

Packages/ Paketler

Import Excel Data / Excel Verisi Al

Data Explorer / Veriye Bir Bakış

Missing Values / Kayıp Veri

Data Manipulation / Veri Manipülasyonu

Creating Leagues / Lig Oluşturma

Create Positions Class / Pozisyon Sınıfı Oluşturma

Data Transformation

Changing The Category Variable to Factor / Kategorik Değişkeni Faktöre Cevirme

Rename Variables / İsmi Değiştirme

Remove Unnecessary Variables / Kullanmayacağım Verileri Atma

Data Analysis & Visualization / Data Analizi ve Görselleştirme

Fifa-2020 VERİ ANALİZİ VE GÖRSELLEŞTİRME

Kevser Bahadır

2023-01-31

Packages/ Paketler

```
library(tidyverse)
library(magrittr)
library(DataExplorer)
library(maps)
library(plotly)
library(DT)
library(tidytext)
library(gridExtra)
library(readxl)
library(ggplot2)
library(dplyr)
library(plotly)
library(tidyr)
```

Import Excel Data / Excel Verisi Al

```
df <- read_excel("fifa20.xlsx")
dim(df)
```

```
## [1] 18278 104
```

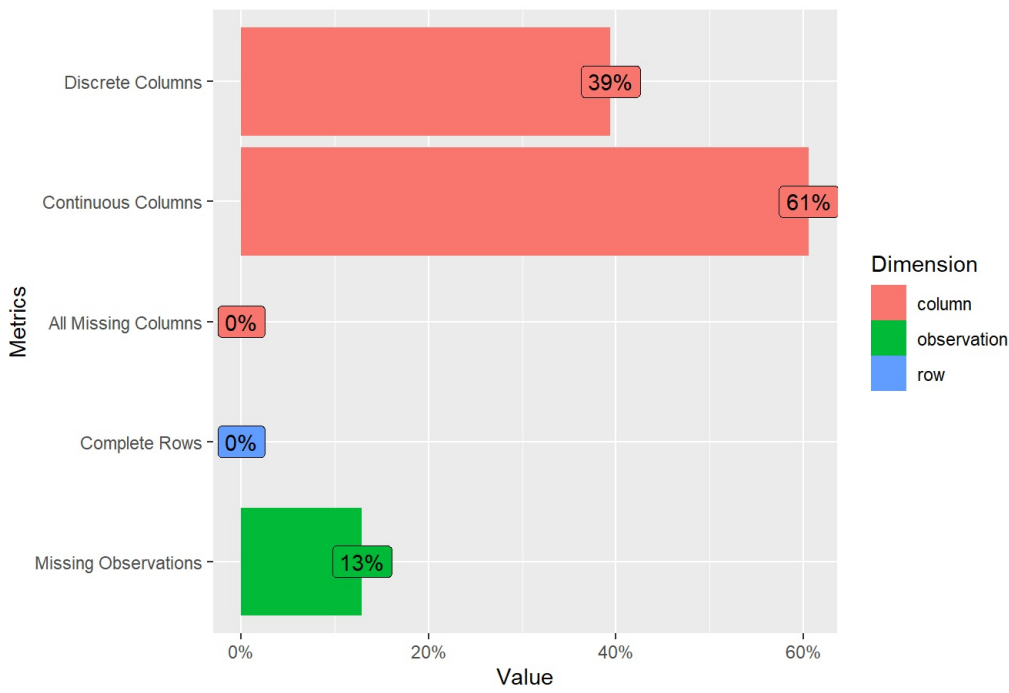
Data Explorer / Veriye Bir Bakış

```
introduce(df)
```

```
## # A tibble: 1 × 9
##   rows columns discrete_columns conti...1 all_m...2 total...3 compl...4 total...5 memor...6
##   <int>   <int>           <int>   <int>   <int>   <int>   <int>   <dbl>
## 1 18278    104             41      63      0 244935      0 1900912 2.03e7
## # ... with abbreviated variable names 1continuous_columns, 2all_missing_columns,
## # 3total_missing_values, 4complete_rows, 5total_observations, 6memory_usage
```

```
plot_intro(df)
```

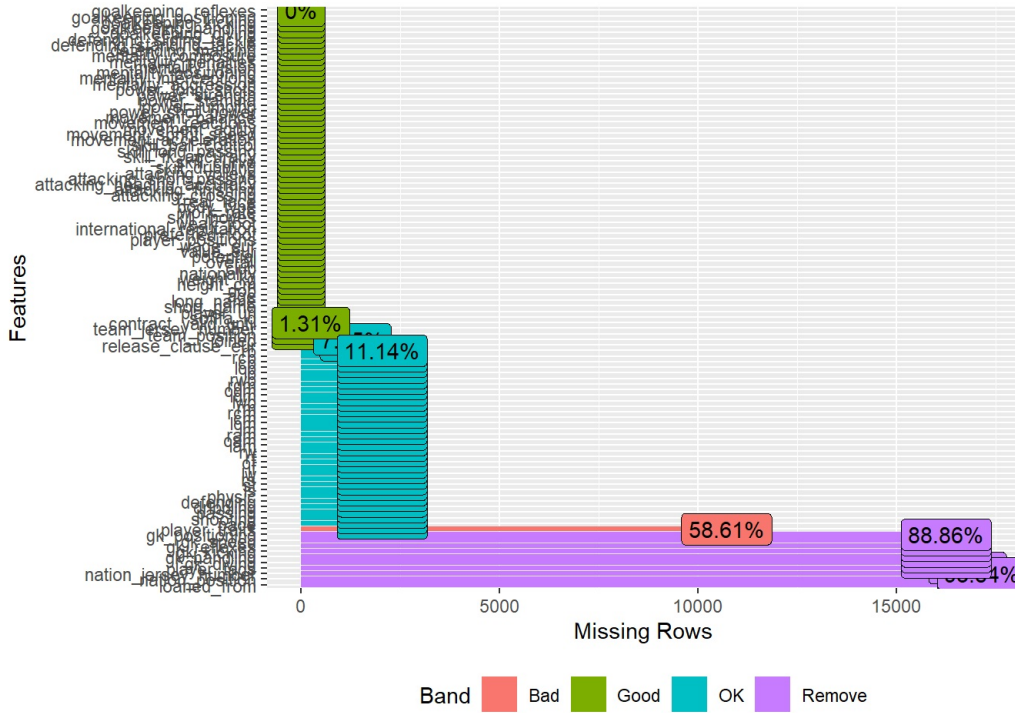
Memory Usage: 19.3 Mb



Missing Values / Kayıp Veri

Veri de kayıp veri olduğunu `plot_missing` ile rahat bir şekilde gözlemliyoruz.

```
plot_missing(df)
```



