R ile İstatistiksel Programlama Final Ödevi

Yusuf KAMAY - 20201101084

İstanbul'a Verilen Temiz Su Miktarları Verisi

1. VERİ SETİNİ YÜKLEME

```
library(readxl)
ist <- read_excel("D:/ist.xlsx")
View(ist)
```

2. GEREKLİ PAKETLERİN YÜKLENMESİ

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.2     ✓ readr     2.1.4
✓ forcats   1.0.0     ✓ stringr   1.5.0
✓ ggplot2   3.4.2     ✓ tibble    3.2.1
✓ lubridate 1.9.2     ✓ tidyr     1.3.0
✓ purrr     1.0.1
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become err
```

```
library(magrittr)
```

```
Attaching package: 'magrittr'

The following object is masked from 'package:purrr':

    set_names

The following object is masked from 'package:tidyr':

    extract
```

```
library(plotly)
```

```
Attaching package: 'plotly'

The following object is masked from 'package:ggplot2':

    last_plot

The following object is masked from 'package:stats':

    filter
```

```
The following object is masked from 'package:graphics':

    layout
```

```
library(readxl)
library(ggplot2)
library(dplyr)
library(plotly)
library(tidyr)
library(naniar)
library(zoo)
```

```
Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```
library(DataExplorer)
```

3. VERİ ANALİZİ

Veri seti boyutunu inceleyelim;

```
dim(ist)
```

```
[1] 12 15
```

Veri setindeki her satır bir ay değerini göstermektedir ve 12 satırdan oluşmaktadır.

14 yılın verileri incelendiğinden 14 + 1 (ilk sütun string aylar) 15 sütuna sahiptir.

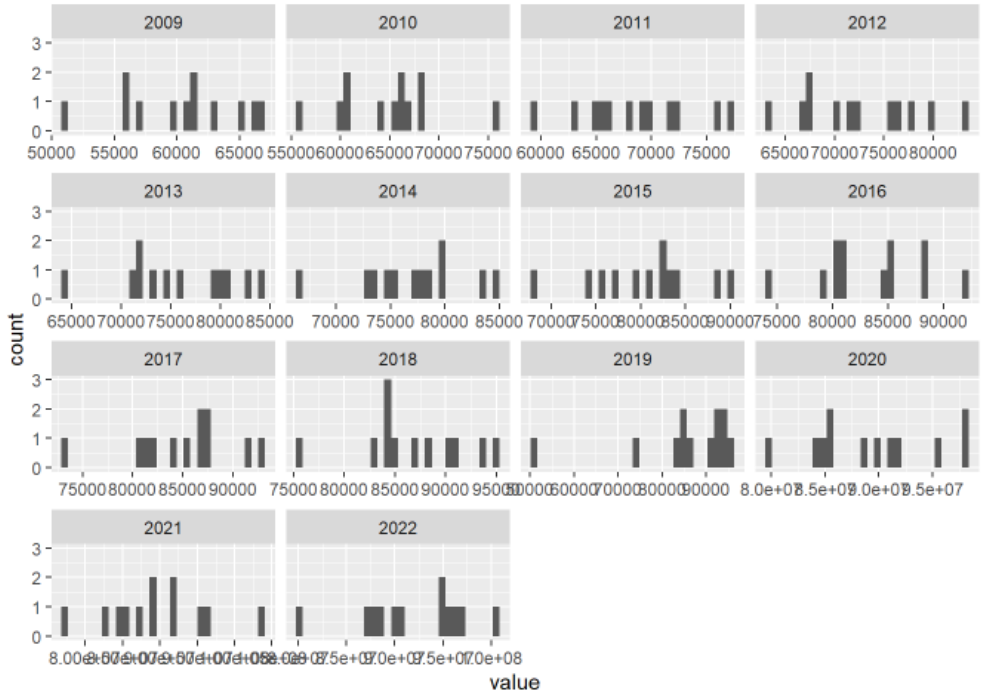
Veri yapısını incelemek için;

