Plots

Libraries and Data

```
load.libraries <- c('data.table', 'testthat', 'gridExtra', 'corrplot', 'GGally', 'ggplot2', 'e1071', 'dp
install.lib <- load.libraries[!load.libraries %in% installed.packages()]
for(libs in install.lib) install.packages(libs, dependences = TRUE)
sapply(load.libraries, require, character = TRUE)
```

```
## Loading required package: data.table

## Loading required package: testthat

## Loading required package: gridExtra

## Loading required package: corrplot

## corrplot 0.85 loaded

## Loading required package: GGally

## Loading required package: ggplot2

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

## Loading required package: e1071

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:testthat':
##
##     matches

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## data.table   testthat  gridExtra   corrplot     GGally    ggplot2      e1071
##       TRUE       TRUE       TRUE       TRUE       TRUE       TRUE       TRUE
##      dplyr
##       TRUE
```

```r
library(data.table)
library(ggplot2) #data visualization
library(plotly) #interactive data visualization
```

```
##
## 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
```

```r
library(psych) #correlation visualization helping
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

## The following object is masked from 'package:testthat':
##
##     describe
```

```r
library(rattle) #graphing decesiion trees
```

```
## Loading required package: tibble

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```r
library(caret) # machine learning
```

```
## Loading required package: lattice
```

```r
library(tree)
library(e1071)
library(rpart)
library(magrittr) # needs to be run every time you start R and want to use %>%
```

```
##
## Attaching package: 'magrittr'

## The following objects are masked from 'package:testthat':
##
##     equals, is_less_than, not
```

```r
library(dplyr)    # alternatively, this also loads %>%
library(class)
library(formattable)
```

```
##
## Attaching package: 'formattable'

## The following object is masked from 'package:plotly':
##
##     style
```

```r
library(funModeling)
```

```
## Loading required package: Hmisc

## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following object is masked from 'package:psych':
##
##     describe

## The following object is masked from 'package:plotly':
##
##     subplot

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following object is masked from 'package:e1071':
##
##     impute

## The following object is masked from 'package:testthat':
##
##     describe

## The following objects are masked from 'package:base':
##
##     format.pval, units

## Registered S3 method overwritten by 'cli':
##   method     from
##   print.tree tree

## funModeling v.1.9.4 :)
## Examples and tutorials at livebook.datascienceheroes.com
##  / Now in Spanish: librovivodecienciadedatos.ai
```

```
##
## Attaching package: 'funModeling'

## The following object is masked from 'package:GGally':
##
##     range01
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x psych::%+%()             masks ggplot2::%+%()
## x psych::alpha()           masks ggplot2::alpha()
## x dplyr::between()         masks data.table::between()
## x dplyr::combine()         masks gridExtra::combine()
## x magrittr::equals()       masks testthat::equals()
## x tidyr::extract()         masks magrittr::extract()
## x plotly::filter()         masks dplyr::filter(), stats::filter()
## x dplyr::first()           masks data.table::first()
## x magrittr::is_less_than() masks testthat::is_less_than()
## x purrr::is_null()         masks testthat::is_null()
## x dplyr::lag()             masks stats::lag()
## x dplyr::last()            masks data.table::last()
## x purrr::lift()            masks caret::lift()
## x tidyr::matches()         masks dplyr::matches(), testthat::matches()
## x magrittr::not()          masks testthat::not()
## x purrr::set_names()       masks magrittr::set_names()
## x Hmisc::src()             masks dplyr::src()
## x Hmisc::summarize()       masks dplyr::summarize()
## x purrr::transpose()       masks data.table::transpose()
```

```r
library(Hmisc)
data <- read.csv("data.csv")
setDT(data)
```

Missing Value plot

```r
plot_Missing <- function(data_in, title = NULL){
  temp_df <- as.data.frame(ifelse(is.na(data_in), 0, 1))
  temp_df <- temp_df[,order(colSums(temp_df))]
  data_temp <- expand.grid(list(x = 1:nrow(temp_df), y = colnames(temp_df)))
  data_temp$m <- as.vector(as.matrix(temp_df))
  data_temp <- data.frame(x = unlist(data_temp$x), y = unlist(data_temp$y), m = unlist(data_temp$m))
  ggplot(data_temp) + geom_tile(aes(x=x, y=y, fill=factor(m))) + scale_fill_manual(values=c("white", "bl
}
```

Selected features covering post 2010

```r
df1 <- data %>%
  select(date, Winner, title_bout, weight_class,B_fighter, B_Height_cms, B_Reach_cms, B_age, B_current_
         R_fighter, R_Height_cms, R_Reach_cms, R_age,
         R_current_lose_streak, R_current_win_streak,R_longest_win_streak, R_losses,R_wins,R_total_round
         R_total_title_bouts,R_win_by_KO.TKO,R_win_by_Submission,
         R_win_by_Decision_Majority,R_win_by_Decision_Split,R_win_by_Decision_Unanimous,R_win_by_TKO_Doc


df1 <- subset.data.frame(df1, subset= date >= "2010-01-01")
```

