HACETTEPE
ÜNİVERSİTESİ

Mayıs 2024

GRAFİKSEL VERİ ANALİZİ

ÖDEV 2: PASTA GRAFİĞİ

Dr. Öğr. Üyesi Onur TOKA

Eda Yaren ÖZEL
2200329007

1 PASTA GRAFİĞİ

1.1 Tanım

Pasta grafiği;

- Bir daireyi orantılı bölümlere bölerek, kategoriler arasında oranlar ve yüzdeler göstermeye yardımcı olur. Her bir yay uzunluğu her bir kategorinin bir oranını temsil ederken, tam daire % 100'e eşit olan tüm verilerin toplamını temsil eder[1].
- Adından da anlaşılacağı gibi bu görselleştirme, bütünü temsil etmek için bir daire ve bütünü oluşturan belirli kategorileri temsil etmek için bu dairenin dilimlerini veya "pastayı" kullanır[2].
- Bu grafik türü, kullanıcının belirli bir bağlamda farklı boyutlar (Örn. kategoriler, ürünler, kişiler, ülkeler vb.) arasındaki ilişkiyi karşılaştırmasına yardımcı olur[2].

1.2 Kullanım Alanları

- Birden fazla kategoriye göre performansı veya katkıyı göstermek için[3],
- Bazı verilerin bir bütünün kesirli bir parçası olarak görselleştirilerek temsil edilmesi gerektiğinde,
- Verileri karşılaştırmak ve birinin diğerinden neden daha küçük/büyük olduğunu göstermek için

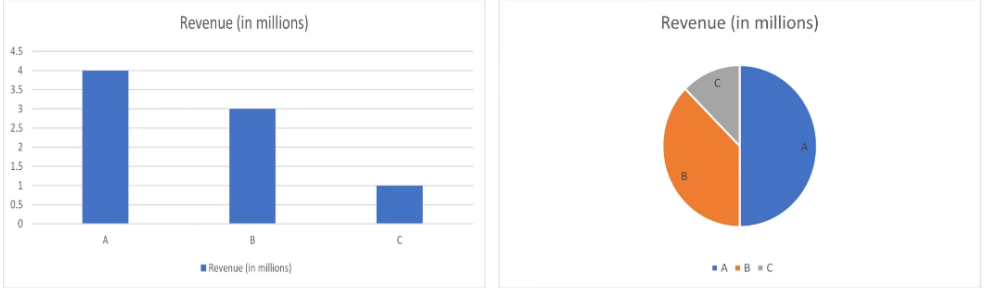
kullanılır. Bu nedenle, sınırlı sayıda grupla ve ayrık veri kümeleriyle uğraşıldığında pasta grafiğinin kullanılması daha avantajlıdır[4].

1.3 Avantajları/Dezavantajları

Avantajları;

- Pasta grafikleri, katkı yüzdesini iletmede çubuk veya sütun grafiklerine göre çok daha iyidir.

  Örnek: Şekil 1'deki sütun grafiğinden A Şehri'nin, toplam gelirin %50'sine katkıda bulunduğunu anlamak daha zordur. Fakat pasta grafiğinden bu kolavca anlasılabilir[3].



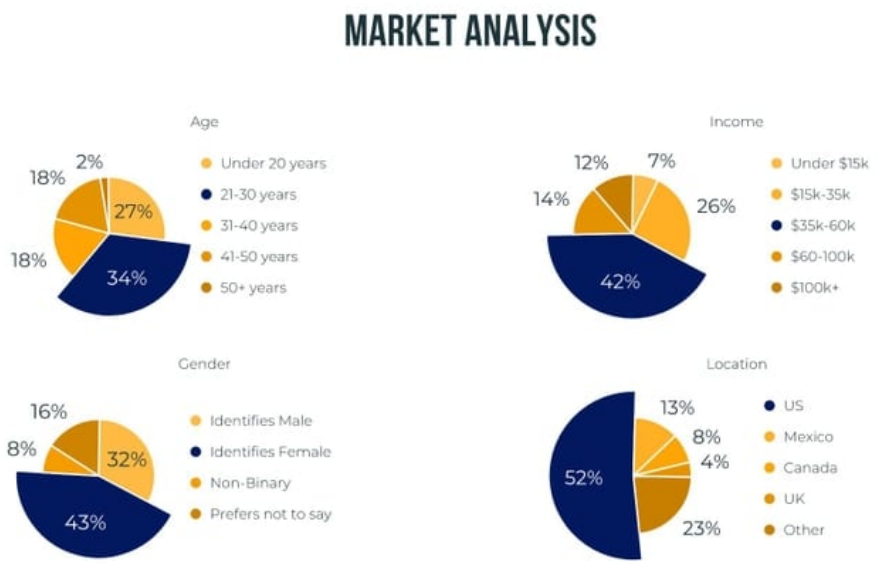
Şekil 1: Şehirlerin toplam gelire katkıları

- Diğer grafik türlerine göre pasta grafiği görsel olarak daha basit, anlaşılması kolay ve etkili bir iletişim aracı sağlar[4].
- Pasta grafiği, iş dünyasında ve medyada yaygın kullanım nedeniyle hızlı bir şekilde analiz yapmak veya bilgileri anlamak için hedef kitleye ilk bakışta veri karşılaştırmasına yardımcı olur[4].