```
str(ist)
```

```
tibble [12 × 15] (S3: tbl_df/tbl/data.frame)
 $ Ay  : chr [1:12] "Ocak" "Şubat" "Mart" "Nisan" ...
 $ 2009: num [1:12] 55926 50838 57261 56034 62878 ...
 $ 2010: num [1:12] 60626 55607 60690 60060 68131 ...
 $ 2011: num [1:12] 65178 59159 65895 63179 69283 ...
 $ 2012: num [1:12] 67137 63613 67727 66769 71704 ...
 $ 2013: num [1:12] 71538 64379 71663 70859 80496 ...
 $ 2014: num [1:12] 74528 66844 73786 72909 79584 ...
 $ 2015: num [1:12] 77053 67912 75459 74177 82905 ...
 $ 2016: num [1:12] 80206 74114 79147 80340 84951 ...
 $ 2017: num [1:12] 81757 73515 81215 80663 87441 ...
 $ 2018: num [1:12] 84212 75692 84274 84239 90369 ...
 $ 2019: num [1:12] 83576 74797 84666 84968 93692 ...
 $ 2020: num [1:12] 84977984 80120814 88457622 84337847 90040181 ...
 $ 2021: num [1:12] 84160554 77068001 85420000 82501000 91395884 ...
 $ 2022: num [1:12] 88812431 80043139 87987877 87155031 95320559 ...
```

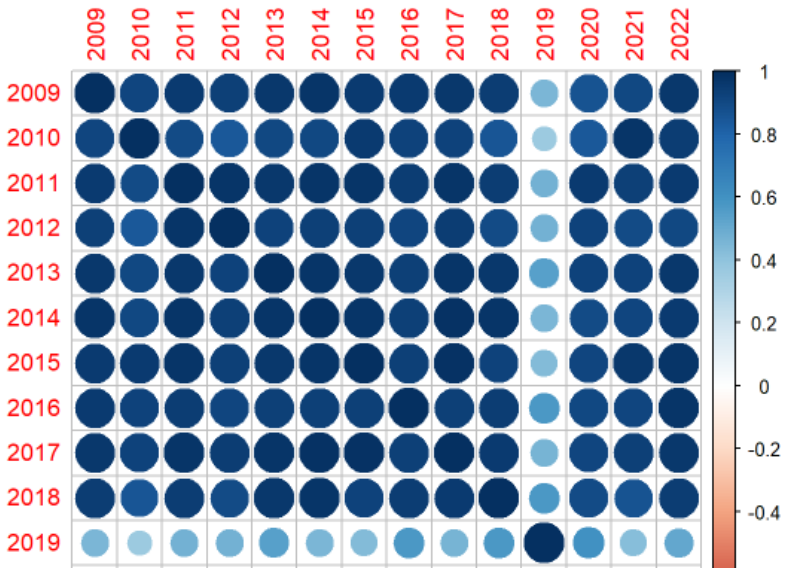
Her bir değişken için dağılım grafikleri;

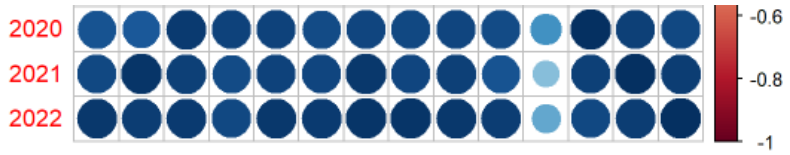
```
ist %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
  geom_histogram(bins = 30) +
  facet_wrap(~key, scales = 'free_x')
```



Değişkenler arasındaki ilişkileri incelemek için;

```
ist %>%
  select_if(is.numeric) %>%
  cor() %>%
  corrplot::corrplot()
```






```
summary(ist)
```

```
     Ay                 2009            2010            2011      
 Length:12          Min.   :50838   Min.   :55607   Min.   :59159  
 Class :character   1st Qu.:56954   1st Qu.:60674   1st Qu.:65642  
 Mode  :character   Median :60861   Median :65882   Median :68783  
                    Mean   :60305   Mean   :64836   Mean   :68646  
                    3rd Qu.:63461   3rd Qu.:67083   3rd Qu.:71961  
                    Max.   :66583   Max.   :75594   Max.   :76919  
     2012            2013            2014            2015      
 Min.   :63613   Min.   :64379   Min.   :66844   Min.   :67912  
 1st Qu.:67580   1st Qu.:71632   1st Qu.:74343   1st Qu.:76655  
 Median :71834   Median :75233   Median :77526   Median :81698  
 Mean   :72745   Mean   :75788   Mean   :77037   Mean   :80430  
 3rd Qu.:76674   3rd Qu.:80246   3rd Qu.:79693   3rd Qu.:83200  
 Max.   :83574   Max.   :84224   Max.   :84905   Max.   :89859  
     2016            2017            2018            2019      
 Min.   :74114   Min.   :73515   Min.   :75692   Min.   :51278  
 1st Qu.:80307   1st Qu.:81622   1st Qu.:84232   1st Qu.:84394  
 Median :82863   Median :86087   Median :85796   Median :88593  
 Mean   :83218   Mean   :85054   Mean   :86747   Mean   :85457  
 3rd Qu.:86099   3rd Qu.:87488   3rd Qu.:90485   3rd Qu.:93274  
 Max.   :91773   Max.   :93177   Max.   :95090   Max.   :96028  
     2020               2021               2022          
 Min.   :80120814   Min.   : 77068001   Min.   : 80043139  
 1st Qu.:85332469   1st Qu.: 85105138   1st Qu.: 88606292  
 Median :89248902   Median : 89490353   Median : 93025068  
 Mean   :89511165   Mean   : 89499225   Mean   : 91972672  
 3rd Qu.:92905343   3rd Qu.: 92722382   3rd Qu.: 95505472  
 Max.   :98344547   Max.   :103430618   Max.   :100430578  
```