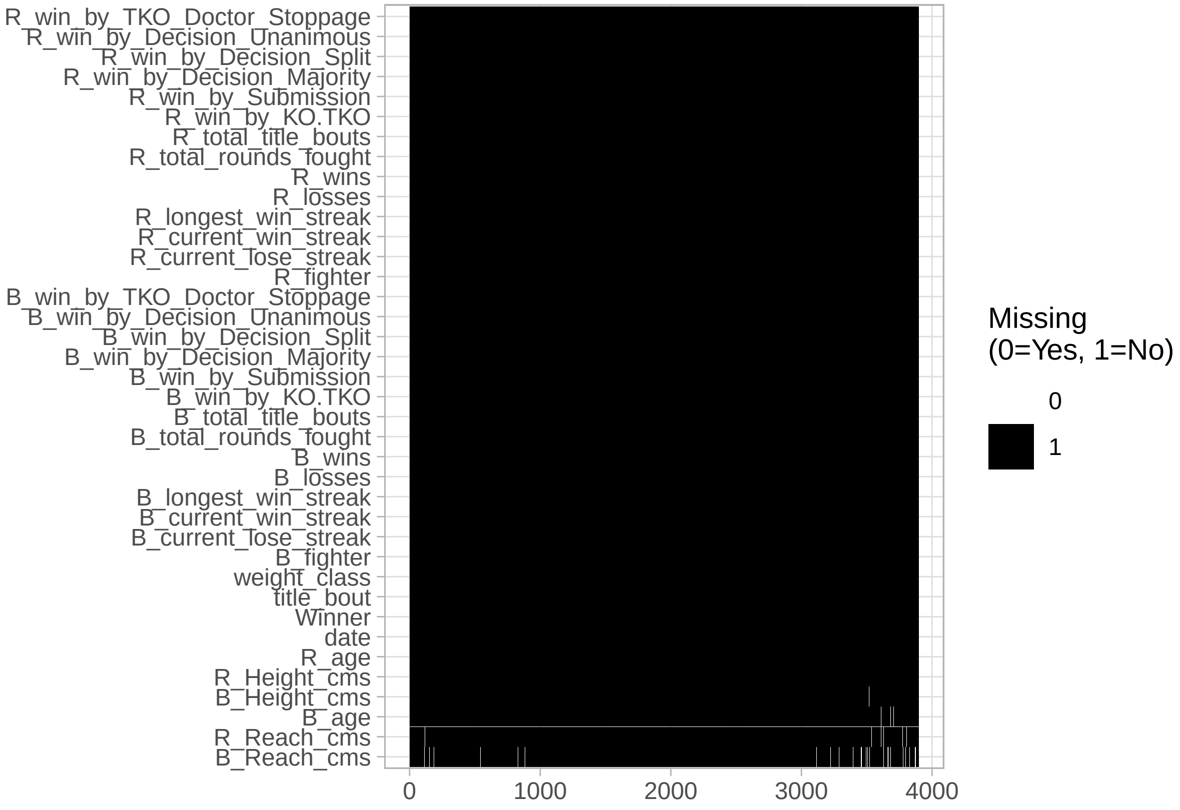
Dimension of new dataset

```
dim(df1)
```

```
## [1] 3897   38
```

Null Value Plot

```
plot_Missing(df1[,colSums(is.na(df1)) >= 0, with = FALSE])
```



### Detection of null values

```
cat_var1 <- names(df1)[which(sapply(df1, is.character))] #kategorik
numeric_var1 <- names(df1)[which(sapply(df1, is.numeric))] #numeric
colSums(sapply(df1[,.SD, .SDcols = cat_var1], is.na))
```

```
##         date       Winner   title_bout weight_class    B_fighter    R_fighter
##            0            0            0            0            0            0
```

```
colSums(sapply(df1[,.SD, .SDcols = numeric_var1], is.na)) #numericte null kontrolu
```

```
##                  B_Height_cms                   B_Reach_cms
##                             2                            97
##                        B_age        B_current_lose_streak
##                            7                            0
##         B_current_win_streak         B_longest_win_streak
##                            0                            0
##                     B_losses                       B_wins
##                            0                            0
##        B_total_rounds_fought          B_total_title_bouts
##                            0                            0
##              B_win_by_KO.TKO          B_win_by_Submission
##                            0                            0
##   B_win_by_Decision_Majority      B_win_by_Decision_Split
##                            0                            0
##  B_win_by_Decision_Unanimous B_win_by_TKO_Doctor_Stoppage
##                            0                            0
##                 R_Height_cms                  R_Reach_cms
##                            2                           39
##                        R_age        R_current_lose_streak
##                            2                            0
##         R_current_win_streak         R_longest_win_streak
##                            0                            0
##                     R_losses                       R_wins
##                            0                            0
##        R_total_rounds_fought          R_total_title_bouts
##                            0                            0
##              R_win_by_KO.TKO          R_win_by_Submission
##                            0                            0
##   R_win_by_Decision_Majority      R_win_by_Decision_Split
##                            0                            0
##  R_win_by_Decision_Unanimous R_win_by_TKO_Doctor_Stoppage
##                            0                            0
```

###New dataset where null values are deleted

```
df2 <- na.omit(df1) ##null rowlari sildi

cat_var2 <- names(df2)[which(sapply(df2, is.character))] #kategorik
numeric_var2 <- names(df2)[which(sapply(df2, is.numeric))] #numeric
colSums(sapply(df2[,.SD, .SDcols = cat_var2], is.na)) #kategorikte null kontrolu
```

```
##         date       Winner   title_bout weight_class    B_fighter    R_fighter
##            0            0            0            0            0            0
```

```
colSums(sapply(df2[,.SD, .SDcols = numeric_var2], is.na)) #numericte null kontrolu
```

```
##                 B_Height_cms                  B_Reach_cms
##                            0                            0
##                        B_age        B_current_lose_streak
##                            0                            0
##         B_current_win_streak         B_longest_win_streak
##                            0                            0
##                     B_losses                       B_wins
##                            0                            0
##        B_total_rounds_fought          B_total_title_bouts
##                            0                            0
##              B_win_by_KO.TKO          B_win_by_Submission
##                            0                            0
##   B_win_by_Decision_Majority      B_win_by_Decision_Split
##                            0                            0
##  B_win_by_Decision_Unanimous B_win_by_TKO_Doctor_Stoppage
##                            0                            0
##                 R_Height_cms                  R_Reach_cms
##                            0                            0
##                        R_age        R_current_lose_streak
##                            0                            0
##         R_current_win_streak         R_longest_win_streak
##                            0                            0
##                     R_losses                       R_wins
##                            0                            0
##        R_total_rounds_fought         R_total_title_bouts
##                            0                            0
##               R_win_by_KO.TKO         R_win_by_Submission
##                            0                            0
##   R_win_by_Decision_Majority      R_win_by_Decision_Split
##                            0                            0
##  R_win_by_Decision_Unanimous R_win_by_TKO_Doctor_Stoppage
##                            0                            0
```