Dezavantajları;

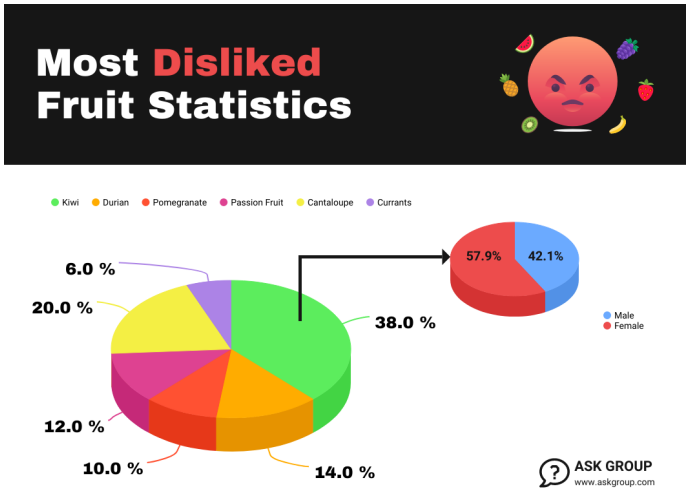
- Verilerdeki değişiklikleri göstermek için birçok pasta grafiğine ihtiyaç duyulabilir. Bu nedenle nedenleri, etkileri veya kalıpları açıklamakta başarısız olur[4].
- Pasta grafikleri, daha az kategorili veriler için kullanıldığında (ideal olarak 5'ten az veya buna eşit) daha etkili olur. Kategori sayısı arttıkça grafiğin okunabilirliği ve benzer tonlardaki (kontrasttaki) renkleri ayırt etmek de zorlaşır[3].

1.4 Örnek Grafikler



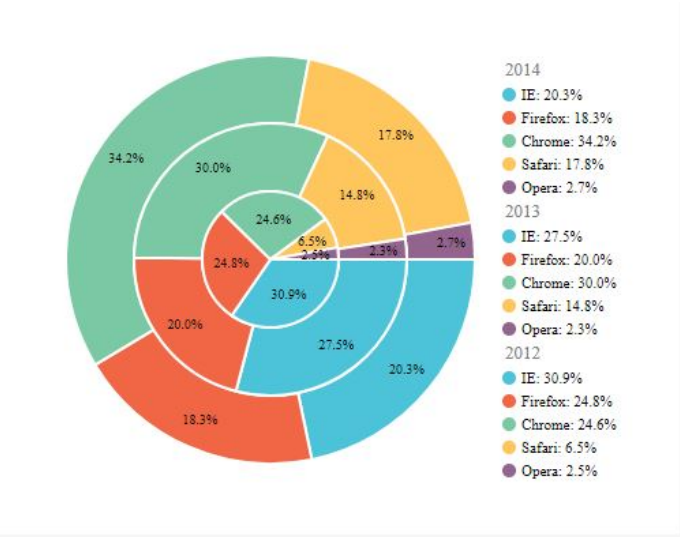
Şekil 2: Marketlerden alışveriş yapan müşteri profilleri[5]

- Şekil 2'deki örnekte marketlerden alışveriş yapan müşterilerin profillerini analiz etmek için dört farklı kategorik değişkenin pasta grafiği ile görselleştirilme yapılmıştır. Grafiklere göre 21-30 yaş arası,Birleşik Devletlerde yaşayan, 35-60 bin dolar gelire sahip kadınların market alışverişlerine daha çok gittiğini söyleyebiliriz.



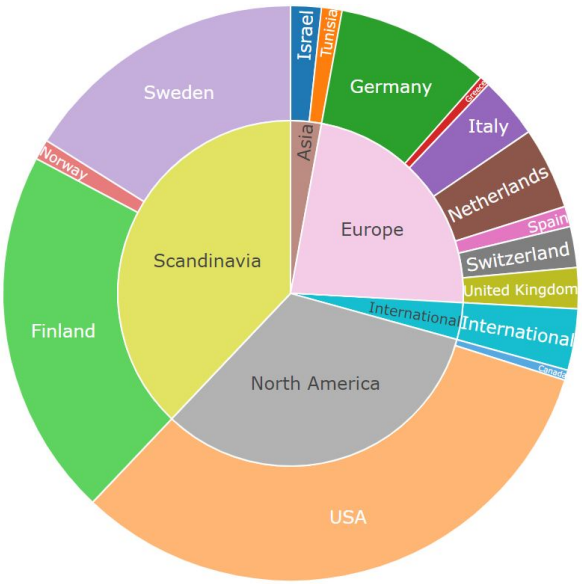
Şekil 3: En sevilmeyen meyve istatistikleri[6]

- Şekil 3'deki örnekte meyveler arasında en sevilmeyen meyvenin ve bu meyveye ait cinsiyetlerin pasta grafiği çizilerek görselleştirilme yapılmıştır. Grafiklere göre en sevilmeyen meyvenin kivi olduğunu ve en çok kadınlar tarafından sevilmediğini söyleyebiliriz.