```
head(ist,7)
```

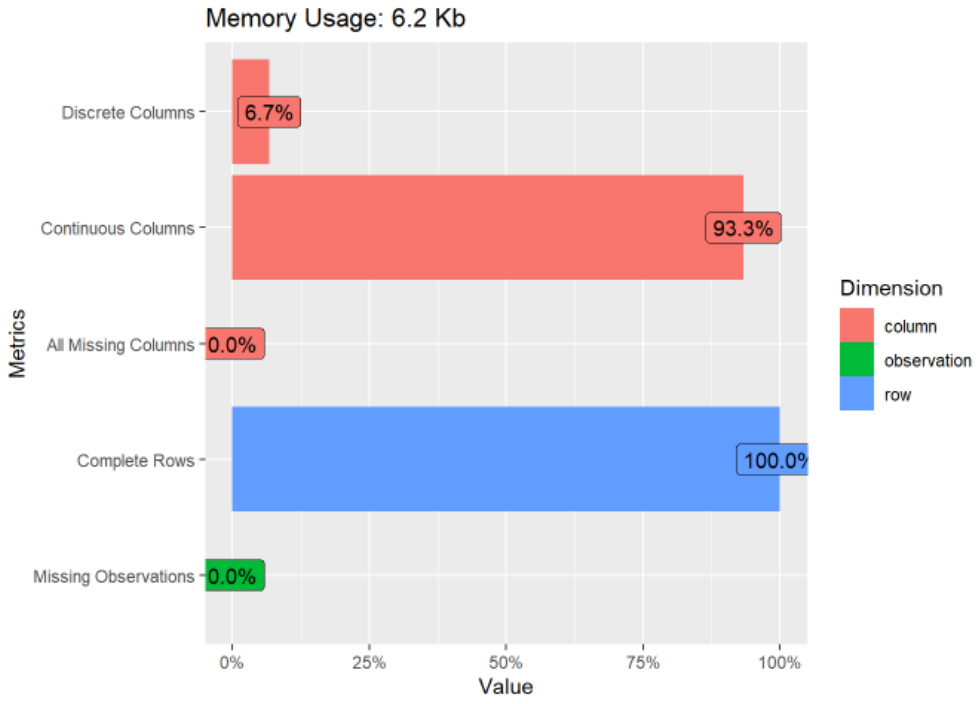
```
# A tibble: 7 × 15
  Ay      `2009` `2010` `2011` `2012` `2013` `2014` `2015` `2016` `2017` `2018`
  <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1 Ocak     55926  60626  65178  67137  71538  74528  77053  80206  81757  84212
2 Şubat    50838  55607  59159  63613  64379  66844  67912  74114  73515  75692
3 Mart     57261  60690  65895  67727  71663  73786  75459  79147  81215  84274
4 Nisan    56034  60060  63179  66769  70859  72909  74177  80340  80663  84239
5 Mayıs    62878  68131  69283  71704  80496  79584  82905  84951  87441  90369
6 Haziran  65211  65795  72126  77507  79433  80020  82472  88033  87008  90831
7 Temmuz   66197  68314  76919  83574  84224  84905  88453  88076  93177  95090
# ℹ 4 more variables: `2019` <dbl>, `2020` <dbl>, `2021` <dbl>, `2022` <dbl>
```