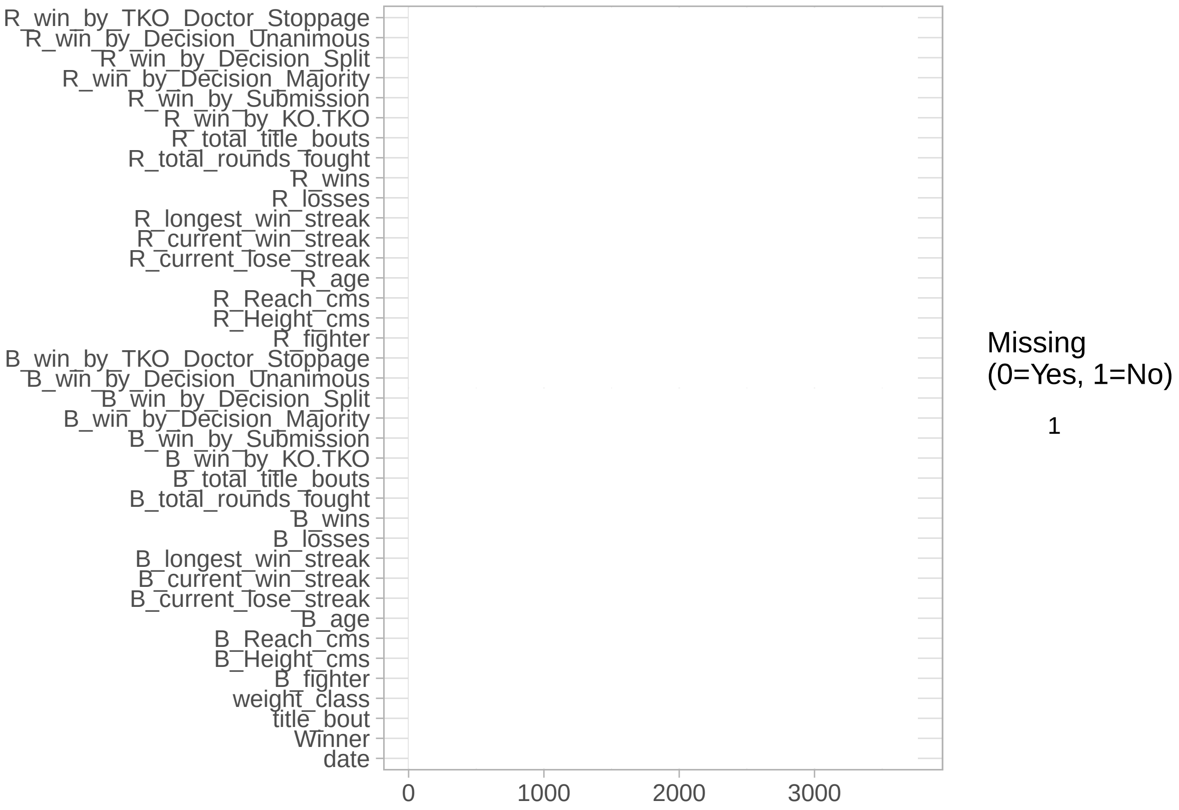
Dimension of final dataset

```
dim(df1)
```

```
## [1] 3897   38
```

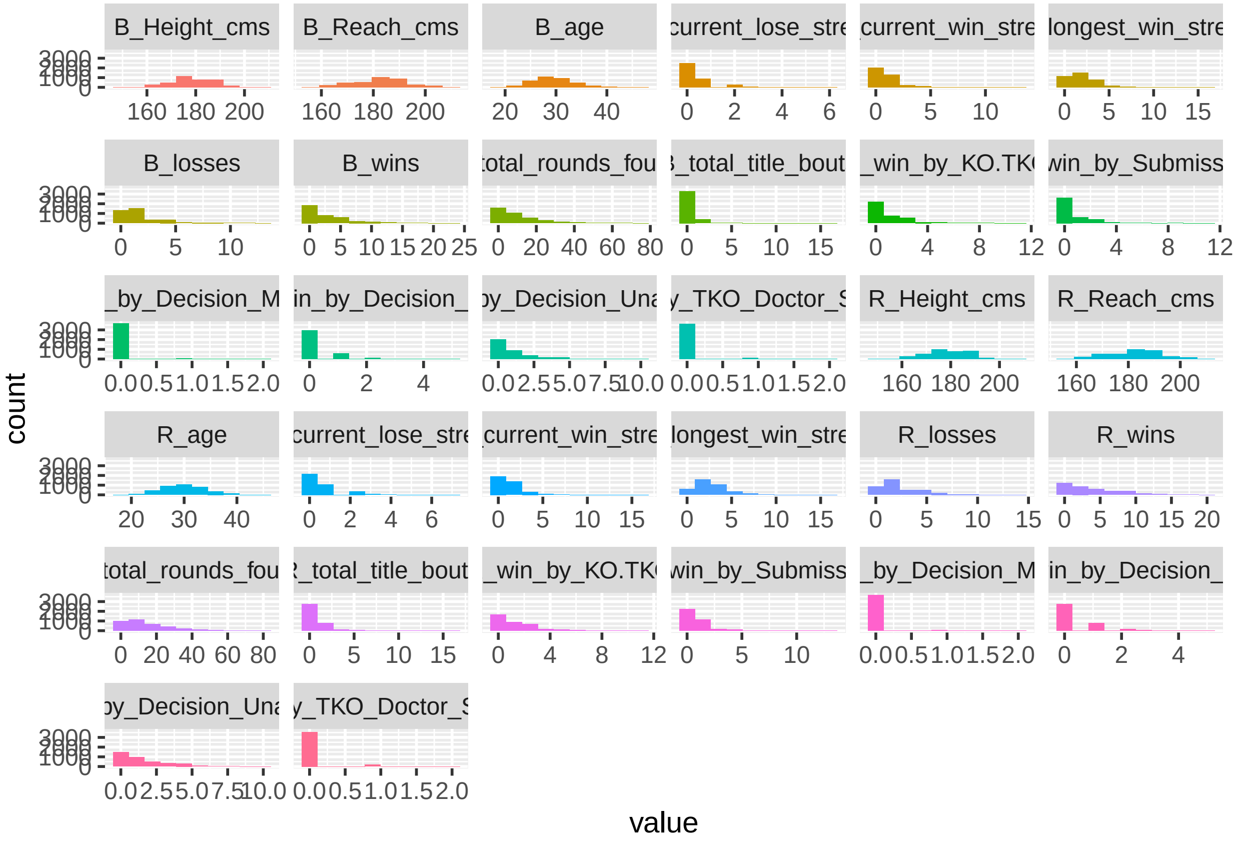
Null Value Plot of final dataset

```
plot_Missing(df2[,colSums(is.na(df2)) >= 0, with = FALSE])
```