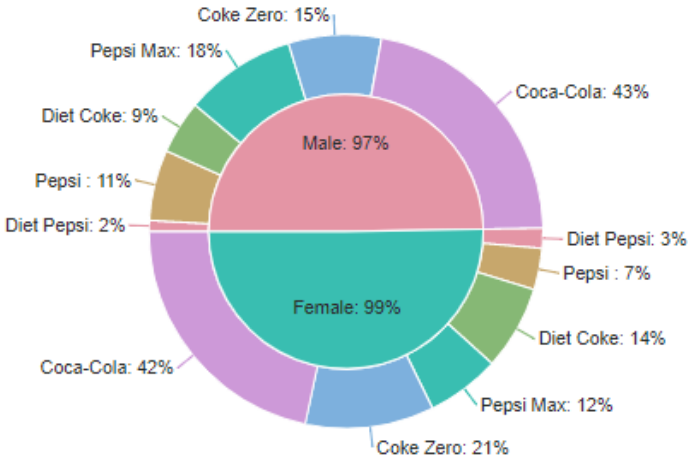
Şekil 4: 2012,2013,2014 yıllarındaki internet tarayıcısı kullanımları[7]

- Şekil 4'deki örnekte 2012, 2013 ve 2014 yıllarında internet tarayıcılarının kullanımları yıllara göre değişimin incelenebilmesi için iç içe pasta grafiği ile görselleştirilme yapılmıştır. Grafiklere göre 2012 yılında en çok kullanılan tarayıcı Internet Explorer olurken 2013 ve 2014 yıllarında en çok kullanılan tarayıcının Chrome olduğunu söyleyebiliriz. Grafik yıllara göre incelendiğinde Internet Explorer ve Firefox'un kullanımının her yıl azaldığını, Chrome ve Safari'nin kullanımının her yıl arttığını, her yıl en az kullanılan tarayıcının Opera olduğunu söyleyebiliriz.



Şekil 5: Ülkelerin bölgelere göre sınıflandırılması[8]

- Şekil 5'deki örnekte ülkeler Asya, Avrupa, İskandinavya, Kuzey Amerika ve Uluslararası sınıflara ayrılmak üzere pasta grafiği çizilerek görselleştirilme yapılmıştır. Grafiğe göre İsrail ve Tunus ülkelerinin Asya, Almanya-Yunanistan-İtalya-Hollanda-İspanya-İsviçre-İngiltere ülkelerinin Avrupa, Amerika ülkesinin Kuzey Amerika ve Finlandiya-Norveç-İsveç ülkelerinin de İskandinav ülkeleri olduğunu söyleyebiliriz.



Şekil 6: Cinsiyete göre kola markalarının tüketimi[9]

- Şekil 6'daki örnekte kola markalarının cinsiyete göre tüketimleri pasta grafiği çizilerek görselleştirme yapılmıştır. Grafiğe göre cinsiyet farketmeksizin en çok tüketilen kolanın markasının Coca Cola, en az tükettilen kolanın markasının Diet Pepsi olduğunu söyleyebiliriz.

2 VERİ

Ödev kapsamında kullanılan veri seti [10], 2020'den 2024'e kadar uzanan bir zaman diliminde, veri bilimi işlerine ilişkin Tablo 1'deki son maaş bilgilerini içermektedir.

Tablo 1: Verisetine İlişkin Değişkenler

Değişkenler	
İş Yılı	Deneyim Seviyesi
İstihdam Tipi	İş Unvanı
Maaş	Maaşın Para Birimi
Dolar Türünden Maaş	Çalışanın İkamet Yeri
İzin Verilen Uzaktan Çalışma Oranı	Şirketin Yeri
Şirketin Büyüklüğü	

3 R KODLARININ AÇIKLANMASI

```
# Kütüphanelerin indirilmesi:
#install.packages("ggplot2")
#install.packages("dplyr")
#install.packages("forcats")
#install.packages("gridExtra")
library(ggplot2)
library(dplyr)
library(forcats)
library(egg)
library(gridExtra)
```

Şekil 7: Veri seti için yüklenen R kütüphaneleri

- **install.packages():** Belirtilen paketlerin yüklenmesini sağlar.
- **library():** Yüklenen paketleri R ortamına getirir ve içindeki fonksiyonlara erişimi sağlar.
- **ggplot2:** Veri görselleştirme için kullanılan güçlü bir pakettir. Basitten karmaşığa kadar çeşitli grafik türlerini oluşturmanızı sağlar.
- **dplyr:** Veri işleme ve veri manipülasyonu için kullanılan bir pakettir. Veri çerçevelerinde sık kullanılan işlemleri (örneğin, filtreleme, sıralama, toplama) kolaylaştırır ve hızlandırır.
- **forcats:** Faktör (Kategorik) değişkenlerin işlenmesi için kullanılan bir pakettir. Kategorik değişkenlerin düzenlenmesi ve dönüştürülmesi için kullanışlı işlevler sağlar.

- **egg:** ggplot2 ile oluşturulan grafiklerin daha gelişmiş kontrolünü sağlayan bir pakettir. Özel grafik ayarlarını kolaylaştırır ve daha karmaşık grafik düzenlemeleri yapmanıza izin verir.
- **gridExtra:** Grafik nesnelerini bir araya getirmek ve düzenlemek için kullanılan bir pakettir. Birden fazla grafik nesnesini tek bir çerçevede birleştirmek veya yan yana yerleştirmek gibi işlemleri gerçekleştirmenizi sağlar.

```r
# Veri setinin yüklenmesi:
data <- read.csv("C:/Users/Yaren/Desktop/DataScience_salaries_2024.csv")
#View(data)

# Veri önişlemenin yapılması:
head(data)
summary(data)
colSums(is.na(data)) # Kayıp veri kontrolü
colnames(data)
```

