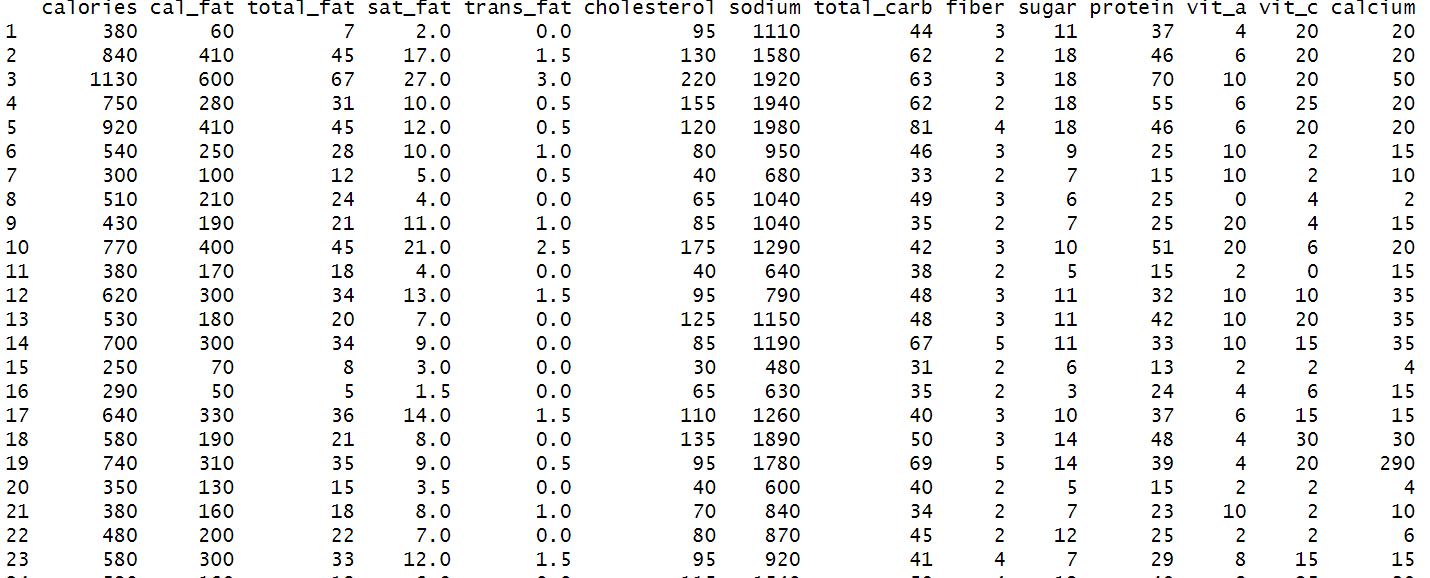
**Öncelikle hipotez kurulur:**

**H0:** Şeker ile kalori arasında anlamlı bir ilişki yoktur.

**H1:** Şeker ile kalori arasında anlamlı bir ilişki vardır.

**>df %>% select(-restaurant) %>% select(where(is.numeric))**

Yukarıdaki kodu kullanarak “fast\_food.csv” verisinden sadece sayısal verileri içeren sütunlar elde edilir.

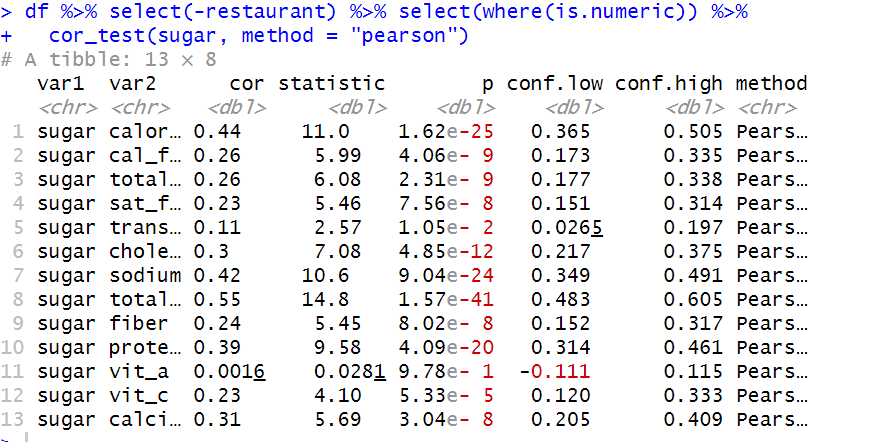


**> df %>% select(-restaurant) %>% select(where(is.numeric)) %>%**

**cor\_test(sugar, calories, method = "pearson")**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| var1 | var2 | cor | statistic | p |
| sugar | calories | 0.44 | 11.0 | 1.62e-25 |