Data Manipulation / Veri Manipülasyonu

veri setinde hangi futbolcu hangi ligte oynuyor bilmiyoruz. Ben ligler üzerinde analiz yapmak istiyorum. Bu yüzden bir Legues sütunu oluşturup takımları hangi lig olduğunu tespit edip oluşturduğum 4 lig (nundesliga, laliga, seria, superlig) 'e atıyoruz.

```

bundesliga <- c( "1. FC Nürnberg" , "1. FSV Mainz 05" , "Bayer 04 Leverkusen" , "FC Bayern MÃ¼nchen" , "Borussia D
ortmund" , "Borussia Mönchengladbach" , "Eintracht Frankfurt" , "FC Augsburg" , "FC Schalke 04" , "Fortuna Düsseldorf" , "Hannover 96" , "Hertha BSC" , "RB Leipzig" , "SC Freiburg" , "TSG 1899 Hoffenheim" , "VfB Stuttgart" , "VfL W
olfsburg" , "SV Werder Bremen"
)

laliga <- c( "Athletic Clup de Bilbao" , "AtlÃ©tico Madrid" , "CD Leganes" , "Deportive Alaves" , "FC Barcelona" , "
Getafe CF" , "Girona FC" , "Levente UD" , "Rayo Vallecano" , "RC Celta" , "RCD Espanyol" , "Real Betis" , "Real Madri
d" , "Real Sociedad" , "Real Valladolid CF" , "SD Eibar" , "SD Huesca" , "Sevilla FC" , "Valencia CF" , "Villarreal C
F"
)

seriea <- c( "Atlanta" , "Bologna" , "Cagliari" , "Chievo Verona" , "Empoli" , "Fiorentina" , "Frosinone" , "Genoa" , "Int
er" , "Juventus" , "Lazio" , "Milan" , "Napoli" , "Parma" , "Roma" , "Sampdoria" , "Sassuolo" , "SPAL" , "Torino" ,
"Udinesed"
)

superlig <- c( "FenerbahÅŒe SK" , "Trabzonspor" , "Alanyaspor" , "Antalyaspor" , "Besiktas" , "BB Erzurumspor" , "
Çaykur Rizespor" , "Denizlispor" , "Fatih Karagömrük" , "Galatasaray SK" , "Gaziantep FK" , "Gençlerbirliđi" , "Göz
tepe" , "Hatayspor" , "İstanbul Başakşehir" , "KasımpaŒa" , "Konyaspor" , "MKE Ankaragücü" , "Sivasspor" , "Yeni Malat
yaspor"
)

```

Creating Leagues / Lig OluŒturma

Œimdi Datamızın içindeki club eđer bundesliga liginden bir takım ise “Bundesliga” yaz, eđer laliaga liglerini içeriyorsa “Laliaga” , seria takımlarını içeriyorsa “Serie A” , superlig takımları ise “Süper Lig” yazsın

Daha sonrasında League hangi ülkeye aitse yanına ülkenin adı yazılsın istiyorum.

```

df %<>% mutate( League = if_else(club %in% bundesliga , "Bundesliga" ,
                                if_else(club %in% laliga , "Laliga" ,
                                          if_else(club %in% seriea , "Serie A",
                                                  if_else(club %in% superlig, "Süper Lig",
                                                          NA_character_ )))),

                                Country = if_else( League == "Bundesliga" , "Germany" ,
                                                  if_else( League == "Laliga" , "Spain" ,
                                                          if_else( League == "Serie A" , "Italy" ,
                                                                  if_else( League == "Süper Lig" , "Turkey" , NA_character_ ))))) %>%
  filter(!is.na(League)) %>% mutate_if(is.factor , as.character())
rm(bundesliga, laliga, seriea, superlig)

```

4 lig tercih etmem tabiiki de beklediđimiz gibi örneklemi küçülttü. 18.278 tane gözlemden 1.719 tane gözleme indi verimiz.

#String Manipulation / Dizi Manipölasyonu

value_eur pazar değeri kategorik veri olsaydı onu numerik hale getirmek için aŒağıdaki işlemler yapılmalıydı.(datamız da £, M ,K gibi Œeyler içermiyor bu bölüm bilgilendirme için verilmiŒtir.)

```

df$values <- str_remove_all(df$value_eur , "£") #remove çıkartmak
df$values <- str_replace_all(df$value_eur , "K" , "000") #k'yi çıkart 000 ekle
df$values <- str_remove_all(df$value_eur , "M") #m 'yi çıkart

df$values <- as.numeric(df$values)

df$wages <- str_remove_all(df$wage_eur , "£")
df$wages <- str_replace_all(df$wage_eur , "K" , "000")

df$wages <- as.numeric(df$wages)

df <- df %>% mutate(Values = if_else(Values < 1000 , Values * 100000000 , Values))

```

Create Positions Class / Pozisyon Sınıfı OluŒturma

```

defance <- c("CB" , "RB" , "LB" , "LWB" , "RWB" , "LCB" , "RCB")
midfielder <- c ("CM" , "CDM" , "CAM" , "LM" , "RM" , "LAM" , "RAM" , "LCM" , "RCM" , "LDM" , "RDM")