```
tail(ist,7)
```

```
# A tibble: 7 × 15
  Ay       `2009` `2010` `2011` `2012` `2013` `2014` `2015` `2016` `2017` `2018`
  <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1 Haziran   65211  65795  72126  77507  79433  80020  82472  88033  87008  90831
2 Temmuz    66197  68314  76919  83574  84224  84905  88453  88076  93177  95090
3 Ağustos   66583  75594  75957  80098  82987  83262  89859  91773  91794  93402
4 Eylül     61154  66734  71906  76396  80163  78215  84083  84767  87630  88053
5 Ekim      61555  66419  70069  75991  76077  77767  82263  85454  87114  86864
6 Kasım     59452  64087  65797  70456  73246  75344  79387  80802  84163  83213
7 Aralık    60567  65969  68282  71964  74388  77285  81133  80958  85165  84728
# ℹ 4 more variables: `2019` <dbl>, `2020` <dbl>, `2021` <dbl>, `2022` <dbl>
```

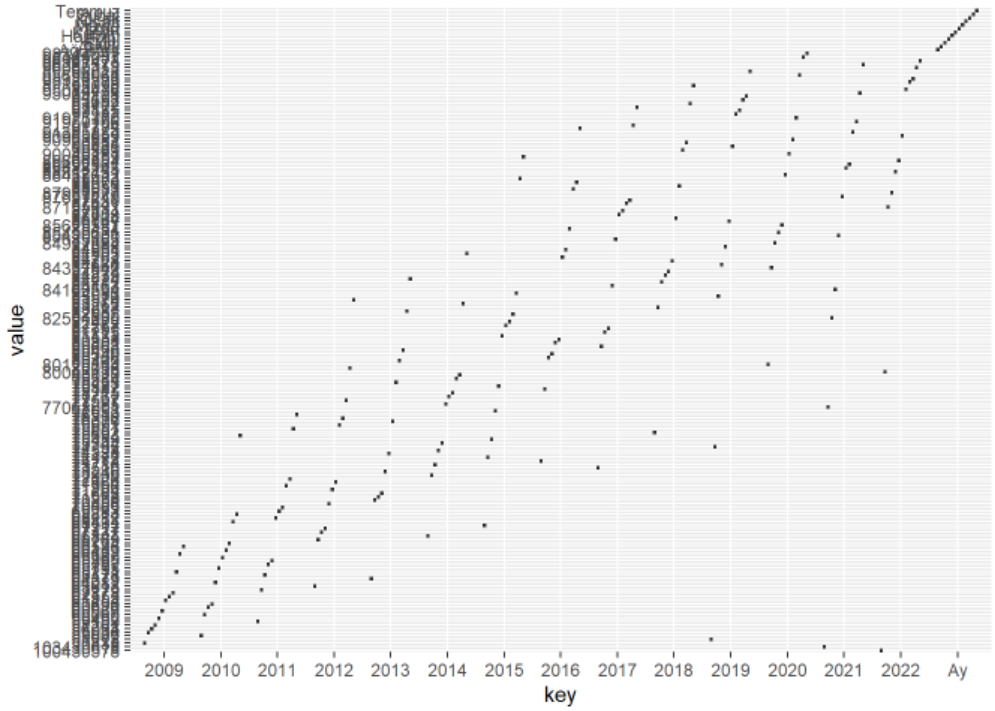
plot_intro() fonksiyonu ile veri seti hakkında bir giriş grafiği oluşturalım;

```
plot_intro(ist)
```



Aykırı değer incelemesi;

```
ist %>%
  gather() %>%
  ggplot(aes(key, value)) +
  geom_boxplot()
```

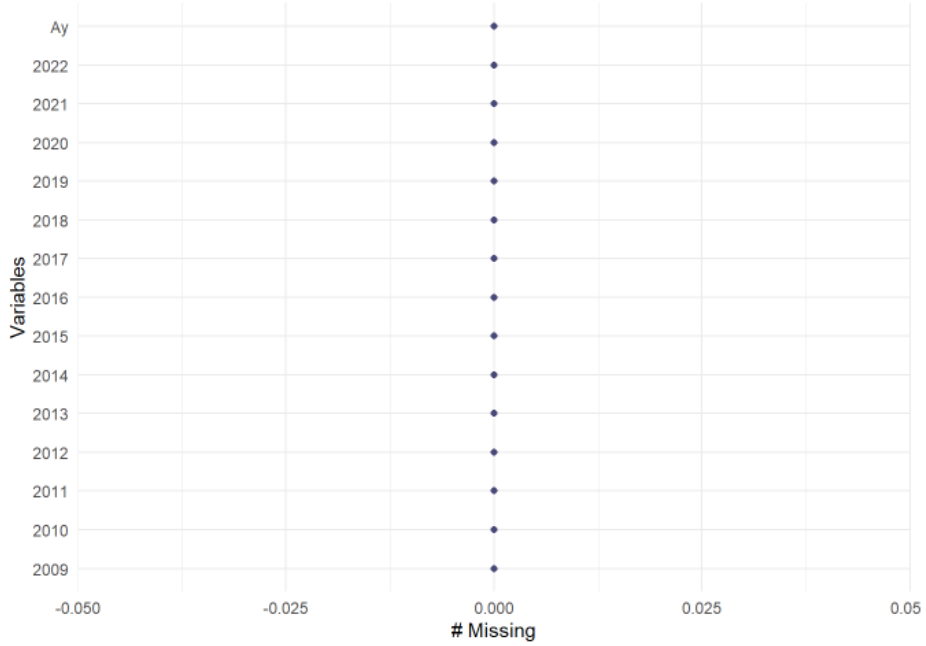


Eksik veri incelemesi;

```
# Her sütunda eksik değer sayısı
ist %>% summarise_all(~sum(is.na(.)))
```

```
# A tibble: 1 × 15
     Ay `2009` `2010` `2011` `2012` `2013` `2014` `2015` `2016` `2017` `2018`
  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>
1     0      0      0      0      0      0      0      0      0      0      0
# ℹ 4 more variables: `2019` <int>, `2020` <int>, `2021` <int>, `2022` <int>
```