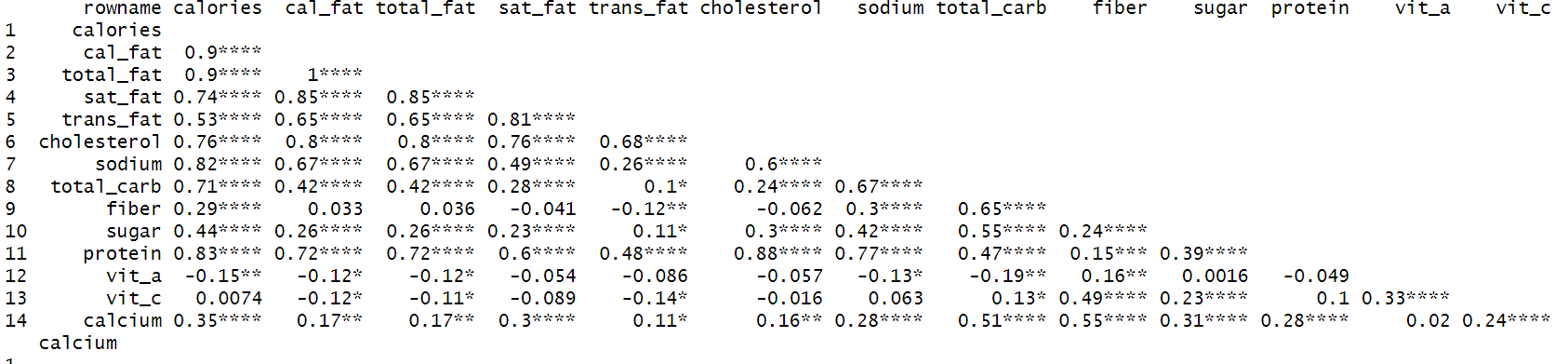
Veri setimizden iki değişkeni(sugar,calories) alarak ikisi arasındaki ilişkiyi değerlendirmek için yukarıdaki Pearson korelasyonu ile analiz edildi ve şeker ile kalori arasında 0.44’lük bir korelasyon bulunduğu ortaya çıktı.Bu anlamda ilişkinin orta düzeyde güçlü olduğunu söyleyebiliriz ve H0 reddedilir diyebiliriz..Güven aralığı ise 0.365-0.505 aralığında yer almakta.Bu analiz sonuçlarına göre ise şeker tüketimi artıkça kalori alımının da arttığını söyleyebiliriz.



Ek olarak “Sugar” değişkeninin diğer değişkenler arasındaki ilişkisini, korelasyon katsayısını elde ettik ve en zayıf ilişkisi “vit\_a”değişkeniyle,en güçlü ilişkisi ise “total\_fat”değişkeniyle olduğunu tespit ettik.

**> df %>% select(-restaurant) %>% select(where(is.numeric)) %>%**

**cor\_mat() %>% cor\_mark\_significant()**

 Bu şekilde de genel olarak tüm değişkenlerin tüm değişkenler arasındaki korelasyon katsayısını ve anlamlılığını gösterebiliriz .

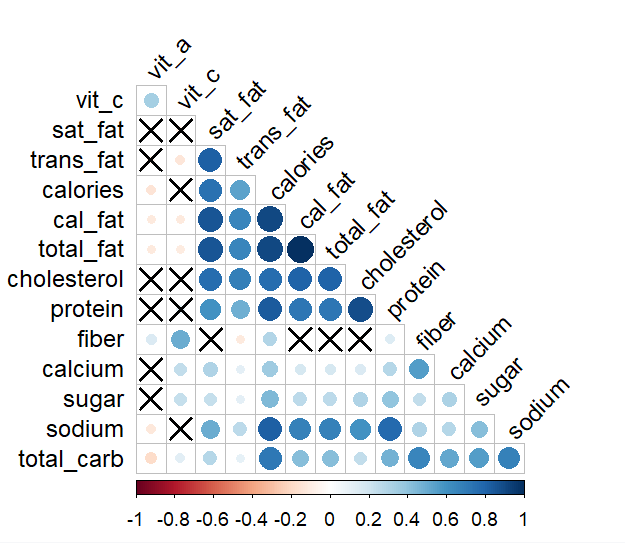
**>df %>% select(-restaurant) %>% select(where(is.numeric)) %>%**

