UFC RESEARCH QUESTION 1

Due 8th December, 2024

```
ufc <- read.csv("ufc-master.csv")

# View the data
head(ufc)
```

```
# data cleaning cell
colSums(is.na(ufc))
```

```
                RedFighter                BlueFighter
                         0                          0
                   RedOdds                   BlueOdds
                       219                        219
           RedExpectedValue          BlueExpectedValue
                       219                        219
                      Date                   Location
                         0                          0
                   Country                     Winner
                         0                          0
                TitleBout                  WeightClass
                         0                          0
                    Gender              NumberOfRounds
                         0                          0
      BlueCurrentLoseStreak       BlueCurrentWinStreak
                         0                          0
                BlueDraws         BlueAvgSigStrLanded
                         0                        930
          BlueAvgSigStrPct               BlueAvgSubAtt
                       765                        832
           BlueAvgTDLanded                BlueAvgTDPct
                       833                        842
```

BlueLongestWinStreak	BlueLosses
0	0
BlueTotalRoundsFought	BlueTotalTitleBouts
0	0
BlueWinsByDecisionMajority	BlueWinsByDecisionSplit
0	0
BlueWinsByDecisionUnanimous	BlueWinsByKO
0	0
BlueWinsBySubmission	BlueWinsByTKODoctorStoppage
0	0
BlueWins	BlueStance
0	0
BlueHeightCms	BlueReachCms
0	0
BlueWeightLbs	RedCurrentLoseStreak
0	0
RedCurrentWinStreak	RedDraws
0	0
RedAvgSigStrLanded	RedAvgSigStrPct
455	357
RedAvgSubAtt	RedAvgTDLanded
357	357
RedAvgTDPct	RedLongestWinStreak
367	0
RedLosses	RedTotalRoundsFought
0	0
RedTotalTitleBouts	RedWinsByDecisionMajority
0	0
RedWinsByDecisionSplit	RedWinsByDecisionUnanimous
0	0
RedWinsByKO	RedWinsBySubmission
0	0
RedWinsByTKODoctorStoppage	RedWins
0	0
RedStance	RedHeightCms
0	0
RedReachCms	RedWeightLbs
0	0
RedAge	BlueAge
0	0
LoseStreakDif	WinStreakDif
0	0
LongestWinStreakDif	WinDif