```
# Eksik değerlerin görselleştirilmesi
gg_miss_var(ist)
```



Yukarıdaki tabloda görüldüğü üzere eksik veri bulunmamaktadır.

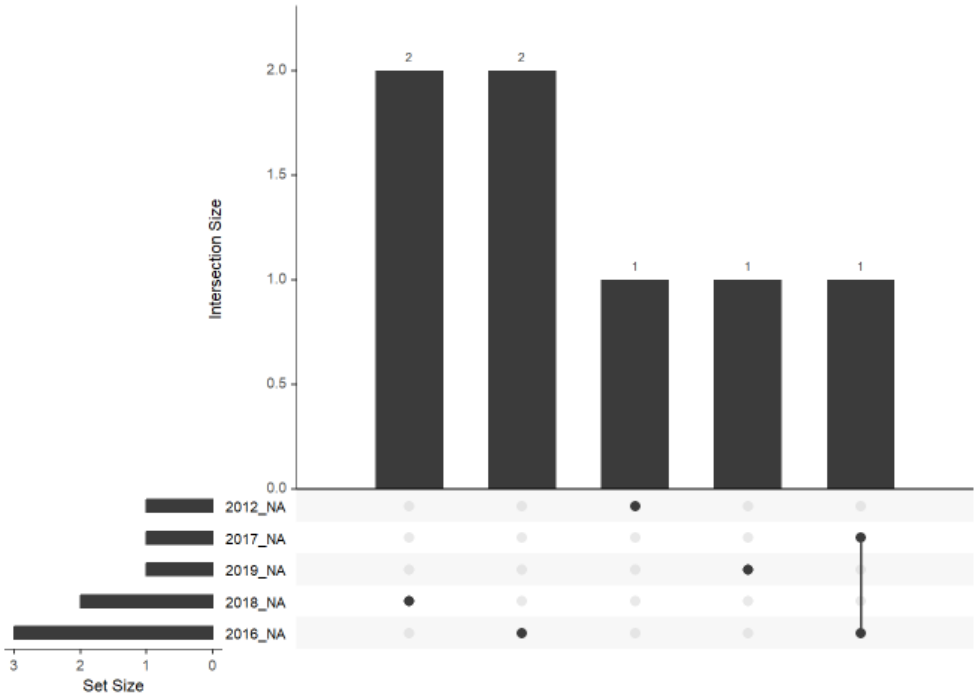
Grafikte görülen mavi noktalar eksik veri olmadığını gösteren "0" değerlerini temsil etmektedir.

Uygulama amaçlı eksik veri oluşturalım:

```
set.seed(123) # Rastgeleliği sabitlemek için

# ist veri setinin boyutunu al
n <- nrow(ist)
p <- ncol(ist)

# Rastgele eksik veri oluşturmak için hücrelerin %5'ini seç
missing_count <- round(0.05 * n * p)

# Eksik veriler için rastgele satır ve sütun indeksleri oluştur
missing_rows <- sample(1:n, missing_count, replace = TRUE)
missing_cols <- sample(1:p, missing_count, replace = TRUE)

# Oluşturulan indekslere göre veri setindeki hücrelere NA atayın
for(i in seq_along(missing_rows)) {
  ist[missing_rows[i], missing_cols[i]] <- NA
}
```

Eksik verileri görselleştirelim;

```
gg_miss_upset(ist)
```



Veri setinin bir zaman serisi veri seti olması nedeiyle eksik hücreleri doldurmada öncesi/sonrası değerleri kullanılarak doldurulmuştur. 'zoo' paketinin 'na.locf()' fonksiyonu, LOCF yani son gözlemin ileri taşınması yöntemiyle verileri doldurmak için;

```
# LOCF yöntemiyle eksik verileri doldurma
ist_filled <- ist
numeric_columns <- sapply(ist_filled, is.numeric) # Sayısal sütunları belirle

# Sayısal sütunlarda LOCF uygulama
ist_filled[, numeric_columns] <- lapply(ist_filled[, numeric_columns], function(x) {
  na.locf(x, na.rm = FALSE)
})
```

Doldurulan eksik verilerin kontrol edilmesi;