**cor\_mat() %>%**

**cor\_reorder() %>%**

**pull\_lower\_triangle() %>%**

**cor\_plot()**



Son olarak bu grafik ile tüm değişkenlerin birbirine bağlılık derecesini daha net bir şekilde görebiliriz.. Grafik üzerindeki koyu mavi noktalar güçlü pozitif korelasyonları ,kırmızıya dönük olan noktalar ise güçlü negatif korelasyonları göstermektedir.Örneğin,bu grafikte “total\_fat değişkeni ile “cal\_fat” değişkeni arasında güçlü pozitif korelasyon görüldüğünü,”vit\_a”ile “protein” arasında güçlü zayıf korelasyon görüldüğünü açıkça söyleyebiliriz. Bizim hipotezini kurduğumuz iki değişken (sugar,calories) arasındaki ilişkiyi de bu grafikte açık bir şekilde görebiliriz.Bu grafikte şeker ve kalori arasında normal düzeyde pozitif korelasyon ilişkisi olduğunu ve bu nedenle Ho hipotezini reddedildiğini de bir kez daha söyleyebiliriz.

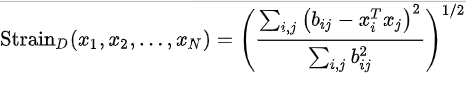
**PART B**

**MULTIDIMENSIONAL SCALING**

Çok boyutlu ölçekleme,veri setinin yüksek boyutlu uzayda temsil eden nesnelerin benzerlik veya farklılık yapılarını düşük boyutlu bir uzayda koruyarak görselleştirmek veya analiz etmek için kullanılan bir istatistiksel tekniktir.Bir veri setindeki nesnelerin arasındaki ilişkileri anlamak ve bu ilişkileri grafiksel temsil etmek amacıyla kullanılır.Benzerlik matrisi nesneler arasındaki benzerlik ölçülerini içerir.Mesafe matrisi ise nesneler arasındaki uzaklık veya farklılık ölçülerini içeren bir matristir.MDS’nin amacı,nesneler arası benzerlik veya farklılığı korurken nesneler arasındaki ilişkileri daha düşük boyutlu bir uzayda,genellikle iki veya üç boyutta temsil etmektir.

**FORMÜL**

MDS nin farklı varyasyonları bulunmasına rağmen, yaygın olarak kullanılan bir yöntem Klasik MDS veya Metrik MDS olarak adlandırılır. Klasik MDS nin formülü, nesneler arasındaki çiftler arası benzerlik veya farklılıkların hesaplanmasıyla başlar ve ardından gerilim fonksiyonunun optimize edilmesiyle devam eder. Gerilim fonksiyonu, orijinal yüksek boyutlu uzayda çiftler arası mesafeler ile düşük boyutlu uzaydaki mesafeler arasındaki tutarsızlığı ölçer. Optimizasyonun amacı, gerilim fonksiyonunu en aza indirerek nesnelerin düşük boyutlu uzaydaki en iyi temsillerini bulmaktır.



Klasik MDS formülü bu şekildedir.MDS’nin arkasındaki temel fikir veri noktalarının aralarındaki ilişkileri koruyacak şekilde görselleştirmektir.Pazar araştırmalarında, tüketicilerin ürünler veya markalar arasındaki tercihlerini anlamak için MDS kullanılabilir. Tüketici anketlerinden elde edilen veriler, MDS analiziyle düşük boyutlu bir uzayda temsil edilerek , ürünler veya markalar arasındaki benzerlik veya farklılık yapıları görselleştirilebilir.

**UYGULAMA**

Gerçek hayatta çok boyutlu ölçeklemenin uygulamalarına örnekler verelim.

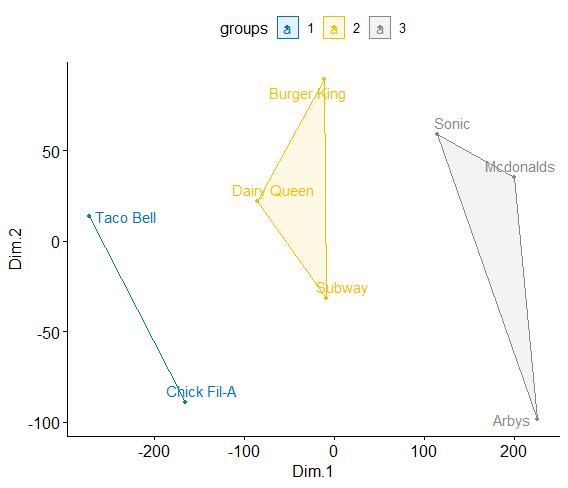
**Psikoloji:** Uyarıcılar veya nesneler arasındaki benzerliklerin algılanması gibi benzerlik değerlendirmelerinin analizi.