0	0
LossDif	TotalRoundDif
0	0
TotalTitleBoutDif	KODif
0	0
SubDif	HeightDif
0	0
ReachDif	AgeDif
0	0
SigStrDif	AvgSubAttDif
0	0
AvgTDDif	EmptyArena
0	1436
BMatchWCRank	RMatchWCRank
5289	4716
RWFlyweightRank	RWFeatherweightRank
6382	6469
RWStrawweightRank	RWBantamweightRank
6334	6324
RHeavyweightRank	RLightHeavyweightRank
6295	6296
RMiddleweightRank	RWelterweightRank
6296	6290
RLightweightRank	RFeatherweightRank
6295	6303
RBantamweightRank	RFlyweightRank
6299	6292
RPFPRank	BWFlyweightRank
6228	6406
BWFeatherweightRank	BWStrawweightRank
6477	6380
BWBantamweightRank	BHeavyweightRank
6371	6332
BLightHeavyweightRank	BMiddleweightRank
6360	6341
BWelterweightRank	BLightweightRank
6360	6359
BFeatherweightRank	BBantamweightRank
6355	6360
BFlyweightRank	BPFPRank
6348	6411
BetterRank	Finish
0	0

```
      FinishDetails                 FinishRound
                  0                         622
    FinishRoundTime           TotalFightTimeSecs
                  0                         622
        RedDecOdds                BlueDecOdds
               1077                        1107
          RSubOdds                   BSubOdds
               1326                        1350
           RKOOdds                    BKOOdds
               1324                        1351
```

```
# removing the data which has way too many missing values

ufc = subset(ufc, select = -c(BMatchWCRank, RMatchWCRank, RWFlyweightRank,
                              RWFeatherweightRank, RWStrawweightRank, RWBantamweightRank,
                              RHeavyweightRank, RLightHeavyweightRank, RMiddleweightRank,
                              RWelterweightRank, RLightweightRank, RFeatherweightRank,
                              RBantamweightRank, RFlyweightRank, RPFPRank, BWFlyweightRank,
                              BWFeatherweightRank, BWStrawweightRank, BWBantamweightRank,
                              BHeavyweightRank, BLightHeavyweightRank, BMiddleweightRank,
                              BWelterweightRank, BLightweightRank, BFeatherweightRank,
                              BBantamweightRank, BFlyweightRank, BPFPRank))

library(dplyr)
```

```
Attaching package: 'dplyr'

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

    filter, lag

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

    intersect, setdiff, setequal, union
```

```
# removing all missing value rows from the columns of interest
ufc_clean <- ufc %>%
  filter(
    !is.na(RedAvgSubAtt),
    !is.na(BlueAvgSubAtt),
```

```
    !is.na(BlueReachCms),
    !is.na(RedReachCms),
    !is.na(BlueAvgSigStrLanded),
    !is.na(RedAvgSigStrLanded),
    !is.na(TotalFightTimeSecs),
    !is.na(WeightClass)
  )
nrow(ufc_clean)
```

```
[1] 4895
```

```
filtered_ufc_blue <- ufc_clean[c("BlueReachCms","BlueAvgSigStrLanded",
                                 "WeightClass", "BlueHeightCms",
                                 "BlueCurrentWinStreak")]
colnames(filtered_ufc_blue) <- c("ReachCms", "AvgSigStrLanded", "WeightClass",
                                 "Height", "WinStreak")
filtered_ufc_red <- ufc_clean[c("RedReachCms","RedAvgSigStrLanded",
                                "WeightClass", "RedHeightCms", "RedCurrentWinStreak")]
colnames(filtered_ufc_red) <- c("ReachCms", "AvgSigStrLanded",
                                "WeightClass", "Height", "WinStreak")

# appending the two data sets
ufc_q1 <- rbind(filtered_ufc_blue, filtered_ufc_red)

# exclude outlier(one observation with 0 cm reach)
ufc_q1 <- ufc_q1[ufc_q1$ReachCms > 0,]
ufc_q1 <- ufc_q1[ufc_q1$AvgSigStrLanded > 0, ]

# check missing value: no missing
colSums(is.na(ufc_q1))
```

```
       ReachCms AvgSigStrLanded     WeightClass          Height       WinStreak
              0               0               0               0               0
```

```
# Log-transform the variables
ufc_q1$LogAvgSigStrLanded <- log(ufc_q1$AvgSigStrLanded)
ufc_q1$LogReachCms <- log(ufc_q1$ReachCms)
model_q1 <- lm(AvgSigStrLanded ~ ReachCms + WeightClass + Height + WinStreak,
               data = ufc_q1)
summary(model_q1)
```

```
Call:
lm(formula = AvgSigStrLanded ~ ReachCms + WeightClass + Height +
    WinStreak, data = ufc_q1)

Residuals:
    Min      1Q  Median      3Q     Max
-32.933 -16.378  -5.291  12.299 130.359

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                      72.069136   7.569175   9.521  < 2e-16 ***
ReachCms                         -0.293536   0.044100  -6.656 2.96e-11 ***
WeightClassCatch Weight         -10.877927   2.362018  -4.605 4.17e-06 ***
WeightClassFeatherweight          2.574233   0.883784   2.913  0.00359 **
WeightClassFlyweight             -2.177334   1.082890  -2.011  0.04439 *
WeightClassHeavyweight            3.866656   1.324240   2.920  0.00351 **
WeightClassLight Heavyweight      6.003263   1.236674   4.854 1.23e-06 ***
WeightClassLightweight            4.238484   0.842526   5.031 4.97e-07 ***
WeightClassMiddleweight           3.005096   1.062903   2.827  0.00470 **
WeightClassWelterweight           5.807064   0.946702   6.134 8.90e-10 ***
WeightClassWomen's Bantamweight  -1.066585   1.309501  -0.814  0.41538
WeightClassWomen's Featherweight -14.146466   3.414904  -4.143 3.46e-05 ***
WeightClassWomen's Flyweight     -11.644051   1.243821  -9.362  < 2e-16 ***
WeightClassWomen's Strawweight    -2.079263   1.207181  -1.722  0.08503 .
Height                           -0.004203   0.057011  -0.074  0.94124
WinStreak                         0.939495   0.121104   7.758 9.51e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 19.86 on 9719 degrees of freedom
Multiple R-squared:  0.03332,   Adjusted R-squared:  0.03183
F-statistic: 22.33 on 15 and 9719 DF,  p-value: < 2.2e-16
```

```r
# Load necessary libraries
library(car)          # For VIF
```

```
Loading required package: carData


Attaching package: 'car'
```