Şekil 8: Veri setinin indirilmesi ve anlaşılması

- **<- :** R programlama dilinde değişken atama operatörü olarak kullanılır. Bu işaret, sağdaki değeri sol tarafındaki değişkene atar.
- **read.csv():** Belirtilen veri setinin ilk birkaç gözlemini görüntüler. Varsayılan olarak, head fonksiyonu veri setinin ilk 6 satırını gösterir, ancak isteğe bağlı olarak başka bir sayı belirtilebilir.
- **View():** Belirtilen veri setinin özet istatistiklerini görüntüler. Bu özet, her bir değişkenin sayısını, ortalamasını, medyanını, minimum ve maksimum değerlerini ve çeyrekliklerini içerir. Böylelikle, veri setinin genel yapısı hakkında hızlı bir genel bakış elde etmiş oluruz.
- **is.na():** Belirtilen veri setindeki eksik değerlerin bulunduğu hücreleri TRUE, eksik olmayanları FALSE yapar.
- **colSums():**Belirtilen veri setindeki her bir sütunda (değişkende) eksik değerlerin sayısını (toplamını) hesaplar.
- **colnames():** Belirtilen veri setinin sütun (değişken) isimlerini görüntüler. Bu kod, veri setinde hangi değişkenlerin bulunduğunu görmek için kullanılır.

```r
## Kullanılacak nicel değişkenin aralıklara ayrılarak kategorikleştirilmesi:
df <- data.frame(year = data$work_year,
                 salary = as.character(cut(data$salary_in_usd,
                                           breaks = 3,
                                           labels = c("Low", "Mid", "High"))))
```

Şekil 9: Kullanılacak nicel değişkenin kategorilere ayrılması

- **data.frame():** Yeni bir veri çerçevesi oluşturmak için kullanılır. Veri çerçevesi, farklı türde değişkenleri bir araya getiren bir veri yapısıdır. Veri çerçeveleri genellikle tablo benzeri veri yapılarıdır, her bir sütun bir değişkeni temsil eder.

- **$:** R'da bir veri çerçevesindeki veya listedeki bir sütuna erişmek için kullanılır.
- **as.character():** Bir nesneyi karakter dizisi türüne dönüştürmek için kullanılır.
- **cut():** Belirli bir sayı dizisini kesme noktalarına göre kategorik değişkenlere dönüştürmek için kullanılır.
- **breaks:** cut() fonksiyonunda kullanılan bir parametredir. Sayı dizisinin bölme noktalarını belirtir.
- **labels:** cut() fonksiyonunda kullanılan bir parametredir. Oluşan kategorik değişkenin etiketlerini belirtir.
- **c():** belirtilen elemanları birleştirmek için kullanılır.

```
> df
   year salary
1  2021    Low
2  2021    Low
3  2020    Low
4  2021    Low
5  2022    Low
6  2021    Low
7  2021    Low
8  2022    Low
9  2022    Low
10 2022    Low
```

Şekil 10: df veri kümesi

Kodlar sonucunda, veri kümesinde görselleştirme için kullanılacak değişkenler çalışma yılı ve maaş olmak üzere belirlenmiş olup, nicel değişken olan 'salary' değişkeni 3 aralığa bölünerek düşük, orta ve yüksek maaş olmak üzere kategorikleştirilmiştir.

```
## Maaş değişkeninin kategorilerine göre yılları gruplayarak toplam maaşları
## hesaplama:
df_summary <- data %>%
  mutate(salary_category = cut(salary_in_usd,
                               breaks = 3,
                               labels = c("Low", "Mid", "High"))) %>%
  group_by(work_year, salary_category) %>%
  summarise(count = n()) %>%
  ungroup()
```

Şekil 11: Maaş kategorilerine göre çalışma yılları gruplanarak toplam maaşların hesaplanması

- %>%**:** Pipe operatörü, bir işlemin çıktısını bir sonraki işleme giriş olarak aktarır. Bu şekilde, kod satırları birbirine bağlanır ve işlem zinciri oluşturulur.

- **mutate():** Belirtilen veri çerçevesinde yeni değişkenler oluşturmak veya mevcut değişkenleri dönüştürmek için kullanılır.
- **group_by():** Belirtilen veri çerçevesini belirli bir veya birden fazla değişkene göre gruplamak için kullanılır.
- **summarise():** Gruplanmış veri çerçevesini özetlemek için kullanılır.
- **n():** Gruplanmış bir veri çerçevesindeki her grubun gözlem sayısını hesaplamak için kullanılır.
- **ungroup():** Bir veri çerçevesini gruplama işleminden çıkarmak için kullanılır. Bu işlem, daha sonra gruplanmış bir veri çerçevesini düzleştirmek veya başka bir işlem yapmak için gereklidir.

```
> df_summary
# A tibble: 12 x 3
   work_year salary_category count
       <int> <fct>           <int>
 1      2020 Low                71
 2      2020 Mid                 4
 3      2021 Low               216
 4      2021 Mid                 2
 5      2022 Low              1633
 6      2022 Mid                19
 7      2023 Low              8158
 8      2023 Mid               355
 9      2023 High                6
10      2024 Low              4143
11      2024 Mid               212
12      2024 High               19
```

Şekil 12: df_summary veri kümesi

Kodlar sonucunda, maaş kategorileri çalışma yıllarına göre gruplandırılarak her grubun gözlem sayısını 'count' isminde bir değişkene yazıp df_summary adında yeni bir veri çerçevesi oluşturulmuştur.

```
## Her bir yıl için yüzdelik değerleri hesaplama:
df_summary <- df_summary %>%
  group_by(work_year) %>%
  mutate(percentage = count / sum(count) * 100) %>%
  ungroup()
```