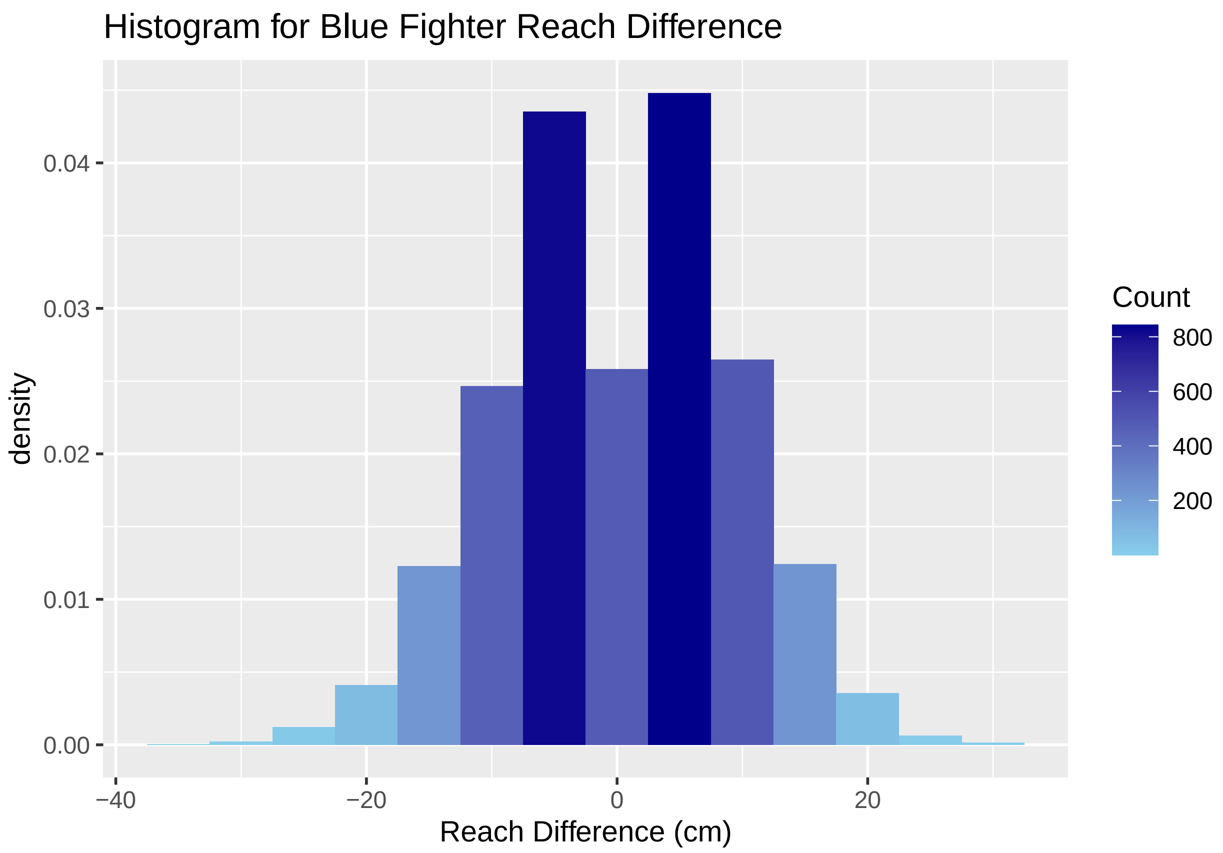
### Visualization of numeric column information

```
plot_num(df2)
```



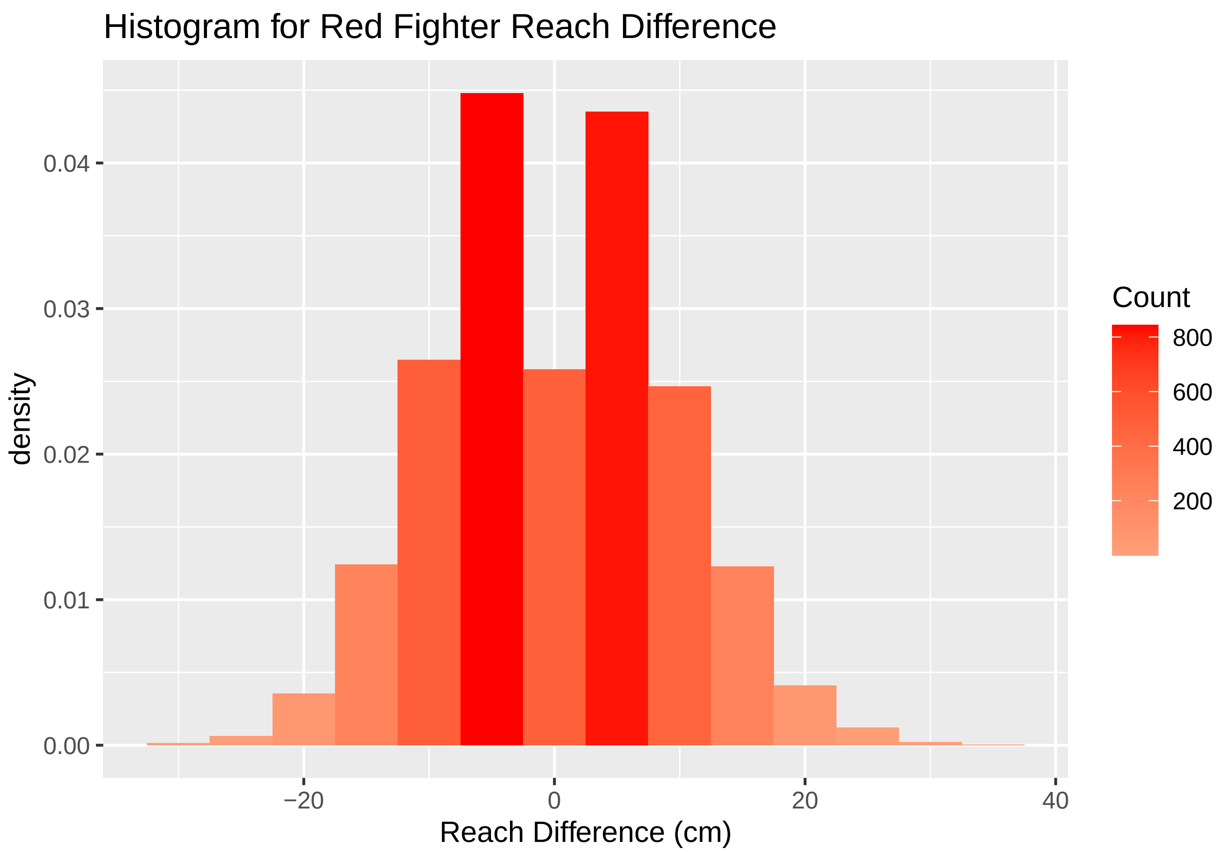
### Histogram for Blue Fighter Reach Difference

```
ggplot(df2, aes(x=(B_Reach_cms - R_Reach_cms), y =..density.., fill=..count..)) + geom_histogram(binwid
```