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

    recode
```

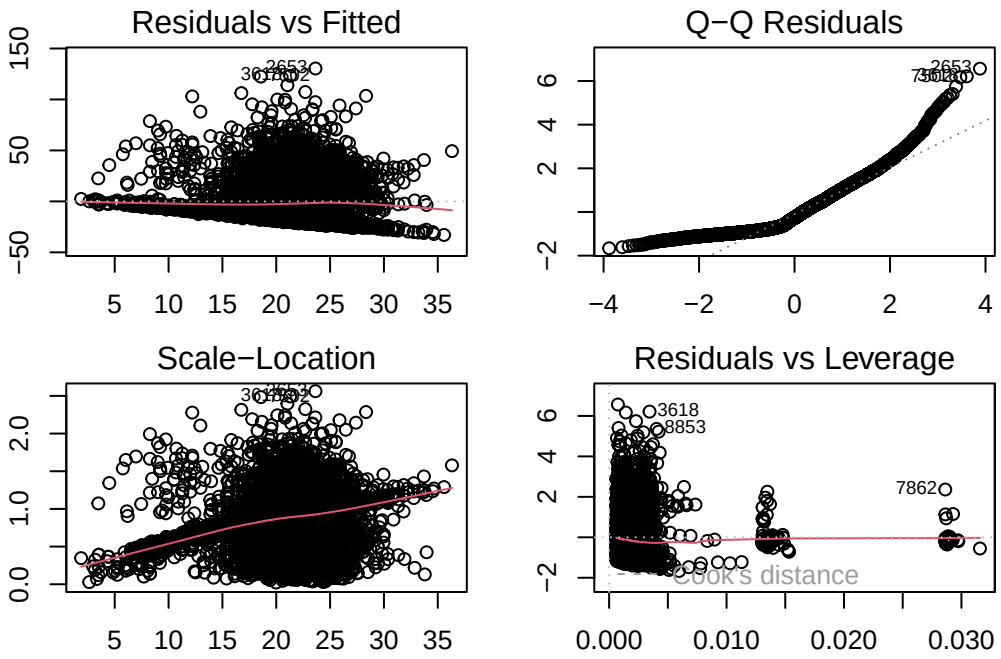
```
library(ggplot2)      # For residual plots

# 1. Check Variance Inflation Factor (VIF) for collinearity
vif_values <- vif(model_q1)
print("Variance Inflation Factor (VIF):")
```