Şekil 13: df_summary veri kümesine yüzdelik değerlerin eklenmesi

```
> df_summary
# A tibble: 12 x 4
   work_year salary_category count percentage
       <int> <fct>           <int>      <dbl>
 1      2020 Low                71    94.7
 2      2020 Mid                 4     5.33
 3      2021 Low               216    99.1
 4      2021 Mid                 2     0.917
 5      2022 Low              1633    98.8
 6      2022 Mid                19     1.15
 7      2023 Low              8158    95.8
 8      2023 Mid               355     4.17
 9      2023 High                6     0.0704
10      2024 Low              4143    94.7
11      2024 Mid               212     4.85
12      2024 High               19     0.434
```

Şekil 14: df_summary veri kümesi

Kodlar sonucunda, df_summary veri kümesinde bulunan her grup için 'count' değişkeni kullanılarak yüzdelik değerler hesaplanıp df_summary veri kümesine 'percentage' isimli bir değişkene yazılmıştır.

```
## Yeni veri seti oluşturma:
df <- data.frame(year = df_summary$work_year,
                 salary_category = df_summary$salary_category,
                 percentage = df_summary$percentage)
```

Şekil 15: yeni veri seti oluşturma

```
> df
    year salary
1   2021    Low
2   2021    Low
3   2020    Low
4   2021    Low
5   2022    Low
6   2021    Low
7   2021    Low
8   2022    Low
9   2022    Low
10  2022    Low
```

Şekil 16: yeni df veri kümesi

Kodlar sonucunda, df_summary veri kümesinde bulunan çalışma yılı, maaş kategorileri ve yüzdelik değerler alınarak df veri kümesine yazılmıştır.

```
## Yüzdelik değerleri "Y" olarak atama:
df$c <- df$percentage
df$c <- sprintf("%.2f", df$c)

df <- df[-(1:4), ]

df <- df %>%
  select(-percentage)
```

Şekil 17: df veri kümesini düzenleme

- **sprintf():** Bir karakter dizisini biçimlendirmek için kullanılır. Genellikle, belirli bir format kullanarak sayıları veya diğer veri tiplerini karakter dizilerine dönüştürmek için kullanılır.
- **%2.f:** Bu bir format dizesidir ve sprintf() fonksiyonunda kullanılır. %f, ondalık sayıları formatlamak için kullanılan bir belirteçtir ve %.2f, ondalık sayıları noktadan sonra iki basamakla biçimlendirmek için kullanılır.
- **-:** R'de vektörlerden, veri çerçevelerinden veya listelerden belirli elemanları çıkarmak veya belirli sıralardaki elemanları silmek için kullanılır.
- **select():** Veri çerçevesinden belirli sütunları veya değişkenleri seçmek veya belirli sütunları hariç tutmak için kullanılır.

```
> df
   year salary_category     c
5  2022             Low 98.85
6  2022             Mid  1.15
7  2023             Low 95.76
8  2023             Mid  4.17
9  2023            High  0.07
10 2024             Low 94.72
11 2024             Mid  4.85
12 2024            High  0.43
```

Şekil 18: düzenlenmiş df veri kümesi

Kodlar sonucunda, df veri kümesinde bulunan yüzdelik değerler 'c' isimli değişken altına noktadan sonraki iki basamağı alınacak şekilde yazılmıştır. Görselleştirme son üç yılı kapsayacak şekilde yapılacağı için veri kümesinde ilk dört satıra denk gelen 2020 ve 2021 yıllarına ait değerler ve 'percentage' sütunu veri kümesinden çıkarılmıştır.

```
new_row <- data.frame(year = 2022, salary_category = "High", c = 0.00)
df <- rbind(df, new_row)

df$salary_category <- as.factor(df$salary_category)

df <- df %>%
  arrange(desc(year))
```

Şekil 19: df veri kümesini düzenleme

```
> df
  year salary_category     c
1 2024             Low 94.72
2 2024             Mid  4.85
3 2024            High  0.43
4 2023             Low 95.76
5 2023             Mid  4.17
6 2023            High  0.07
7 2022             Low 98.85
8 2022             Mid  1.15
9 2022            High     0
```

Şekil 20: düzenlenmiş df veri kümesi

- **rbind():** R'de veri çerçeveleri veya matrisler arasında satır bazında birleştirme yapmak için kullanılır. Yani, yeni bir satır eklemek için kullanılır.
- **as.factor():** Bir değişkenin faktör veri tipine dönüştürülmesini sağlar. Faktörler, belirli bir kategori setinden oluşan ve genellikle kategorik değişkenler olarak kullanılan veri tipidir.
- **arrange():** Belirtilen veri çerçevesindeki satırları belirli bir değişkene veya değişkenlere göre sıralamak için kullanılır.
- **desc():** Sıralama işlemi için kullanılan bir dönüşümdür. arrange() fonksiyonu ile birlikte kullanıldığında, belirli bir değişkeni azalan sırayla sıralamak için kullanılır.