```

Yukarıda League nasıl yaptıysak aynı işlemleri Positions'a uyarlaması Œeklinde uyguluyoruz.

```
df %<>% mutate(Class = if_else(team_position %in% "GK", "Goal Keeper" ,
                                if_else(team_position %in% defance , "Defender" ,
                                          if_else(team_position %in% midfielder , "Midfielder" , "Forward"))))
rm(defance,midfielder)
```

Data Transformation

Changing The Category Variable to Factor / Kategorik Değişkeni Faktöre Çevirme

Preferred foot (tercih edilen ayak) kategorik bir değişken onu faktöre çeviriyoruz.

```
df %<>% filter(preferred_foot %in% c("Left" , "Right"))
df$preferred_foot <- as.factor(as.character(df$preferred_foot))
```

Rename Variables / Ismi Değiştirme

```
df %<>%
  rename(

    "long.name" = long_name
  )
```

Remove Unnecessary Variables / Kullanmayacağım Verileri Atma

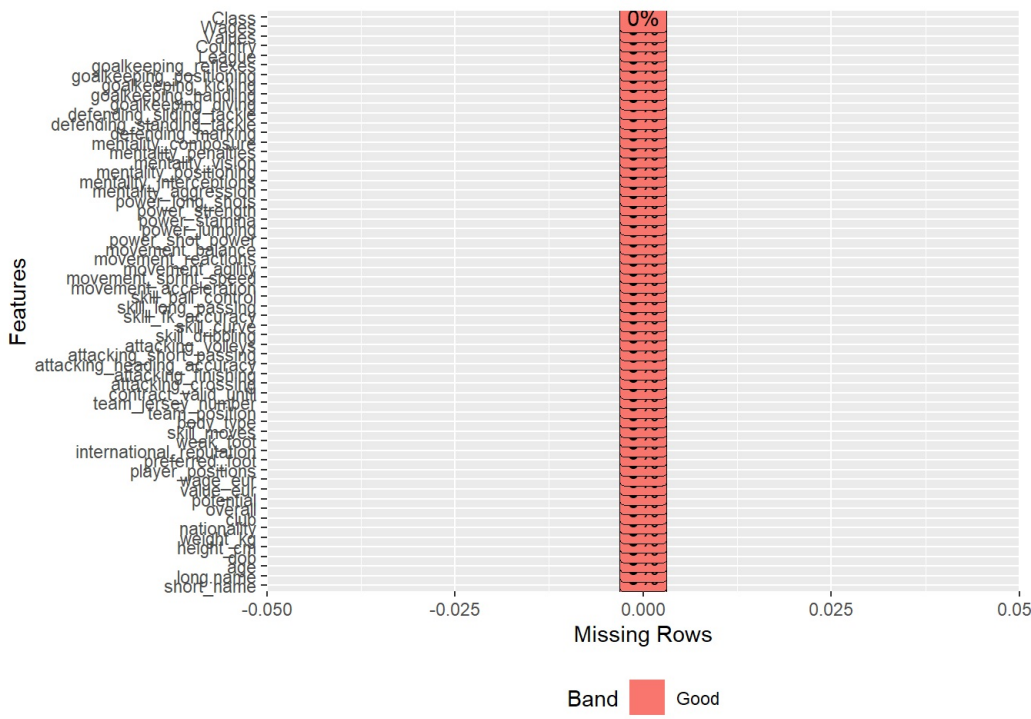
```
df %<>% select( - sofifa_id , -player_url , -work_rate , -real_face , -loaned_from, -nation_position ,
               -nation_jersey_number , -player_tags,-pace , -shooting , -passing , -dribbling, -defending, -physic
               ,
               -gk_diving , -gk_handling , -gk_kicking , -gk_reflexes , -gk_speed , -gk_positioning , -player_traits
               )

df %<>% select( -ls , -st , -rs , -lw , -lf , -cf , -rf , -rw , -lam , -cam , -ram , -lm , -lcm , -cm , -rcm , -rm
               , -lwb , -ldm , -cdm , -rdm , -rwb , -lb , -lcb , -cb , -rcb , -rb)

df %<>% select( -joined , -release_clause_eur)
```

Kayıp veri var mı diye bakıyoruz.

```
plot_missing(df)
```

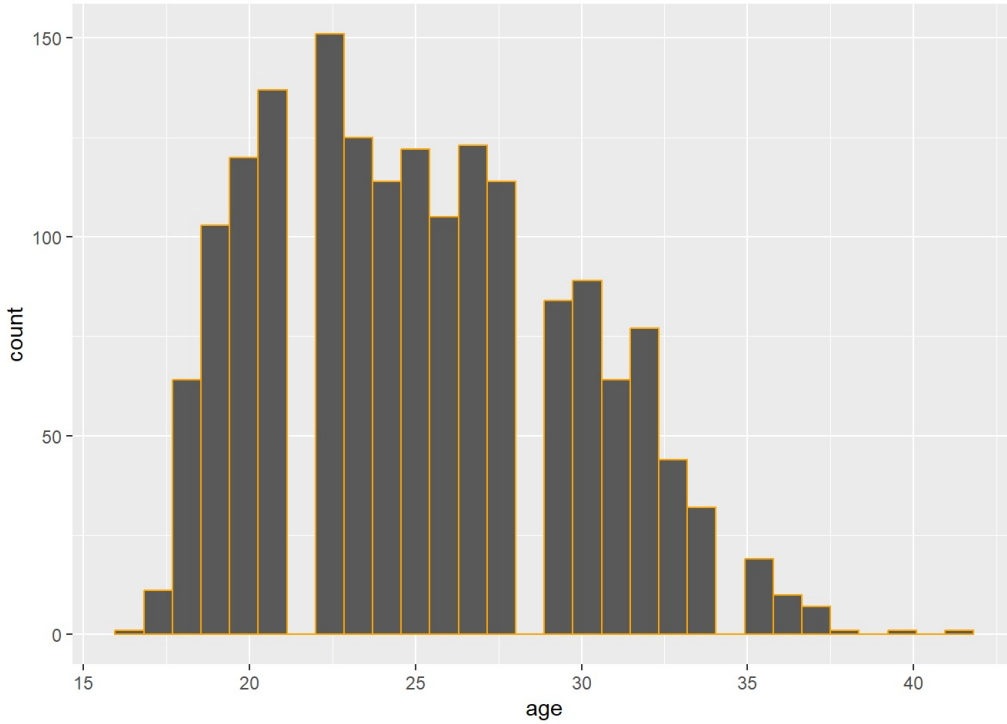


Data Analysis & Visualization / Data Analizi ve Görselleştirme

SORU: Liglerin yaş ortalamasını bulalım.