```
print(ist_filled)
```

```
# A tibble: 12 × 15
   Ay       `2009` `2010` `2011` `2012` `2013` `2014` `2015` `2016` `2017` `2018`
   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
 1 Ocak      55926  60626  65178  67137  71538  74528  77053  80206  81757  84212
 2 Şubat     50838  55607  59159  67137  64379  66844  67912  74114  73515  75692
 3 Mart      57261  60690  65895  67727  71663  73786  75459  74114  73515  84274
 4 Nisan     56034  60060  63179  66769  70859  72909  74177  80340  80663  84239
 5 Mayıs     62878  68131  69283  71704  80496  79584  82905  80340  87441  90369
 6 Haziran   65211  68131  72126  77507  79433  80020  82472  80340  87008  90831
 7 Temmuz    66197  68314  76919  83574  84224  84905  88453  88076  93177  95090
 8 Ağustos   66583  75594  75957  80098  82987  83262  89859  91773  91794  93402
 9 Eylül     61154  66734  71906  76396  80163  78215  84083  84767  87630  88053
10 Ekim      61555  66419  70069  75991  76077  77767  82263  85454  87114  88053
11 Kasım     59452  64087  65797  70456  73246  75344  79387  80802  84163  88053
12 Aralık    60567  65969  68282  71964  74388  77285  81133  80958  85165  84728
# ℹ 4 more variables: `2019` <dbl>, `2020` <dbl>, `2021` <dbl>, `2022` <dbl>
```

4. VERİ GÖRSELEŞTİRME

Veri setindeki bir sütun yılları temsil etmekte olup, satırlar ayları temsil etmektedir.

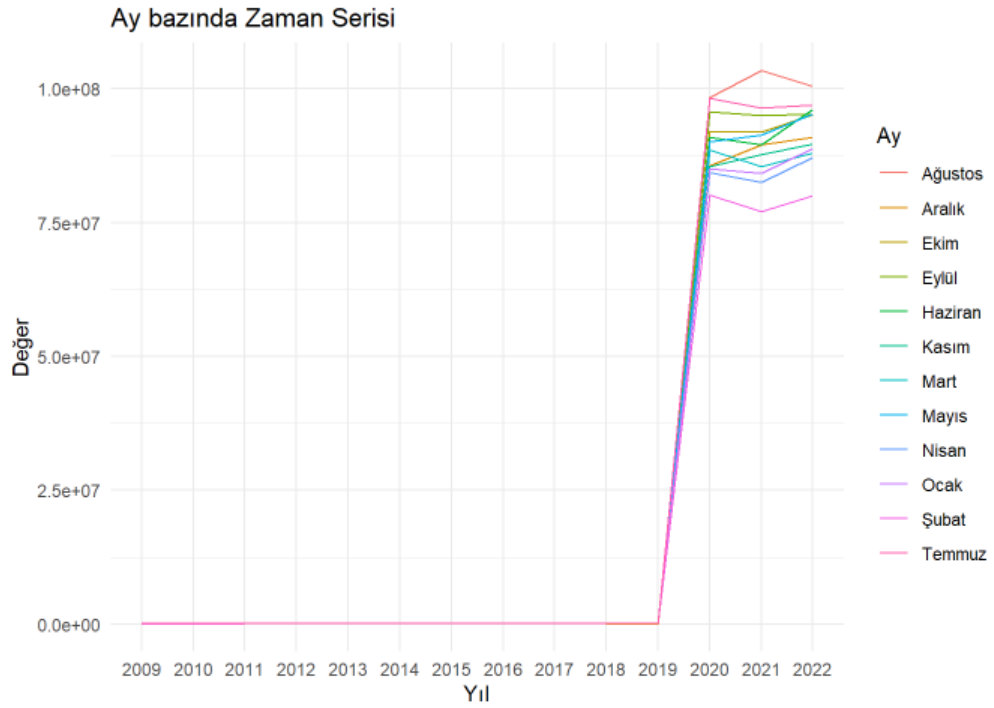
Görselleştirmelerin yapılabilmesi için her bir yıl-ay çifti için tek bir satır oluşturacak şekilde uzun formata dönüştürelim;

```
# Veri setini 'uzun' formata dönüştürme işlemi
ist_long <- pivot_longer(
  ist_filled,
  cols = -Ay, # 'Ay' sütunu dışındaki tüm sütunları dönüştür
  names_to = "Yıl",
  values_to = "Değer"
)
```

Daha sonra zamanla değişen değerleri göstermek için zaman serisi grafiği çizdirelim;

4.1 Zaman Serisi Grafiği