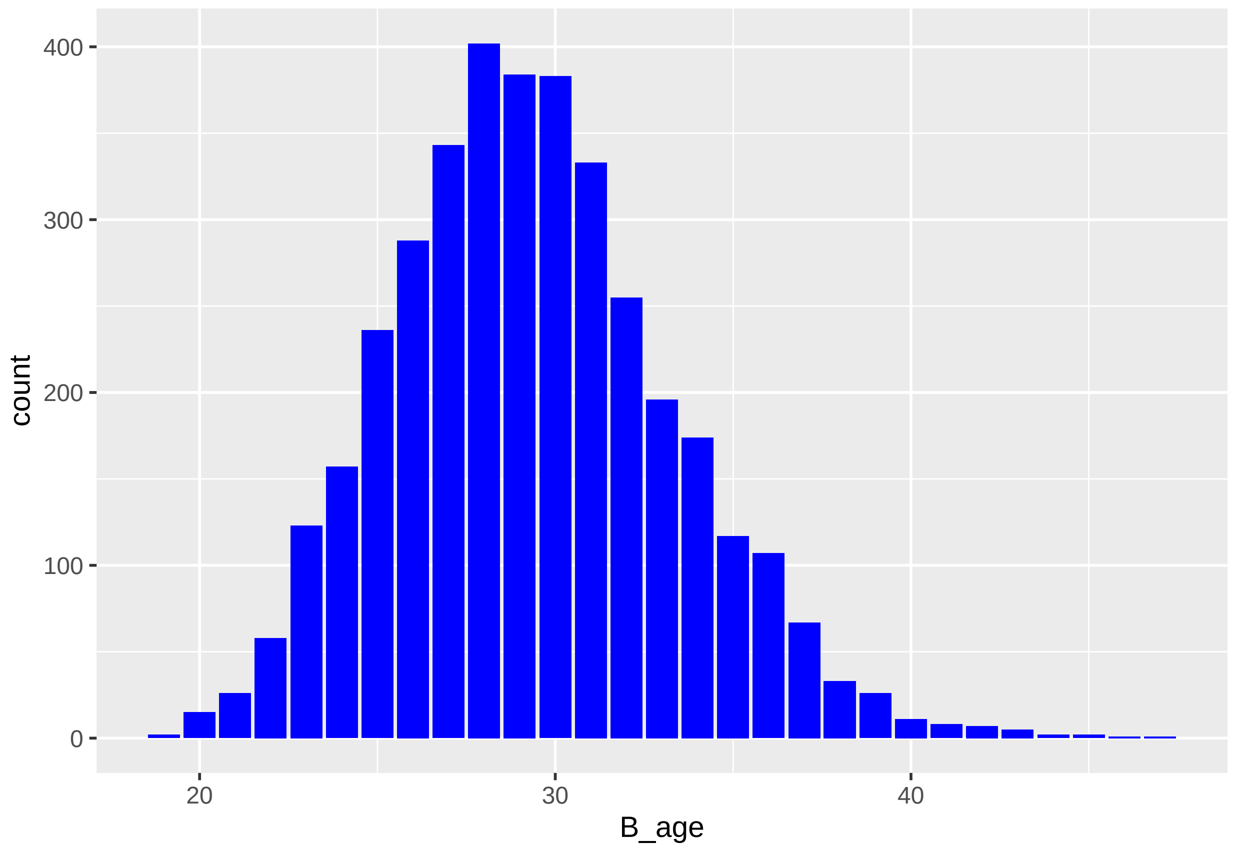
### Histogram for Red Fighter Reach Difference

```
ggplot(df2, aes(x=(R_Reach_cms - B_Reach_cms), y =..density.., fill=..count..)) + geom_histogram(binwid
```