Kodlar sonucunda, veri setinde 2022 çalışma yılının 'High' maaş kategorisine ait yüzde değeri yer almadığı için df veri kümesine yeni bir satır olarak eklenmiştir. Maaş kategorisi değişkeni kategorikleştirilip df veri kümesi yıllara göre azalan sırada sıralanmıştır.

```
# Grafik çizimi:
df$Y <- as.numeric(as.character(df$c))
foo <- data.frame(cumsum(table(df$year)) + 1:length(unique(df$year)))
foo$year <- rownames(foo)
colnames(foo)[1] <- "row"
```

Şekil 21: foo veri çerçevesinin oluşturulması

- **as.numeric():** Bir nesneyi sayısal bir vektöre dönüştürmek için kullanılır.
- **cumsum():** Bir vektörün kümülatif toplamını hesaplamak için kullanılır. Yani, her bir elemanı kendisinden önceki tüm elemanların toplamıyla değiştirir.
- **table():** Bir faktör veya vektördeki değerlerin frekans tablosunu oluşturmak için kullanılır. Yani, her bir farklı değerin kaç kez tekrarlandığını sayar.
- **length():** Bir nesnenin uzunluğunu veya boyutunu belirler. Vektörün uzunluğunu, matrisin satır veya sütun sayısını veya bir karakter dizisinin uzunluğunu belirlemek için kullanılabilir.
- **unique():** Bir vektördeki veya faktördeki benzersiz değerleri döndürmek için kullanılır. Yani, tekrarlanan değerleri bir kez gösterir.
- **rownames():** bir veri çerçevesinin veya matrisin satır adlarını (veya satır indekslerini) döndürmek için kullanılır.

```
> foo
     row year
2022   4 2022
2023   8 2023
2024  12 2024
```

Şekil 22: foo veri kümesi

```
> df
  year salary_category     c     Y
1 2024             Low 94.72 94.72
2 2024             Mid  4.85  4.85
3 2024            High  0.43  0.43
4 2023             Low 95.76 95.76
5 2023             Mid  4.17  4.17
6 2023            High  0.07  0.07
7 2022             Low 98.85 98.85
8 2022             Mid  1.15  1.15
9 2022            High     0  0.00
```

Şekil 23: düzenlenmiş df veri kümesi

Kodlar sonucunda, df veri çerçevesi içindeki 'c' olarak atanan yüzdelik değerleri sayısal bir vektöre dönüştürülerek 'Y' olarak isimlendirilip yeniden df veri çerçevesine atanmıştır. Ardından görselleştirilmede kullanılmak amacıyla yıl değerlerinin frekans tablosu oluşturularak foo isimli yeni bir veri çerçevesine atanmıştır. Son olarak, foo veri çerçevesine 'row' isimli bir sıra numarası sütunu eklenerek satır isimleri foo veri çerçevesine atanmıştır.

```r
## Maaş kategorileri için satırların atanması:
df$row <- which(do.call(rbind, by(df, df$year, rbind, ""))$year != "")
## Renklerin belirlenmesi:
colors <- c("#ee799f", "#9370db", "#32cd32" )
```

Şekil 24: Oluşturulacak grafik için renklerin belirlenmesi

- **which():** Belirli bir koşulu sağlayan elemanların indislerini döndürür. Yani, koşulu sağlayan elemanların indekslerini bulmak için kullanılır.
- **do.call():** Bir fonksiyon ve bir listeyi veya bir vektörü argüman olarak alır ve bu fonksiyonu bu argümanlarla çağırır. Yani, bir fonksiyonu bir listeye veya vektöre uygulamak için kullanılır.
- **by():** Bir veri çerçevesini belirli bir faktöre veya faktör kombinasyonuna göre parçalara böler ve her parçayı belirli bir işlevle işler. Yani, bir veri çerçevesini gruplamak ve her grup için bir işlev uygulamak için kullanılır.
- **!= :** İki değerin birbirine eşit olmadığını kontrol eder.

```
> df
  year salary_category     c     Y row
1 2024             Low 94.72 94.72   1
2 2024             Mid  4.85  4.85   2
3 2024            High  0.43  0.43   3
4 2023             Low 95.76 95.76   5
5 2023             Mid  4.17  4.17   6
6 2023            High  0.07  0.07   7
7 2022             Low 98.85 98.85   9
8 2022             Mid  1.15  1.15  10
9 2022            High     0  0.00  11
```

Şekil 25: düzenlenmiş df veri kümesi

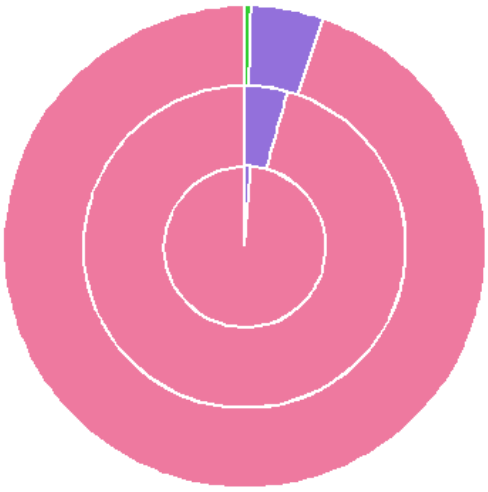
Kodlar sonucunda, oluşturulacak grafikte kullanılacak renkler vektörü oluşturulup df veri çerçevesindeki her yıl için satırlar gruplanarak satır indeksleri bulunmuştur. Ardından, bu satır indeksleri kullanılarak her yılın başlangıç satırları belirlenerek df veri çerçevesine 'row' isimli yeni bir sütun değişkenine atanmıştır.

```r
### İlk grafiğin çizimi:
p1 <- ggplot(df, aes(factor(year), Y, fill = salary_category)) +
  geom_bar(stat = "identity", width = 1, size = 1, color = "white") +
  coord_polar("y") +
  theme_void() +
  theme(legend.position = "none") +
  scale_fill_manual(values = colors)
```

Şekil 26: Birinci grafiğin oluşturulması

- **ggplot():** Bir ggplot2 grafiği oluşturmak için kullanılır. Temel olarak, veri setini ve estetik özellikleri (x ekseni, y ekseni, renk vb.) belirler.