```
[1] "Variance Inflation Factor (VIF):"
```

```
print(vif_values)
```

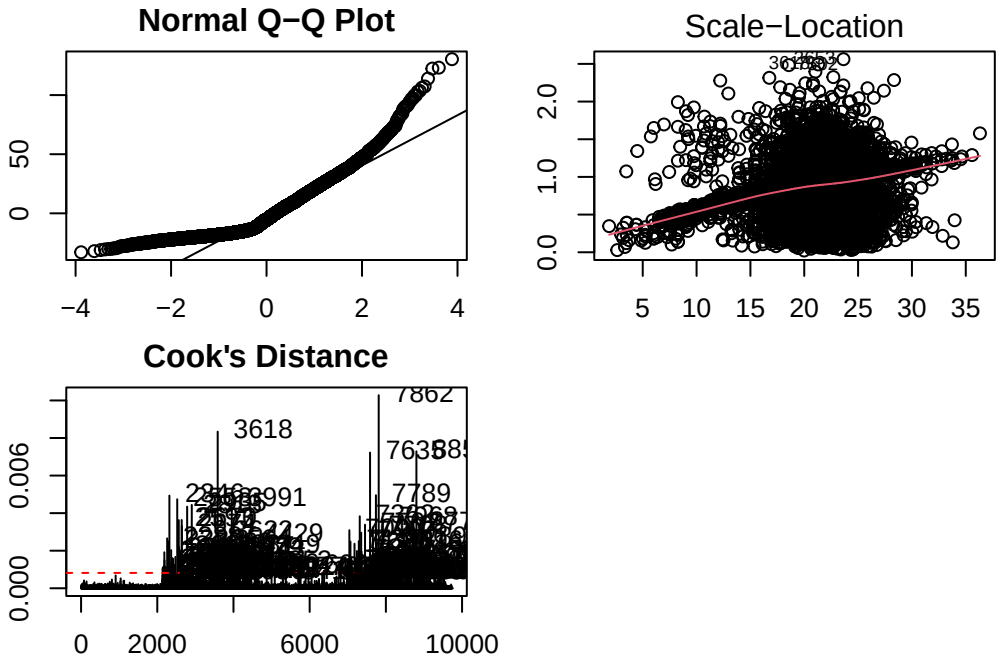
```
               GVIF Df GVIF^(1/(2*Df))
ReachCms    5.796169  1        2.407523
WeightClass 4.223489 12        1.061866
Height      6.625134  1        2.573934
WinStreak   1.004542  1        1.002268
```

```
# 2. Residuals vs Fitted Plot for Linearity
par(mfrow = c(2, 2), mar = c(2, 2, 2, 2))  # Set plotting layout
plot(model_q1)
```



```r
# 3. Normal Q-Q Plot for Normality of Residuals
qqnorm(residuals(model_q1))
qqline(residuals(model_q1))

# 4. Scale-Location Plot for Homoscedasticity
plot(model_q1, which = 3)

# 5. Check for influential points using Cook's Distance
cooksd <- cooks.distance(model_q1)
plot(cooksd, type = "h", main = "Cook's Distance", ylab = "Cook's Distance")

# Highlight observations with Cook's Distance > threshold
threshold <- 4 / nrow(ufc_clean)
influential <- which(cooksd > threshold)
abline(h = threshold, col = "red", lty = 2)
text(x = influential, y = cooksd[influential], labels = names(cooksd[influential]), pos = 4)

# 6. R-squared value
r_squared <- summary(model_q1)$r.squared
cat("R-squared:", r_squared, "\n")
```

```
R-squared: 0.033321
```



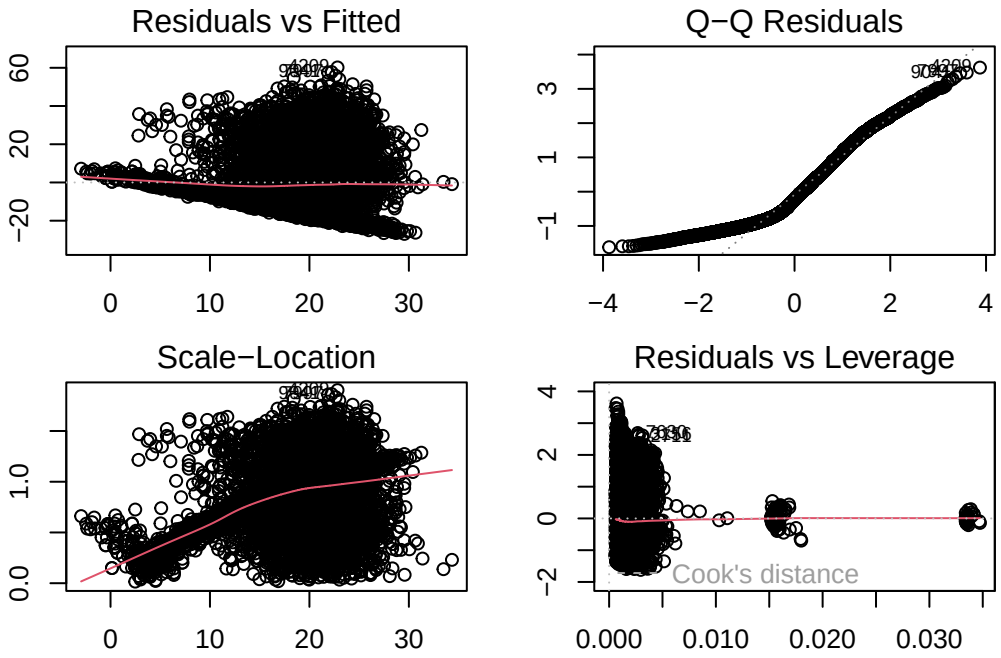
Linearity assumption is not being met as the residuals show a curved pattern or non-linearity.

Polynomial does not work

Removing influential points

OPTION: REMOVING INFLUENTIAL POINTS AND TAKING LOG