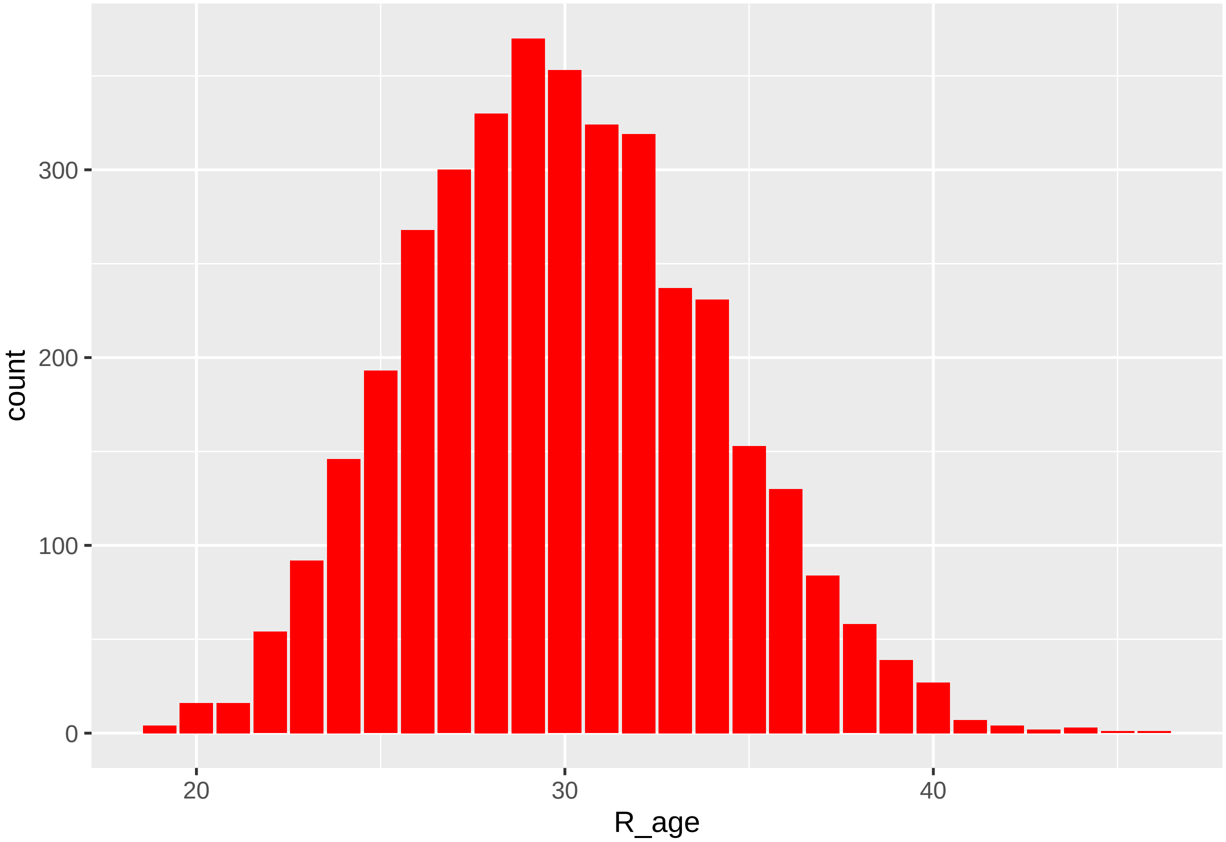
### Barplot of Blue Fighter Age

```
ggplot(df2, aes(x = B_age)) + geom_bar(fill = "#0000FF") #B_age
```



### Barplot of Blue Fighter Age

```
ggplot(df2, aes(x = R_age)) + geom_bar(fill = "#FF0000") #R_age
```



List of Blue fighter's winning average

```
temp <- df2 %>% select(B_fighter,B_wins)
temp <- temp %>% group_by(B_fighter) %>% summarise(avg=mean(B_wins))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
temp <- temp %>% arrange(desc(avg))
temp <- temp[1:10,]
temp %>%
  formattable(list(avg = color_bar("#85C1E9")), align = 'l')
```

B_fighter

avg

Georges St-Pierre

19.00000

Anderson Silva

16.50000

Randy Couture

16.00000

Frank Mir

15.00000

Tito Ortiz

15.00000

Diego Sanchez

14.20000

Jim Miller

14.00000

Josh Koscheck

14.00000

Michael Bisping

13.83333

Andrei Arlovski

13.33333

List of Red fighter's winning average

```r
temp <- df2 %>% select(R_fighter,R_wins)
temp <- temp %>% group_by(R_fighter) %>% summarise(avg=mean(R_wins))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
temp <- temp %>% arrange(desc(avg))
temp <- temp[1:10,]
temp %>%
  formattable(list(avg = color_bar("#FF0000")), align = 'l')
```

R_fighter

avg

Matt Hughes

17.66667

Chuck Liddell

16.00000

Georges St-Pierre

15.50000

Andrei Arlovski

14.33333

Tito Ortiz

14.33333

Josh Koscheck

14.20000

Randy Couture

14.00000

Rich Franklin

14.00000

Michael Bisping

13.92308

Anderson Silva

13.90000

##The winning Blue fighter according to weight_class

```
df2 %>% filter(Winner == "Blue") %>% count(weight_class) #weight_class'a göre kazanan blue
```

```
##              weight_class   n
##  1:          Bantamweight 152
##  2:          Catch Weight   5
##  3:         Featherweight 171
##  4:             Flyweight  69
##  5:           Heavyweight 117
##  6:     Light Heavyweight 132
##  7:           Lightweight 285
##  8:          Middleweight 200
##  9:          Welterweight 296
## 10:  Women's Bantamweight  46
## 11: Women's Featherweight   6
## 12:     Women's Flyweight  20
## 13:   Women's Strawweight  52
```

##The winning Red fighter according to weight_class

```
df2 %>% filter(Winner == "Red") %>% count(weight_class) #weight_class'a göre kazanan red
```

```
##              weight_class   n
##  1:          Bantamweight 210
##  2:          Catch Weight  11
##  3:         Featherweight 242
##  4:             Flyweight 112
##  5:           Heavyweight 169
##  6:     Light Heavyweight 174
##  7:           Lightweight 405
##  8:          Middleweight 266
##  9:          Welterweight 378
## 10:  Women's Bantamweight  63
## 11: Women's Featherweight   4
## 12:     Women's Flyweight  25
## 13:   Women's Strawweight  86
```

Splitting columns containing numeric data

```
numeric_data <- select_if(df2, is.numeric)
summary(numeric_data)
```