- **aes():** Grafiğin estetik (aes) özelliklerini belirtmek için kullanılır. Yani, grafikte kullanılacak veri değişkenlerini ve bu değişkenlerin hangi görsel özelliklere (x ekseni, y ekseni, renk vb.) atandığını belirtir.
- **fill:** aes() içinde kullanılan bir parametre, çubuk grafikteki dolgu rengini belirler. fill parametresine bir değişken atanırsa, her kategoriye farklı bir dolgu rengi atanır.
- **geom_bar():** Çubuk grafik oluşturmak için kullanılır.
- **stat:** geom_bar() içinde kullanılan bir parametre, istatistiksel hesaplamaların nasıl yapılacağını belirler.
- **size:** geom_bar() içinde kullanılan bir parametre, çubukların kalınlığını belirler.
- **coord_polar():** Kutupsal koordinat sistemi kullanarak çubuk grafiklerin dairesel olarak gösterilmesini sağlar.
- **theme():** Grafiğin görünümünü ayarlamak için kullanılır. Örneğin, arka plan rengi, eksen çizgileri, başlık vb. özellikler belirlenebilir.
- **legend.position:** theme() içinde kullanılan bir parametre, grafiğin nerede bir açıklama kutusu (legend) içereceğini belirler.
- **scale_fill_manual():** Özel renk paletlerini belirlemek için kullanılır. **values** parametresi, farklı kategori veya faktör seviyelerine atanacak renkleri içerir.



Şekil 27: Birinci grafik

Kodlar sonucunda;

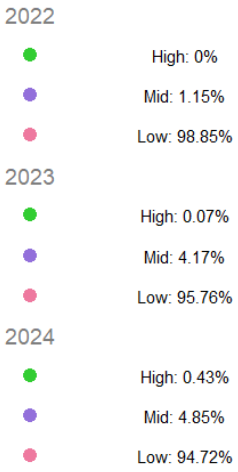
- df veri çerçevesindeki 'year' değişkeni grafiğin x ekseninde, yüzdelik değerleri içeren 'Y' değişkeni ise y ekseninde yer alacak şekilde,
- grafikte kullanılacak dolgu rengi maaş kategorileri olacak şekilde,
- grafikte kullanılacak her bir barın veri setindeki değerlerle aynı genişlikte olacağı şekilde,
- çubuk grafikleri dairesel olacağı şekilde,
- grafiğin arka planı temizlenecek ve gereksiz unsurlar gizlenecek şekilde,
- maaş kategorilerine özel renkler atanacak şekilde

df veri çerçevesindeki maaş kategorilerinin son üç çalışma yılına göre dağılımını gösteren bir pasta grafiği oluşturularak p1 olarak atanmıştır.

```
### İkinci grafiğin çizimi:
p2 <- ggplot(df, aes(y = row)) +
  geom_point(aes(0, color = salary_category), size = 4) +
  geom_text(data = foo, aes(0, label = rev(year)), size = 5, color = "grey50") +
  geom_text(aes(0.5, label = paste0(salary_category, ": ", c, "%"))) +
  theme_void() +
  theme(legend.position = "none") +
  scale_x_discrete() +
  scale_color_manual(values = colors)
```

Şekil 28: İkinci grafiğin oluşturulması

- **geom_point():** Nokta geometrisi eklemek için kullanılır. aes() içinde belirtilen estetik özelliklerine (x ekseni, y ekseni, renk vb.) göre noktalar çizilir.
- **geom_text():** Metin etiketleri eklemek için kullanılır. aes() içinde belirtilen estetik özelliklerine (x ekseni, y ekseni, renk vb.) göre metin etiketleri yerleştirilir.
- **rev():** Bir vektörün elemanlarını tersine çevirir. Yani, verilen vektörün elemanlarını sondan başa doğru sıralar.
- **theme_void():** Grafiği temizlemek için kullanılır. Yani, arka plan rengini kaldırır ve eksen çizgilerini ve etiketlerini gizler.
- **scale_x_discrete():** X ekseninin ölçeklendirilmesini ayarlamak için kullanılır. Bu özellik, x ekseninin bir kategorik değişken olduğu durumlarda kullanılır.
- **scale_color_manual():** Özel renk paletlerini belirlemek için kullanılır. values parametresi, belirli kategori veya faktör seviyelerine atanacak renkleri içerir.



Şekil 29: İkinci grafik

Kodlar sonucunda,

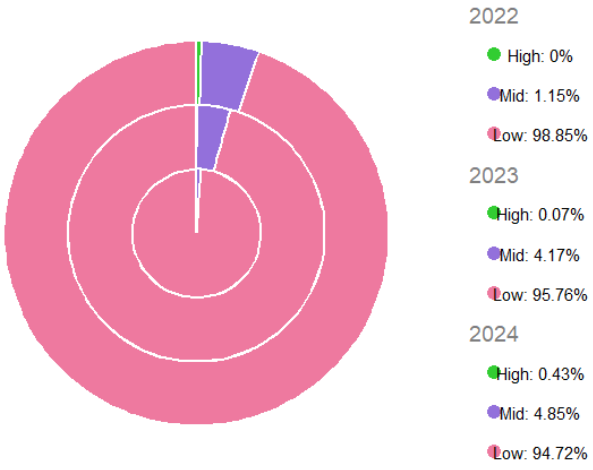
- df veri çerçevesindeki 'row' değişkeni grafiğin y ekseni üzerinde yer alacak şekilde,
- Grafikte noktaların renkleri maaş kategorilerine göre belirlenecek şekilde,
- foo veri çerçevesinden alınan yıl bilgileri, gri renkte ve büyük boyutta grafiğe eklenecek şekilde,
- Maaş kategorileri ve yüzdelik değerleri, grafiğin ortasında yer alacak şekilde,
- Grafiğin arka planı temizlenmiş ve gereksiz unsurlar gizlenecek şekilde,
- Maaş kategorilerine özel renkler atanacak şekilde

df veri çerçevesindeki maaş kategorilerinin son üç çalışma yılına göre nokta grafiği oluşturulmuş ve p2 olarak atanmıştır.

```
# Grafiklerin birleştirilmesi:
ggarrange(p1, p2, nrow = 1, widths = c(3, 1))
```