Genel Yaş Dağılımı

```
ggplot(df, aes(age)) +  
  geom_histogram( color= "orange")
```



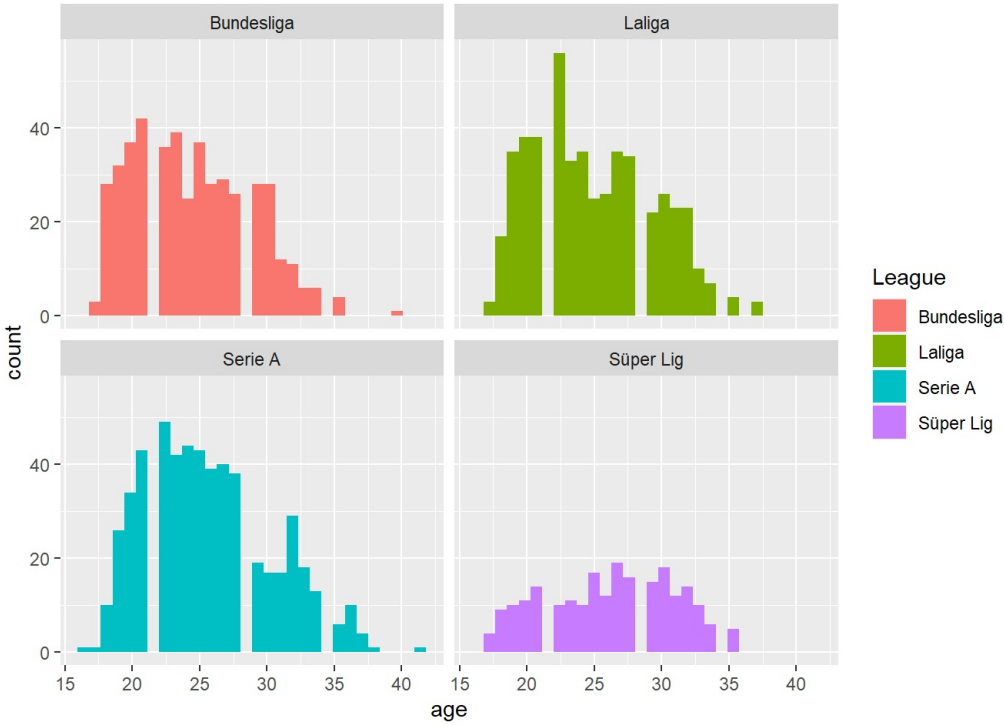
Liglere Göre Yaş Ortalamaları

```
df %>%  
  group_by(League) %>% #kırınımlar liglere göre olsun  
  summarise(Avg.age = mean(age)) %>%  
  arrange(desc(Avg.age)) #küçükten büyüğe sıralama
```

```
## # A tibble: 4 × 2
##   League   Avg.age
##   <chr>     <dbl>
## 1 Süper Lig  26.2
## 2 Serie A    25.6
## 3 Laliga     24.9
## 4 Bundesliga 24.4
```