```
ggplot(ist_long, aes(x = Yıl, y = Değer, group = Ay, color = Ay)) +
  geom_line() +
  theme_minimal() +
  labs(title = "Ay bazında Zaman Serisi", x = "Yıl", y = "Değer")
```



Yukarıdaki zaman serisi grafiğinde, aylara göre yıllık bazda bir değerin nasıl değiştiği görülmektedir.

Grafiğe göre değerlerin çoğu zaman içinde birbirine paralel bir artış ya da azalış göstermektedir.

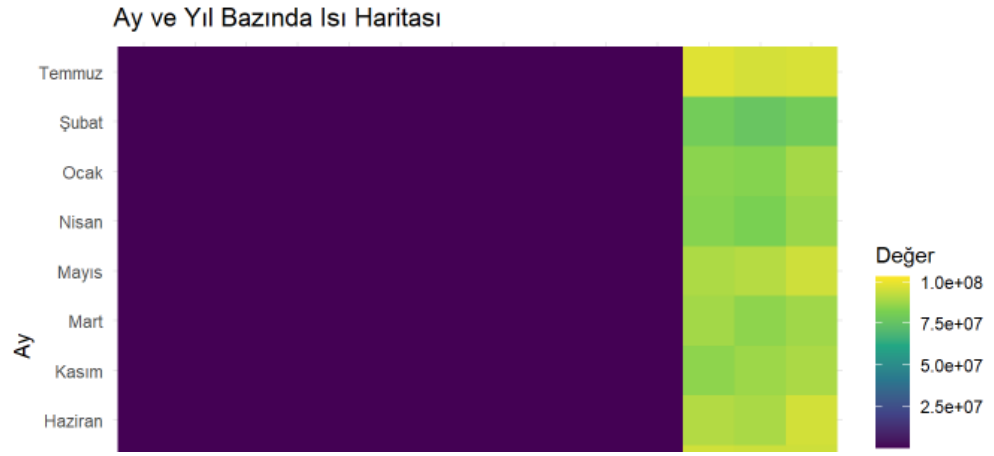
Ancak 2020 yılına gelindiğinde 2022 yılına kadar değerlerde ani bir şekilde değişim görülmektedir.

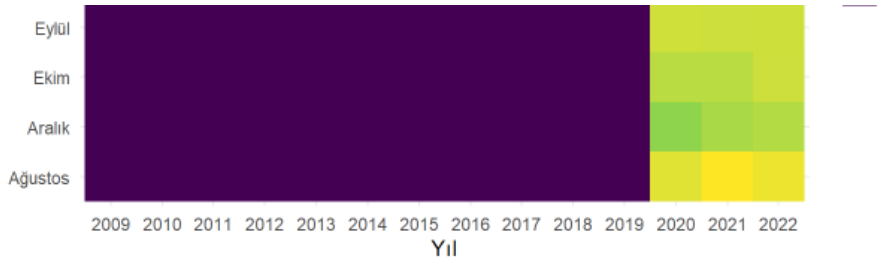
Bu değişimin kaynağı veri seti incelendiğinde açıkça görülmektedir ki veri toplama yönteminde bir değişikliğe gidilmiş olup değerler öncek yıllara göre değerlerde sayısal artış gözlenmektedir.

4.2 Isı haritası;

Isı haritasında zaman serisinde tanımlanan ve kullanılan veri çerçervesi üzerinden devam edilmektedir.

```
ggplot(ist_long, aes(x = Yıl, y = Ay, fill = Değer)) +
  geom_tile() +
  scale_fill_viridis_c() + # Renk paleti için viridis kullanabilirsiniz
  theme_minimal() +
  labs(title = "Ay ve Yıl Bazında Isı Haritası", x = "Yıl", y = "Ay")
```





Isı haritasında, her sütun bir yılı ve her satır bir ayı temsil etmektedir.

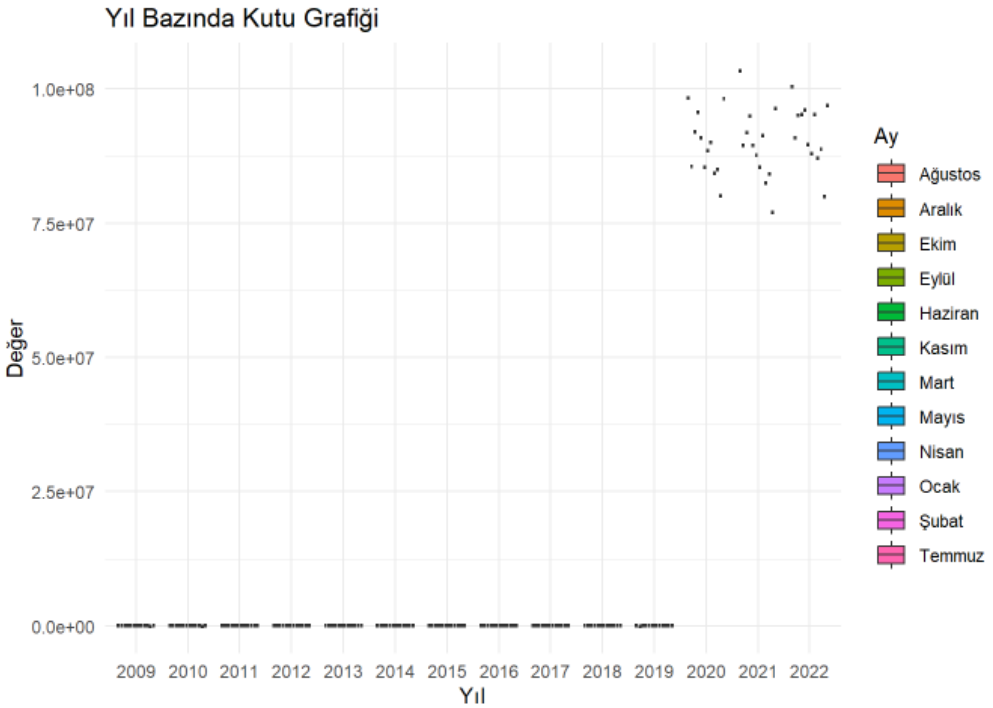
Renkler, o ayda o yıl için kaydedilen değerlerin büyüklüğünü gösterir.

Koyu renkler (mor) daha düşük değerleri gösterirken, açık (yeşil) renkler ise daha yüksek değerleri işaret etmektedir.

Yukarıda incelenen mevsimsellik grafiğinde de incelemiş olduğumuzdan aynı yorumların geçerli olduğu söylenebilir.

4.3 Kutu Grafiği;