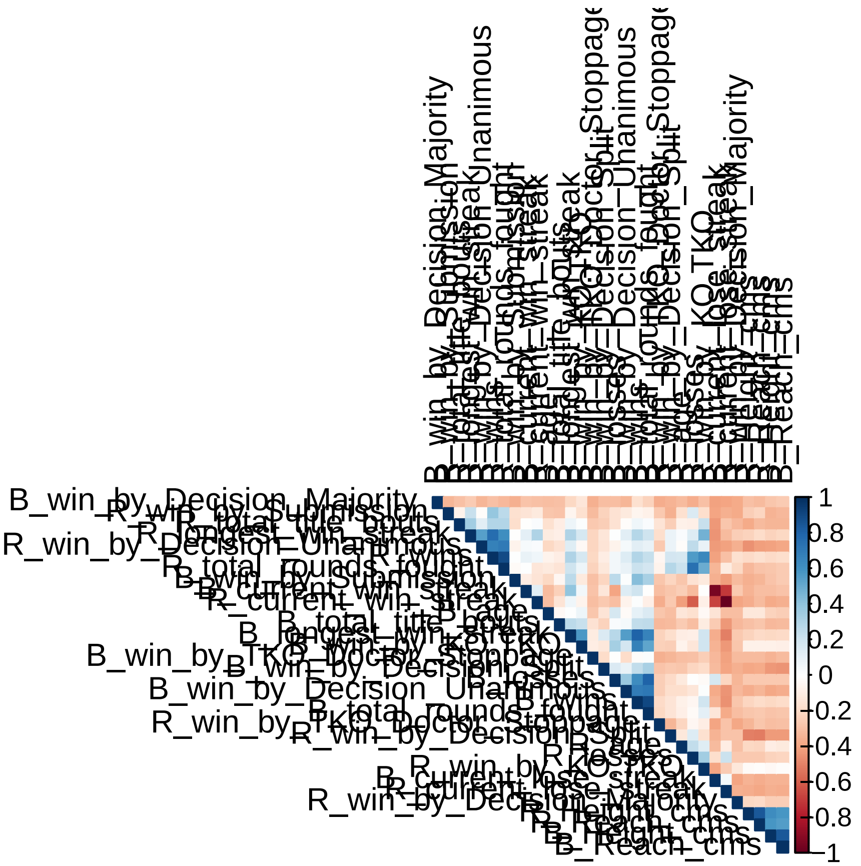
```
##   B_Height_cms    B_Reach_cms        B_age       B_current_lose_streak
##  Min.   :152.4   Min.   :152.4   Min.   :19.00   Min.   :0.0000
##  1st Qu.:172.7   1st Qu.:177.8   1st Qu.:27.00   1st Qu.:0.0000
##  Median :177.8   Median :182.9   Median :29.00   Median :0.0000
##  Mean   :178.4   Mean   :182.8   Mean   :29.35   Mean   :0.4572  
##  3rd Qu.:185.4   3rd Qu.:190.5   3rd Qu.:32.00   3rd Qu.:1.0000  
##  Max.   :210.8   Max.   :213.4   Max.   :47.00   Max.   :6.0000  
##  B_current_win_streak B_longest_win_streak    B_losses         B_wins      
##  Min.   : 0.0000      Min.   : 0.000       Min.   : 0.00   Min.   : 0.000  
##  1st Qu.: 0.0000      1st Qu.: 0.000       1st Qu.: 0.00   1st Qu.: 0.000  
##  Median : 0.0000      Median : 1.000       Median : 1.00   Median : 2.000  
##  Mean   : 0.8892      Mean   : 1.729       Mean   : 1.67   Mean   : 2.778  
##  3rd Qu.: 1.0000      3rd Qu.: 3.000       3rd Qu.: 2.00   3rd Qu.: 4.000  
##  Max.   :13.0000      Max.   :16.000       Max.   :13.00   Max.   :23.000  
##  B_total_rounds_fought B_total_title_bouts B_win_by_KO.TKO 
##  Min.   : 0.00         Min.   : 0.0000     Min.   : 0.0000  
##  1st Qu.: 2.00         1st Qu.: 0.0000     1st Qu.: 0.0000  
##  Median : 6.00         Median : 0.0000     Median : 0.0000  
##  Mean   :10.32         Mean   : 0.2384     Mean   : 0.9479  
##  3rd Qu.:15.00         3rd Qu.: 0.0000     3rd Qu.: 1.0000  
##  Max.   :75.00         Max.   :16.0000     Max.   :11.0000  
##  B_win_by_Submission B_win_by_Decision_Majority B_win_by_Decision_Split
##  Min.   : 0.0000     Min.   :0.00000            Min.   :0.0000         
##  1st Qu.: 0.0000     1st Qu.:0.00000            1st Qu.:0.0000         
##  Median : 0.0000     Median :0.00000            Median :0.0000         
##  Mean   : 0.5747     Mean   :0.01489            Mean   :0.2634         
##  3rd Qu.: 1.0000     3rd Qu.:0.00000            3rd Qu.:0.0000         
##  Max.   :11.0000     Max.   :2.00000            Max.   :5.0000         
##  B_win_by_Decision_Unanimous B_win_by_TKO_Doctor_Stoppage  R_Height_cms  
##  Min.   : 0.000              Min.   :0.00000              Min.   :152.4  
##  1st Qu.: 0.000              1st Qu.:0.00000              1st Qu.:172.7  
##  Median : 0.000              Median :0.00000              Median :177.8  
##  Mean   : 0.932              Mean   :0.03934              Mean   :178.3  
##  3rd Qu.: 1.000              3rd Qu.:0.00000              3rd Qu.:185.4  
##  Max.   :10.000              Max.   :2.00000              Max.   :210.8  
##   R_Reach_cms        R_age       R_current_lose_streak R_current_win_streak
##  Min.   :152.4   Min.   :19.00   Min.   :0.0000        Min.   : 0.000      
##  1st Qu.:175.3   1st Qu.:27.00   1st Qu.:0.0000        1st Qu.: 0.000      
##  Median :182.9   Median :30.00   Median :0.0000        Median : 0.000      
##  Mean   :182.8   Mean   :29.94   Mean   :0.6058        Mean   : 1.041      
##  3rd Qu.:190.5   3rd Qu.:33.00   3rd Qu.:1.0000        3rd Qu.: 1.000      
##  Max.   :213.4   Max.   :46.00   Max.   :7.0000        Max.   :16.000      
##  R_longest_win_streak    R_losses          R_wins       R_total_rounds_fought
##  Min.   : 0.000       Min.   : 0.000   Min.   : 0.000   Min.   : 0.00        
##  1st Qu.: 1.000       1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 4.00        
##  Median : 2.000       Median : 2.000   Median : 3.000   Median :11.00        
##  Mean   : 2.472       Mean   : 2.279   Mean   : 4.088   Mean   :15.11        
##  3rd Qu.: 4.000       3rd Qu.: 3.000   3rd Qu.: 6.000   3rd Qu.:22.00        
##  Max.   :16.000       Max.   :14.000   Max.   :20.000   Max.   :80.00        
##  R_total_title_bouts R_win_by_KO.TKO  R_win_by_Submission
##  Min.   : 0.0000     Min.   : 0.000   Min.   : 0.0000    
##  1st Qu.: 0.0000     1st Qu.: 0.000   1st Qu.: 0.0000    
##  Median : 0.0000     Median : 1.000   Median : 0.0000    
##  Mean   : 0.6002     Mean   : 1.392   Mean   : 0.8437    
##  3rd Qu.: 1.0000     3rd Qu.: 2.000   3rd Qu.: 1.0000    
##  Max.   :16.0000     Max.   :11.000   Max.   :13.0000    
##  R_win_by_Decision_Majority R_win_by_Decision_Split R_win_by_Decision_Unanimous
##  Min.   :0.00000            Min.   :0.0000          Min.   : 0.000             
##  1st Qu.:0.00000            1st Qu.:0.0000          1st Qu.: 0.000
##  Median :0.00000            Median :0.0000          Median : 1.000
##  Mean   :0.02632            Mean   :0.3503          Mean   : 1.408
##  3rd Qu.:0.00000            3rd Qu.:1.0000          3rd Qu.: 2.000
##  Max.   :2.00000            Max.   :5.0000          Max.   :10.000
##  R_win_by_TKO_Doctor_Stoppage
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.05981
##  3rd Qu.:0.00000
##  Max.   :2.00000
```