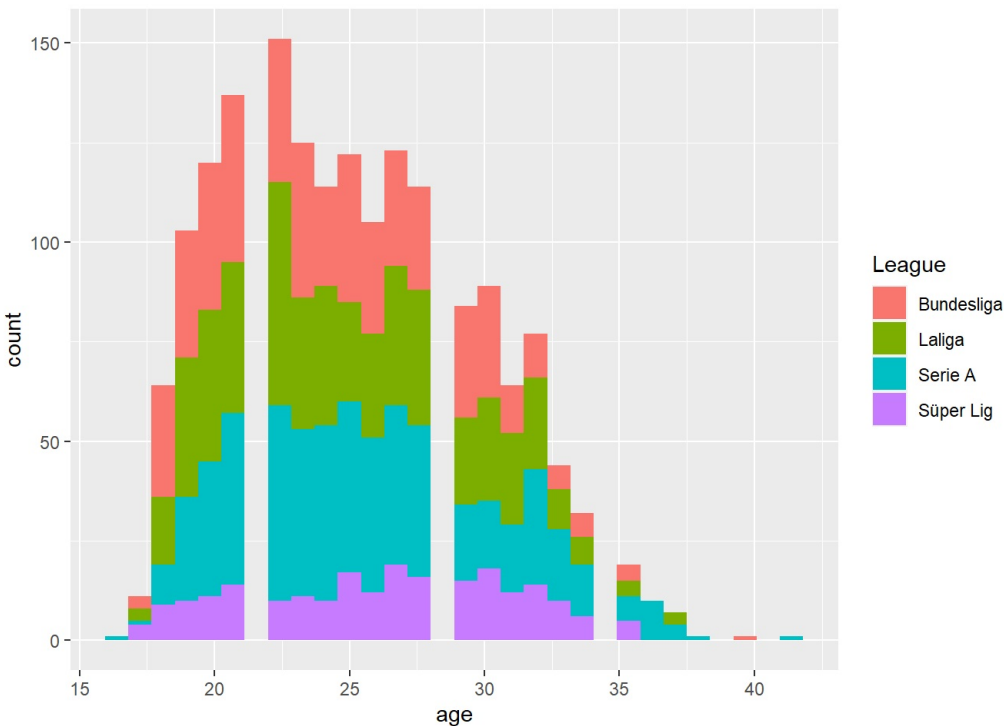
```
ggplot(df, aes(age , fill = League)) +
  geom_histogram()+
  facet_wrap(League~.) #her bir ligin yaş dağılımlarını
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
#bu yolla da yapılabilir.
ggplot() +
  geom_histogram(df, mapping = aes(age, fill = League))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



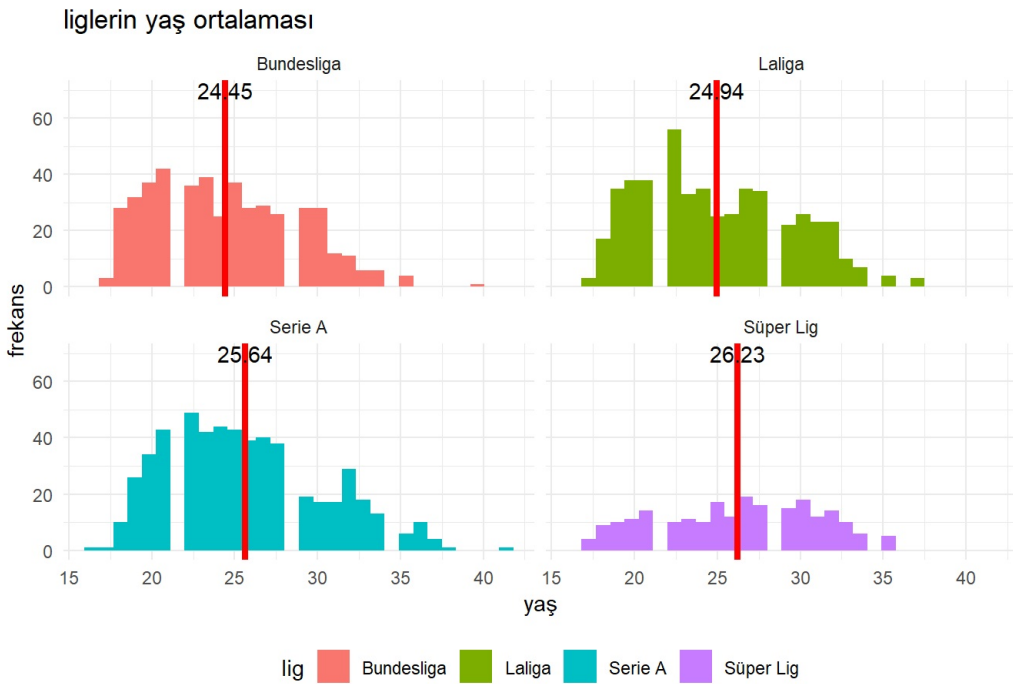
Liglerin Yaş Dağılımlarını Ve Ortalamalarını Gösteren Grafik

```
summ <- df %>%
  group_by(League) %>% #kırınımlar liglere göre olsun
  summarise(Age = mean(age) , median = median(age))

ggplot() +
  geom_histogram(df, mapping = aes(age, fill = League)) + #fill estetik özellik
  facet_wrap(League~.) +
  geom_vline(summ , mapping = aes(xintercept = Age) , color = "red" , size = 1.5) + #age ortalamalarını çizdi
  geom_text(summ ,mapping = aes(x= Age , y=70 , label = round(Age,2))) +
  theme_minimal() + #sadece arka plandaki grimsi yapı gitti.
  theme(legend.position = "bottom") + #league yerini değiştirme
  labs(x= "yaş" , y = "frekans" , fill ="lig" , title = "liglerin yaş ortalaması", caption = "@EA Sports-Fifa19")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



@EA Sports-Fifa19

Soru: Her lige göre yaş ortalaması 25'ten büyük kaç kişi var?

```
df %>%
  filter(age > 25) %>%
  group_by(League) %>%
  count(short = TRUE)
```

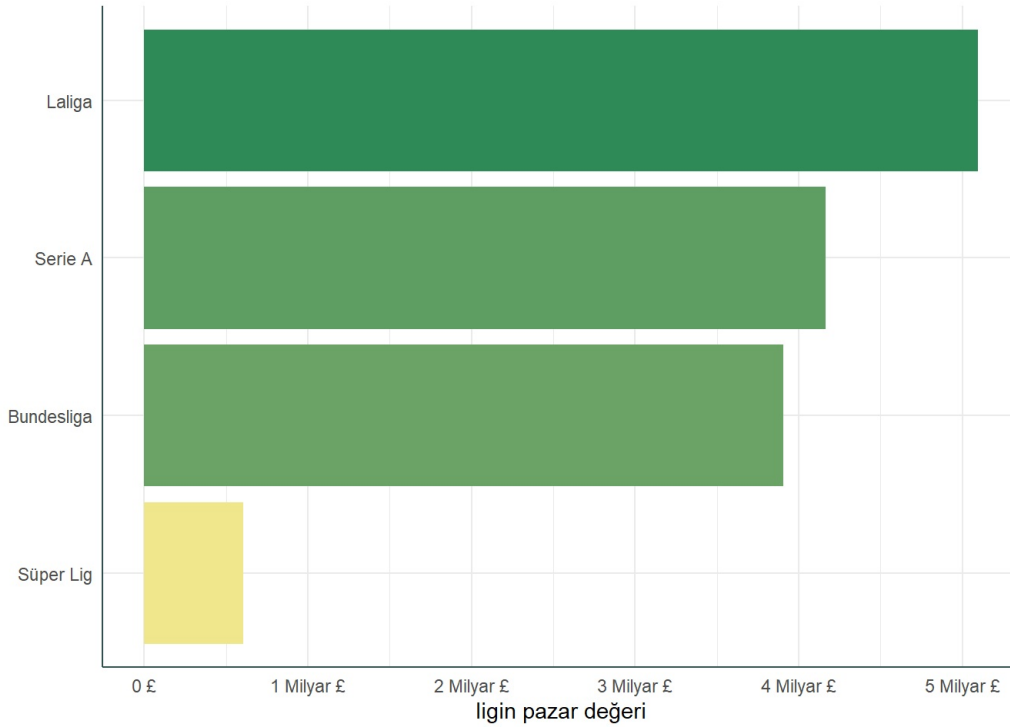
```
## # A tibble: 4 x 3
## # Groups:   League [4]
##   League short    n
##   <chr>   <lg> <int>
## 1 Bundesliga TRUE    179
## 2 La Liga TRUE    213
## 3 Serie A TRUE    252
## 4 Süper Lig TRUE    127
```