```
# 'ist_long' veri çerçevesi kullanılarak kutu grafiği çizimi
ggplot(ist_long, aes(x = Yıl, y = Değer, fill = Ay)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Yıl Bazında Kutu Grafiği", x = "Yıl", y = "Değer")
```



4.4 Pasta Grafiği

Veri setindeki her satır bir ay değerini göstermektedir ve 12 satırdan oluşmaktadır.

15 sütun vardır. İlk sütun ayları içermekte olup, stringtir. 14 yılın verisi diğer sütunları oluşturmakta olup numerictir.

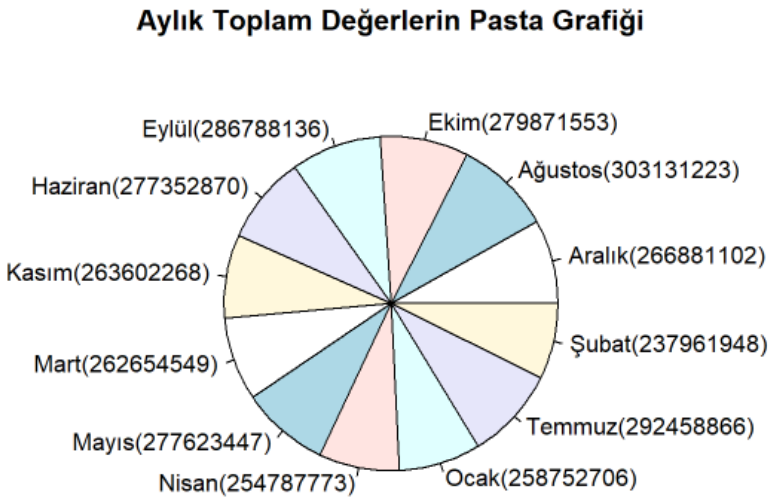
Satır toplamları alınarak aylara göre mevsimsellik durumu olup olmadığını incelemenin bir yolu da pasta grafiğidir.

Öncelikle aylık bazda değerleri incelemek için satır bazlı toplamları alalım;

```
# Her ay için yıllık toplamları hesapla
monthly_totals <- ist_filled %>%
  gather(Yıl, Değer, -Ay) %>%
  group_by(Ay) %>%
  summarise(Total = sum(Değer, na.rm = TRUE))

# Pasta grafiği için verileri hazırla
monthly_totals$label <- paste(monthly_totals$Ay, "(", monthly_totals$Total, ")", sep="")
```

Aldığımız satır toplamlarını görselleştirelim;

```
# Pasta grafiği çiz
pie(monthly_totals$Total, labels = monthly_totals$label, main = "Aylık Toplam Değerlerin Pasta G
```



Grafikteki en yüksekdeğere sahip olan ay ile en düşük değere sahip olan ayın yüzde olarak farkını hesaplayalım;

```
# En yüksek ve en düşük değerleri içeren ayları bul
max_value <- max(monthly_totals$Total)
min_value <- min(monthly_totals$Total)
```

```r
# Bu değerlere sahip ayları bul
max_month <- monthly_totals$Ay[which.max(monthly_totals$Total)]
min_month <- monthly_totals$Ay[which.min(monthly_totals$Total)]

# Yüzde olarak farkı hesapla
percent_difference <- ((max_value - min_value) / max_value) * 100

# Sonuçları yazdır
cat("En yüksek değere sahip ay:", max_month, "ile", max_value, "değeri.\n")
```

```
En yüksek değere sahip ay: Ağustos ile 303131223 değeri.
```

```r
cat("En düşük değere sahip ay:", min_month, "ile", min_value, "değeri.\n")
```

```
En düşük değere sahip ay: Şubat ile 237961948 değeri.
```

```r
cat("Bu iki ay arasındaki yüzde fark:", percent_difference, "%\n")
```

```
Bu iki ay arasındaki yüzde fark: 21.4987 %
```