Correlation Matrix

- *Korelasyonun büyüklüğü (0-1) iki değişken arasındaki ilişkinin gücünü gösterirken işareti (+,-) değişkenlerin aynı yönde (+) artıp azaldığını ya da zıt yönlerde (-) artış ve azalış gösterdiğini belirtir. Eğer iki değişken arasında hiç ilişki yoksa korelasyon katsayısı sıfır ya da sıfıra yakın bulunur. • Eğer iki değişken birbiriyle yüzde yüz oranında ilişkili ise korelasyon maksimum (1) değeri (mükemmel ilişki) alır. r<0.20 ve sıfıra yakın değerler ilişkinin olmadığı ya da çok zayıf ilişkiyi işaret eder. • 0.20-0.39 arasında ise zayıf ilişki • 0.40-0.59 arasında ise orta düzeyde ilişki • 0.60-0.79 arasında ise yüksek düzeyde ilişki • 0.80-1.0 ise çok yüksek ilişki olduğu yorumu yapılır.*

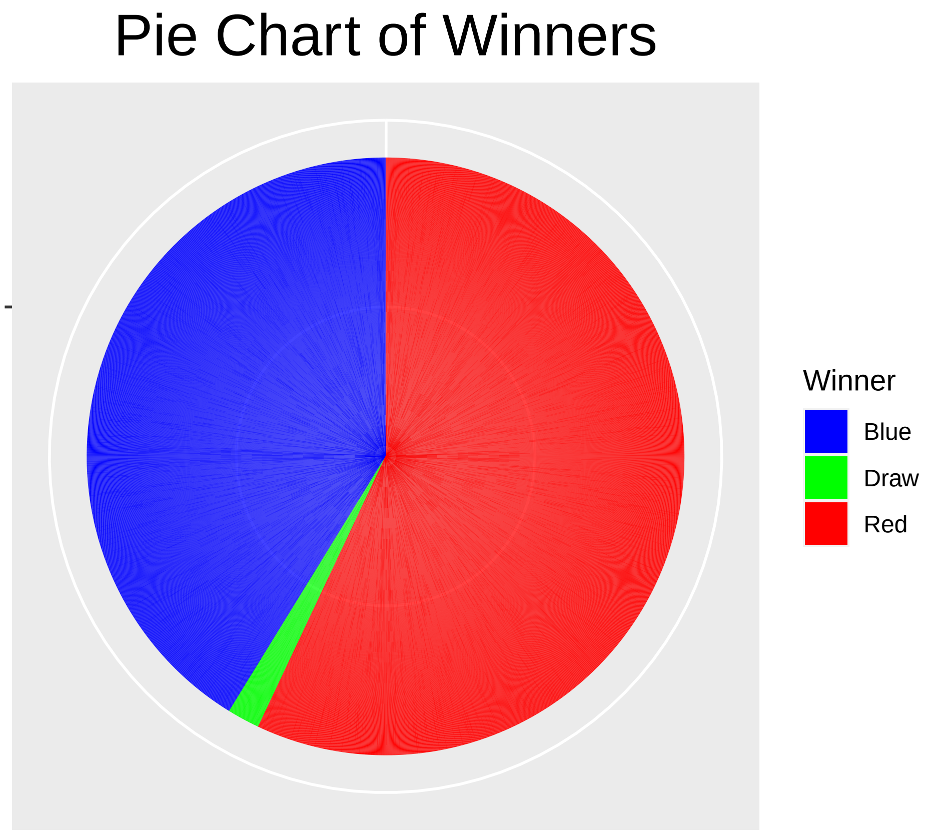
+1,00 a yaklaştıkça iki değişken arasında aynı yöndeki ilişki artar.Değişkenlerden biri artarken diğeri de artar. -1,00 a yaklaştıkça iki değişen arasında ters yönde ilişki artar. Değişkenlerden biri artarken diğeri azalır. 0,00'a yaklaştıkça iki değişken arasındaki ilişki azalır.

```r
cor_data <- cor(numeric_data)
corrplot(cor_data, method = "color", type = "upper", tl.col = "black", order="hclust")
```



### Pie chart showing the winning fighter

```
custom_col <- c("blue", "green", "red")
ggplot(df2, aes(x = "", y="", fill = factor(Winner))) +
  geom_bar(width = 1, stat = "identity") +
  theme(axis.line = element_blank(),
        plot.title = element_text(hjust=0.5,size=22)) +
  labs(fill="Winner",
       x=NULL,
       y=NULL,
       title="Pie Chart of Winners") + coord_polar(theta = "y", start=0)+
  scale_fill_manual(values=custom_col)
```


Pie Chart of Winners
Winner
Blue
Draw
Red