Soru: Liglerin pazar değerlerinin toplamı sürelerini görselleştirelim.

```
df %>%
  group_by(League) %>%
  summarise(Total.Value = sum(as.integer(Values), na.rm = TRUE)) %>%
  arrange(desc(Total.Value)) #büyükten küçüğe sıralama
```

```
## # A tibble: 4 × 2
##   League      Total.Value
##   <chr>      <dbl>
## 1 Laliga      5093870000
## 2 Serie A      4166220000
## 3 Bundesliga  3904165000
## 4 Süper Lig    604270000
```

```
df %>%
  group_by(League) %>%
  summarise(Total.Value = sum(as.integer(Values),na.rm = TRUE)) %>%
  ggplot(aes(reorder(League , Total.Value) , Total.Value , fill= Total.Value)) +
  geom_col(show.legend = FALSE) +
  coord_flip()+ #X VE Y KOORDİNATLARINI DEĞİŞTİRDİ
  theme_minimal()+ #RENGİNİ AÇTI
  labs(x= NULL, y= "ligin pazar değeri") + #SADECE Y DEĞERİNİ İSİMLENDİRDİ
  scale_fill_gradient(low="khaki" , high = "seagreen") + #DÜŞÜK OLANDAN YÜKSEK OLANA RENK GEÇİŞİ
  theme(axis.line.y = element_line(colour = "darkslategray"),
        axis.line.x = element_line(colour = "darkslategray")) +
  scale_y_continuous(labels = c("0 £" , "1 Milyar £" , "2 Milyar £" , "3 Milyar £" , "4 Milyar £" , "5 Milyar £"))
#y eksenini değerlerini değiştirdik.
```



SORU: İnteraktif Dünya haritası ile futbolcuları ülkelerine göre dağılımlarını görelim.

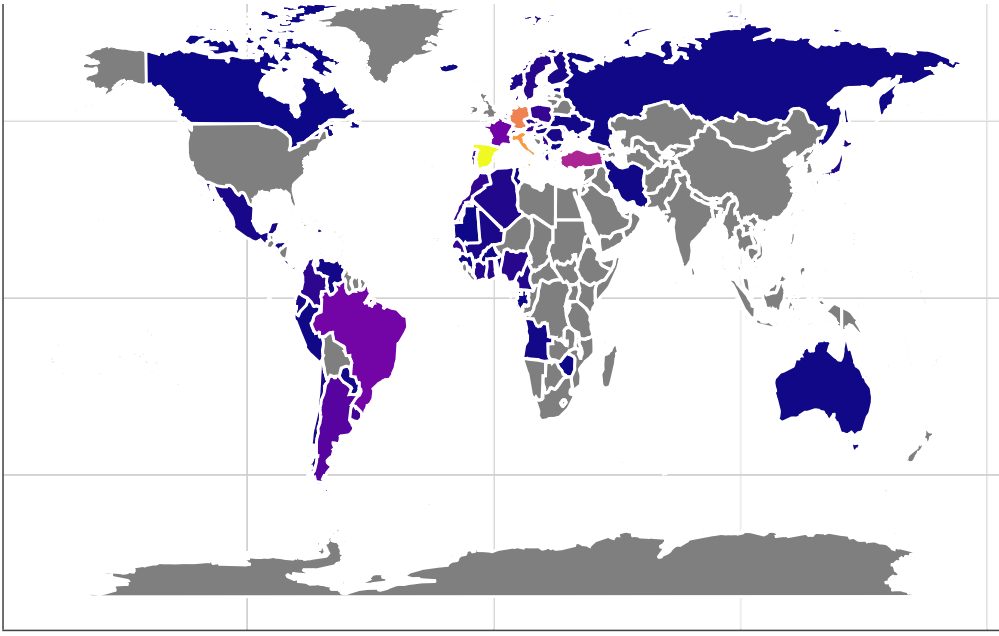
```
world_map <- map_data("world")

numofplayers <- world_map %>%
  mutate( region = as.character(region)) %>%
  left_join((df %>% mutate( nationality = as.character(nationality) ,
                          nationality = if_else(nationality %in% "England" ,
                                                "UK" , nationality)) %>%
            count(nationality , name = "Number of Player" ) %>%
            rename(region = nationality) %>%
            mutate(region = as.character(region))) , by= "region")

ggplotly(
  ggplot(numofplayers , aes(long , lat , group = group )) +
  geom_polygon(aes(fill = `Number of Player` ) , color = "white" , show.legend = FALSE) +
  scale_fill_viridis_c(option = "C") +
  theme_void() +
  labs(fill = "Futbolcu Sayısı" ,
       title = "Hangi Ülkeden Kaç Futbolcu Var?")
)
```

Hangi Ülkeden Kaç Futbolcu Var?





Soru: Futbolcuların Kitle Endeksinin Hesaplayınız.