```
cooks_d <- cooks.distance(model_q1)
influential <- which(cooks_d > (4 / nrow(ufc_q1)))
ufc_q1_clean <- ufc_q1[-influential, ]

model_clean <- lm(AvgSigStrLanded ~ LogReachCms + WeightClass + log(Height) + WinStreak, data

par(mfrow = c(2, 2), mar = c(2,2,2,2))
plot(model_clean)
```



This proved somewhat helpful.

Linearity is still an issue.

Addressing the high VIF in the original model:

```r
model_simple <- lm(LogAvgSigStrLanded ~ LogReachCms + WeightClass + log(Height) + WinStreak,
vif(model_simple)
```

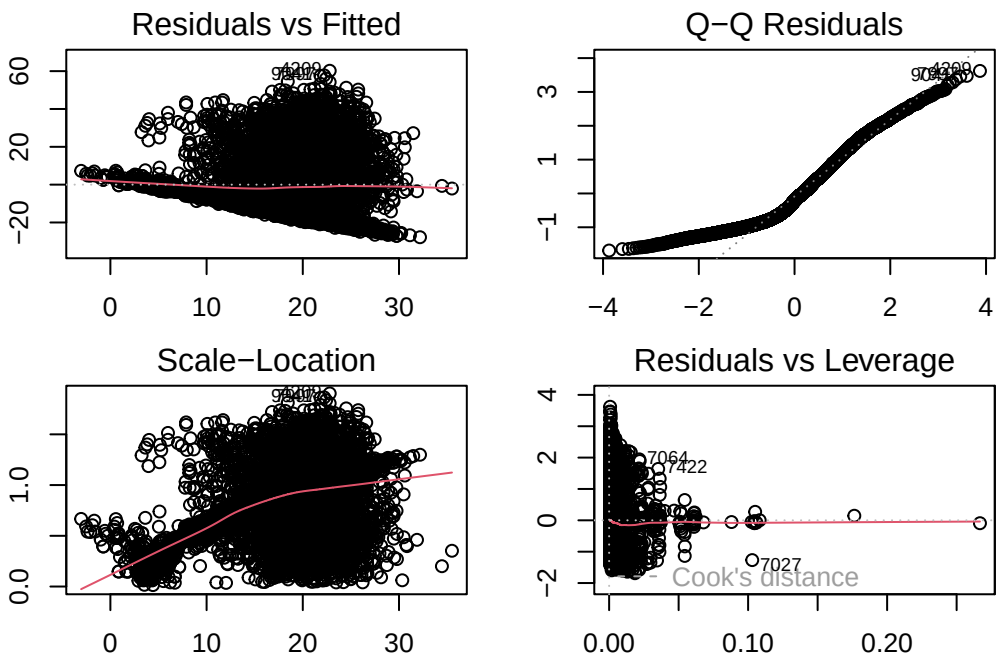
```
                GVIF Df GVIF^(1/(2*Df))
LogReachCms 5.894761  1        2.427913
WeightClass 4.294365 12        1.062602
log(Height) 6.695866  1        2.587637
WinStreak   1.004200  1        1.002098
```