**Pazarlama:** Tüketici tercihlerini ve marka konumlandırmasını benzerlik veya farklılık verilerine dayanarak inceleme.

**Ekoloji:** Ekolojik özelliklere dayanarak türler arasındaki benzerliği araştırma.

**Sosyal bilimler:** Sosyal ağları analiz etme ve çeşitli özelliklere dayanarak bireyler veya gruplar arasındaki benzerliği inceleme.

Kendi veri setimize uyguladığımızda aşağıda görüldüğü gibi bir sonuç elde etmekteyiz.

****

Veri setimizde 8 adet fast food markasının menülerinin içerdiği kalori, kolesterol , kalsiyum , trans yağ , sodyum, şeker , protein ,a vitamini ve c vitamini bakımından burger king,Daily Queen ve Subway markalarının menülerinin içerikleri benzerdir.Sonic, Mcdonalds , Arby's markalarının menülerinin içerikleri birbirleriyle benzerdir.Taco Bell ve Chick Fil-A markalarının menülerinin içerikleri birbirleriyle benzerdir.

İlk olarak gerekli paketler yüklendi

magrittr: Veri manipülasyonu için kullanılan bir paket.

dplyr: Veri işleme ve dönüşümü için kullanılan bir paket.

ggpubr: ggplot2 paketini genişleten bir paket, daha gelişmiş grafiksel sunumlar için kullanılır.

**# Load required packages**

**library(magrittr)**

**library(dplyr)**

**library(ggpubr)**

**# Compute MDS**

**mds <- new\_data1 %>%**

**dist() %>%**

**cmdscale() %>%**

**as\_tibble()**

**colnames(mds) <- c("Dim.1", "Dim.2")**

**# Plot MDS**

**ggscatter(mds, x = "Dim.1", y = "Dim.2",**

**label = rownames(new\_data1),**

**size = 1,**

**repel = TRUE)**

**# K-means clustering**

**clust <- kmeans(mds, 3)$cluster %>%**

**as.factor()**

**mds <- mds %>%**

**mutate(groups = clust)**

**# Plot and color by groups**

**ggscatter(mds, x = "Dim.1", y = "Dim.2",**

**label = rownames(new\_data1),**

**color = "groups",**

**palette = "jco",**

**size = 1,**

**ellipse = TRUE,**

**ellipse.type = "convex",**

**repel = TRUE)**

**MDS Hesaplanması**

new\_data1 veri kümesi üzerindeki öklidyen uzaklık matrisi hesaplanır (**dist()** fonksiyonu).

**cmdscale()** fonksiyonu, MDS'yi hesaplar ve iki boyutlu bir uzayda konumları döndürür.

**as\_tibble()** fonksiyonu, MDS sonuçlarını tibble formatına dönüştürür ve mds değişkenine atar.

**colnames()** fonksiyonu, sütun adlarını "Dim.1" ve "Dim.2" olarak ayarlar.

**MDS Görselleştirme**

**ggscatter()** fonksiyonu, MDS sonuçlarını x ve y eksenlerine yerleştirir ve noktaları görselleştirir.

label parametresi, noktaların üzerindeki etiketleri gösterir.

size parametresi, noktaların boyutunu belirler.

repel parametresi, noktaların birbirlerini itmesini sağlar.

**K-means kümeleme**

**kmeans()** fonksiyonu, MDS sonuçlarını temel alarak k-means kümeleme algoritmasını uygular.mds veri setini ve küme sayısını belirten 3 parametresini alır.

$cluster özniteliği, her veri noktasının hangi kümeye ait olduğunu belirtir.

**as.factor()** fonksiyonu, küme numaralarını faktör veri tipine dönüştürür.

**mutate()** fonksiyonu, mds veri setine bir sütun ekler.

groups sütunu, k-means kümeleme sonuçlarını temsil eder ve clust değişkenine eşittir.

**Gruplara göre görselleştirme**

İkinci bir **ggscatter()** fonksiyonu çağrılır ve color parametresi, grupları belirlemek için groups sütununu kullanır.

palette parametresi, renk paletini belirler.

size parametresi,noktaların boyutunu belirler

ellipse parametresi, her gruba ait bir elipsin çizilmesini sağlar.

ellipse.type parametresi, elipsin şeklini belirler (konveks veya düzlemsel).

repel parametresi, etiketlerin birbirlerini itmesini sağlar.

**REFERANSLAR**

[**Multidimensional scaling - Wikipedia**](https://en.wikipedia.org/wiki/Multidimensional_scaling)

[**Fastfood Nutrition | Kaggle**](https://www.kaggle.com/datasets/ulrikthygepedersen/fastfood-nutrition)