BMI'a göre;

- İdeal Kilonun Altı: < 18,49
- İdeal Kilo: 18.5-24,99
- İdeal Kilonun Üzere: 25-29,99
- İdeal Kilonun Çok Üzeri: > 30

```
#vucut kitle indeksi hesabı
```

```
bmi <- df %>%  
  filter(club == "Juventus") %>%  
  mutate(BMI = round(weight_kg/(height_cm/100)^2 , digits = 4 )) %>%  
  arrange(-BMI) %>%  
  select(short_name , age , player_positions , Class , height_cm , weight_kg , BMI)
```

```
#interaktif sonuc
```

```
datatable(bmi)
```

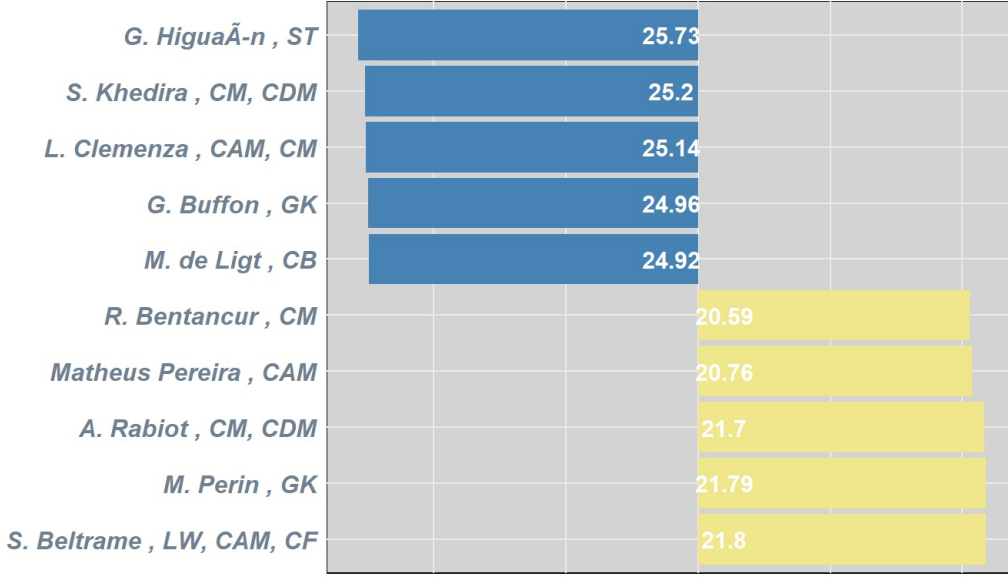
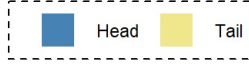
```
#ilk ve son 5'er gözlemler
```

```
bmi2 <- rbind(  
  bmi %>% head(5) %>% mutate(BMI = BMI * -1),  
  bmi %>% tail(5)  
) %>% mutate(Type = if_else(BMI < 0, "Head" , "Tail"))
```

Futbolcuların vücut kitle endekslerinin görselleştirilmesi

```
bmi2 %>%  
  ggplot(aes(fct_reorder(paste(short_name, "," , player_positions) , desc(BMI)) , BMI))+  
  geom_col(aes(fill = Type))+  
  geom_text(aes(y = c(rep(-2,5), rep(2,5)) , label= round(abs(BMI) , digits = 2 )) , color = "white" , fontface = "  
bold" , size = 4 )+  
  coord_flip()+  
  theme_minimal()+  
  theme(axis.text.x = element_blank() ,  
    legend.position = "top",  
    panel.background = element_rect(fill = "lightgray") ,  
    panel.grid.minor = element_blank(),  
    axis.text = element_text(color = "slategray" , face = "bold.italic" , size = 12),  
    title = element_text(color = "slategray" , face = "bold.italic" , size = 20),  
    legend.box.background = element_rect(linetype = 2))+  
  labs(x = NULL , y = NULL , fill = NULL , title = "BMI INDEX") +  
  scale_fill_manual(values = c("steelblue" , "khaki"))
```

BMI INDEX



Tablo yorumu:

Tabloda en yüksek vücut kitle indeksi olan ve en düşük vücut kitle indekse sahip olanlar grafikte gösterilmektedir. Oyuncuların bir çoğu ideal kilodadır.

Soru: Bitiricilikle ayak tercihi arasında istatistikel olarak anlamlı fark var mıdır, yok mudur?(%99 anlamlılık düzeyinde) test ediniz.

h0: bitiricilikle ayak tercihi arasında istatistikel olarak anlamlı bir ilişki yoktur. h1: bitiricilikle ayak tercihi arasında istatistiksel olarak anlamlı bir ilişki vardır.

Normallik testi

```
kor <- df %>%
  filter(League == "Serie A" , Class == "Forward") %>%
  select(short_name , preferred_foot, attacking_finishing, power_shot_power)
shapiro.test(kor$power_shot_power)
```

```
##
## Shapiro-Wilk normality test
##
## data: kor$power_shot_power
## W = 0.96811, p-value = 1.78e-07
```

Anlamlılık testi

```
xt1 <- kor %>% filter(preferred_foot == "Left") %>% select(power_shot_power) %>% pull()
xt2 <- kor %>% filter(preferred_foot == "Right") %>% select(power_shot_power) %>% pull()
yt1 <- kor %>% filter(preferred_foot == "Right") %>% select(attacking_finishing) %>% pull()
yt2 <- kor %>% filter(preferred_foot == "Left") %>% select(attacking_finishing) %>% pull()
xht <- wilcox.test(xt1, xt2, alternative = "two.sided")
yht <- wilcox.test(yt1, yt2, alternative = "two.sided")
xht;yht
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: xt1 and xt2
## W = 15775, p-value = 0.2375
## alternative hypothesis: true location shift is not equal to 0
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: xt1 and xt2
## W = 15775, p-value = 0.2375
## alternative hypothesis: true location shift is not equal to 0
```

alpha < p value ho red edilemez. iki değer arasında anlamlı bir fark yoktur.