```r
vif(model_clean)
```

```
                GVIF Df GVIF^(1/(2*Df))
LogReachCms 5.735358  1        2.394861
WeightClass 4.195464 12        1.061571
log(Height) 6.624430  1        2.573797
WinStreak   1.004118  1        1.002057
```

OPTION : INTERACTION TERM

```r
model_clean_int <- lm(AvgSigStrLanded ~ LogReachCms + WinStreak * WeightClass + log(Height),

par(mfrow = c(2, 2), mar = c(2,2,2,2))
plot(model_clean_int)
```



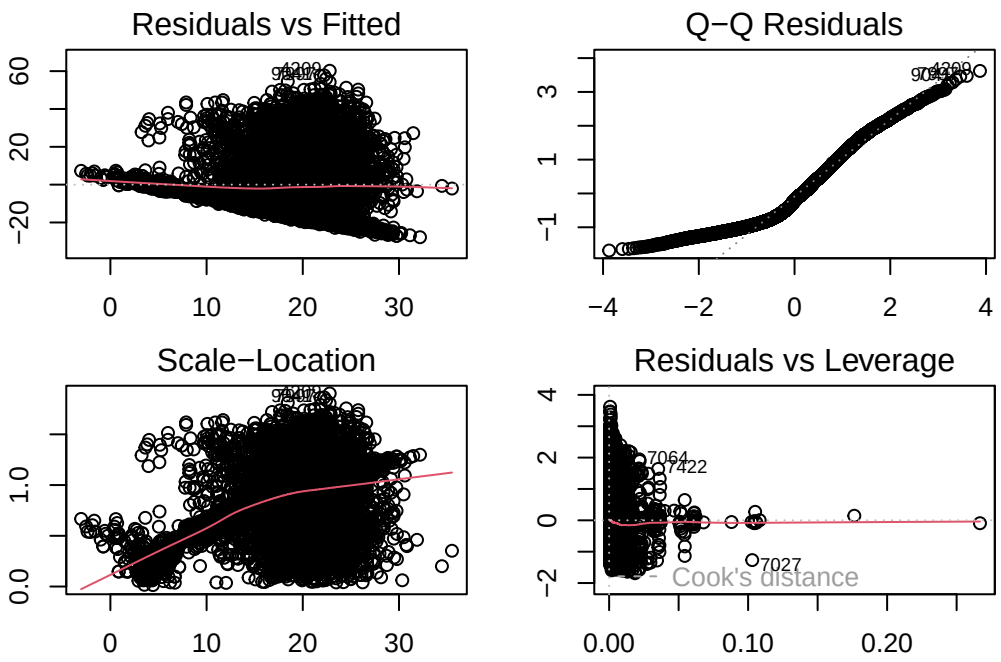

```
vif(model_clean_int)
```

```
there are higher-order terms (interactions) in this model
consider setting type = 'predictor'; see ?vif
```

```
                              GVIF Df GVIF^(1/(2*Df))
LogReachCms               5.750458  1        2.398011
WinStreak                 9.975393  1        3.158385
WeightClass            1011.233303 12        1.334142
log(Height)               6.646197  1        2.578022
WinStreak:WeightClass 2303.903508 12        1.380711
```

OPTION: INTERACTION TERM WITH THE OG DATA

```
model_int <- lm(AvgSigStrLanded ~ LogReachCms + WinStreak * WeightClass + log(Height), data =

par(mfrow = c(2, 2), mar = c(2,2,2,2))
plot(model_clean_int)
```




```
vif(model_int)
```

```
there are higher-order terms (interactions) in this model
consider setting type = 'predictor'; see ?vif
```

```
                              GVIF Df GVIF^(1/(2*Df))
LogReachCms               5.919067  1        2.432913
WinStreak                11.703034  1        3.420970
WeightClass             483.945667 12        1.293798
log(Height)               6.727039  1        2.593654
WinStreak:WeightClass  1247.012930 12        1.345844
```