Şekil 30: İki grafiğin birleştirilmesi

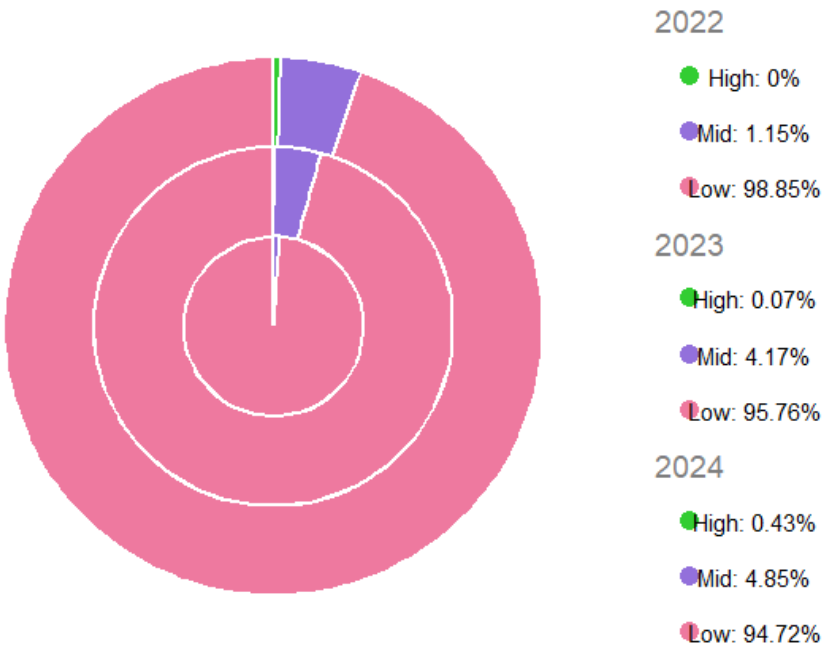
- **ggarrange():** ggplot2 grafiği nesnelerini düzenlemek için kullanılır. Birden fazla grafiği yan yana veya alt alta düzenlemek için kullanılır.
- **widths:** Her bir grafik nesnesinin genişliğini belirlemek için kullanılır. ggarrange() içindeki grafiklerin genişliklerini orantılı olarak belirlemek için kullanılır.



Şekil 31: Sonuç grafiği

Kodlar sonucunda, p1 ve p2 olarak atanan iki grafik genişlikleri sırasıyla 3 birim ve 1 birim olacak şekilde birleştirilerek sonuç grafiği oluşturulmuştur.

4 SONUÇLAR



Şekil 32: Sonuç grafiği

Oluşturulan grafik incelendiğinde;

- Veri bilimci maaşlarının 2022 yılında en çok düşük maaş kategorisinde yer aldığını ve 2022 yılında yüksek maaş alan hiç veri bilimci bulunmadığını,

- Veri bilimci maaşlarının 2023 yılında en çok düşük maaş kategorisinde yer aldığını fakat 2022 yılına oranla bir kaç düşük maaşlı veri bilimcinin ya orta maaş ya da yüksek maaş alarak maaşlarının yükseldiğini,
- Veri bilimci maaşlarının yıllar geçtikçe az da olsa yükseldiğini fakat 2024 yılında veri bilimcilerin %94'ünün hala düşük maaş aldığını ve yüksek maaş alan veri bilimcilerin sayısının az da olsa arttığını

söyleyebiliriz.

Kaynaklar

[1] https://datavizcatalogue.com/TR/yontemleri/pasta_grafik.html[Accessed: 25.04.2024].

[2] https://www.tableau.com/data-insights/reference-library/visual-analytics/charts/pie-charts [Accessed: (25.04.2024)].

[3] https://inforiver.com/insights/pie-chart-101-how-to-use-when-to-avoid-them/ [Accessed: (25.04.2024)].

[4] https://www.cuemath.com/data/pie-charts/ [Accessed: (25.04.2024)].

[5] https://www.beautiful.ai/templates/pie-chart [Accessed: (25.04.2024)].

[6] https://venngage.com/templates/charts/pie-pie-chart-512ce41d-b730-42e0-b32f-45ec2ddb7384 [Accessed: (25.04.2024)].

[7] https://stackoverflow.com/questions/48588312/labelled-multi-level-pie-chart. [Accessed: (25.04.2024)].

[8] https://brunofuga.adv.br/?s=plotly-how-to-do-nested-pie-chart-in-r-cc-EJon0qhG. [Accessed: (25.04.2024)].

[9] https://help.displayr.com/hc/en-us/articles/360003170836-How-to-Create-a-Grouped-or-Clustered-Pie-Chart. [Accessed: (25.04.2024)].

[10] https://www.kaggle.com/datasets/saurabhbadole/latest-data-science-job-salaries-2024?resource=download. [Accessed: (29.04.2024)].