SORU: Bitiricilikle yaş arasında bir fark var mıdır? Saçılım grafiği kullanınız.

```
p3d <- df %>% filter(League == "Süper Lig")

plot_ly(p3d , x= ~attacking_finishing , y = ~age , z= ~power_shot_power , color = ~Class , text = ~short_name ) %>%
  add_markers() %>%
  layout(
    scene = list(xaxis = list(title = "bitiricilik"),
                  yaxis= list(title = "yaş"),
                  zaxis = list (title = "sut gücü"))
  )
```

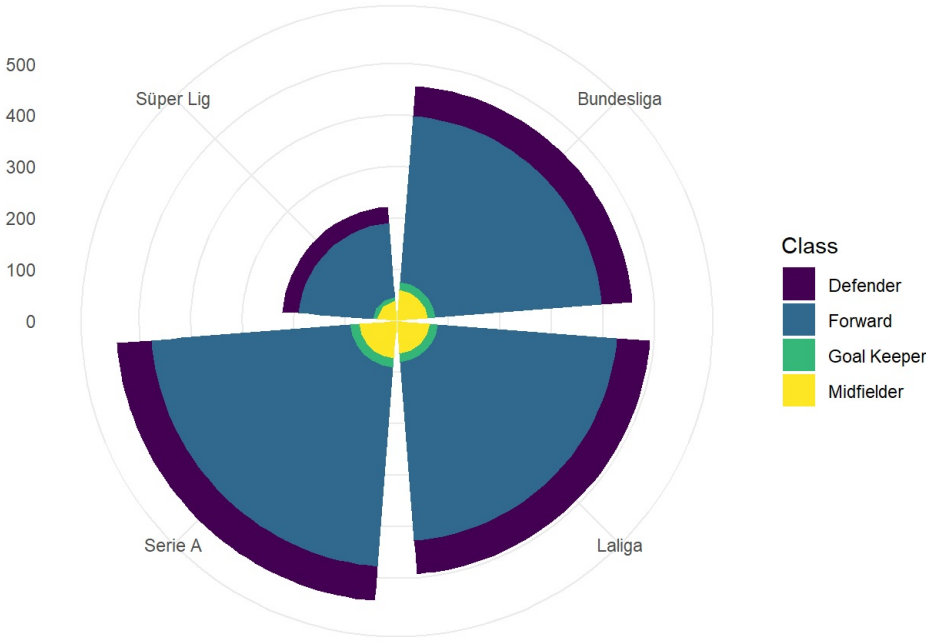


- Defender
- Forward
- Goal Keeper
- Midfielder

WebGL is not supported by
your browser - visit
<https://get.webgl.org> for
more info

SORU: Ligler içerisindeki pozisyon sınıflarının dağılımı nasıldır?

```
df %>% group_by(League) %>% count(Class) %>%
  ggplot (aes(League, n, fill = Class)) +
  geom_col()+
  coord_polar()+ #RADAR GRAFİK YAPIYOR, BAR GRAFİKLE BİRLİKTE KULLANILİYOR GENELDE
  scale_fill_ordinal()+
  theme_minimal()+
  labs(x= NULL , y= NULL)
```



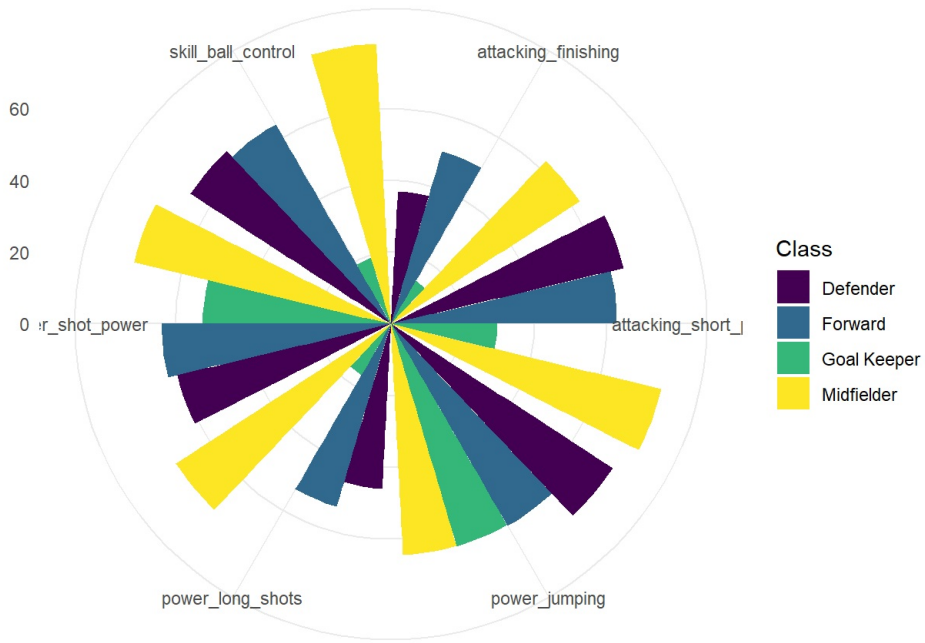
TABLO YORUMU:

Forvet sayısı olarak seria A > Laliga > Bundesliga > Süper Lig olarak sıralanabilir. Belki şöyle yorumlar yapılabilir; Forvet sakatlanmaları fazla olduğu için Serie A 'da fazla forvet bulunuyor denilebilir. kaleci sayıları eşitken Süper Lig yine eşitliği bozmaktadır.

SORU: Pozisyon sınıfına göre Serie A liginin futbolcuların ortalama özet istatistiklerinin grafiğini yazdırınız.

```
df %>%
  filter(League == "Serie A") %>%
  select(Class, attacking_finishing, attacking_short_passing , power_shot_power , power_long_shots , power_jumping
, skill_ball_control ) %>%
  group_by(Class) %>%
  summarise_at(vars(attacking_finishing:skill_ball_control), funs(mean)) %>%
  gather(variables , values, -Class) %>%
  ggplot(aes(variables, values , fill = Class)) +
  geom_col(position = "dodge")+
  coord_polar()+
  scale_fill_ordinal()+
  theme_minimal()+
  labs(x =NULL , y = NULL)
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```



TABLO YORUMU:

SERİA A ligi için skill_ball_control ortalaması çoktan aza sıralaması; Midfielder > Forward > Defender > Goal Keeper şeklindedir.

SERİA A ligi için power_jumping ortalaması çoktan aza sıralaması; Goal Keeper > Midfielder > Forward > Defender şeklindedir.

SERİA A ligi için power_long_shots ortalaması çoktan aza sıralaması; Midfielder > Forward > Defender > Goal Keeper şeklindedir.