Final-4870

2025-04-23

Data

```
data_2023 = read.csv("C:\\Users\\wrnaf\\Downloads\\Stats_Final_Data\\WHR_2023.csv")

head(data_2023)
```

```
##       country                       region happiness_score gdp_per_capita
## 1     Finland               Western Europe           7.804          1.888
## 2     Denmark               Western Europe           7.586          1.949
## 3     Iceland               Western Europe           7.530          1.926
## 4      Israel Middle East and North Africa           7.473          1.833
## 5 Netherlands               Western Europe           7.403          1.942
## 6      Sweden               Western Europe           7.395          1.921
##   social_support healthy_life_expectancy freedom_to_make_life_choices
## 1          1.585                   0.535                         0.772
## 2          1.548                   0.537                         0.734
## 3          1.620                   0.559                         0.738
## 4          1.521                   0.577                         0.569
## 5          1.488                   0.545                         0.672
## 6          1.510                   0.562                         0.754
##   generosity perceptions_of_corruption
## 1      0.126                    0.535
## 2      0.208                    0.525
## 3      0.250                    0.187
## 4      0.124                    0.158
## 5      0.251                    0.394
## 6      0.225                    0.520
```

Preprocessing Anaylsis

```
sum(is.na(data_2023))
```

```
## [1] 1
```

```
data_2023 = na.omit(data_2023)
```

```
summary(data_2023)
```

```
##    country             region          happiness_score gdp_per_capita
##  Length:136         Length:136         Min.   :1.859   Min.   :0.000
##  Class :character   Class :character   1st Qu.:4.702   1st Qu.:1.098
##  Mode  :character   Mode  :character   Median :5.694   Median :1.452
##                                        Mean   :5.544   Mean   :1.409
##                                        3rd Qu.:6.343   3rd Qu.:1.798
##                                        Max.   :7.804   Max.   :2.200
##  social_support   healthy_life_expectancy freedom_to_make_life_choices
##  Min.   :0.0000   Min.   :0.0000          Min.   :0.0000
##  1st Qu.:0.9597   1st Qu.:0.2485          1st Qu.:0.4587
##  Median :1.2255   Median :0.3895          Median :0.5575
##  Mean   :1.1551   Mean   :0.3662          Mean   :0.5409
##  3rd Qu.:1.4013   3rd Qu.:0.4875          3rd Qu.:0.6567
##  Max.   :1.6200   Max.   :0.7020          Max.   :0.7720
##    generosity     perceptions_of_corruption
##  Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0985   1st Qu.:0.05975
##  Median :0.1375   Median :0.11200
##  Mean   :0.1491   Mean   :0.14648
##  3rd Qu.:0.1993   3rd Qu.:0.18825
##  Max.   :0.4220   Max.   :0.56100
```

The summary shows a few things. We notice the happiness score spans from 1.86 - 7.80, with a mean value of 5.54. With most of the score ranging between 4.7 and 6.3.

```
headers = data_2023[sapply(data_2023, is.numeric)]

happiness_corr = cor(headers, use = "complete.obs")["happiness_score", ]

round(happiness_corr, 2)
```

```
##              happiness_score               gdp_per_capita
##                         1.00                         0.78
##               social_support      healthy_life_expectancy
##                         0.84                         0.75
## freedom_to_make_life_choices                   generosity
##                         0.66                         0.04
##    perceptions_of_corruption
##                         0.47
```

Next I wanted to understand the correlation happiness score has with its predictors. Based on the data we can see the strongest predictors are social support, GDP, and freedom to make life choices. Perceptions of corruption and Health life expectancy have a moderate to strong positive correlation where generosity has almost no relationship with the overall happiness score.

Model

```
X = headers[, -which(names(headers) == "happiness_score")]
Y = data_2023$happiness_score
```

```r
set.seed(100)
train_data = sample(1:nrow(data_2023), size = 0.75 * nrow(data_2023))

Xtraining1 = X[train_data, ]
Ytraining1 = Y[train_data]
Xtest = X[-train_data, ]
Ytest = Y[-train_data]

base_model = lm(Ytraining1 ~ ., data = Xtraining1)

summary(base_model)
```

```
##
## Call:
## lm(formula = Ytraining1 ~ ., data = Xtraining1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.58567 -0.22093  0.03942  0.32797  1.07211
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.3789     0.2507   5.499 3.21e-07 ***
## gdp_per_capita                  0.8199     0.2985   2.747  0.00719 **
## social_support                  1.2759     0.2789   4.575 1.44e-05 ***
## healthy_life_expectancy         0.5789     0.6667   0.868  0.38739
## freedom_to_make_life_choices    2.0787     0.4312   4.821 5.41e-06 ***
## generosity                      0.5344     0.7853   0.680  0.49786
## perceptions_of_corruption       0.6794     0.5246   1.295  0.19839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5071 on 95 degrees of freedom
## Multiple R-squared:  0.8223, Adjusted R-squared:  0.8111
## F-statistic: 73.29 on 6 and 95 DF,  p-value: < 2.2e-16
```

```r
predictions = predict(base_model, newdata = Xtest)


MPSE1= mean((predictions - Ytest)^2)
print(MPSE1)
```

```
## [1] 0.1968347
```

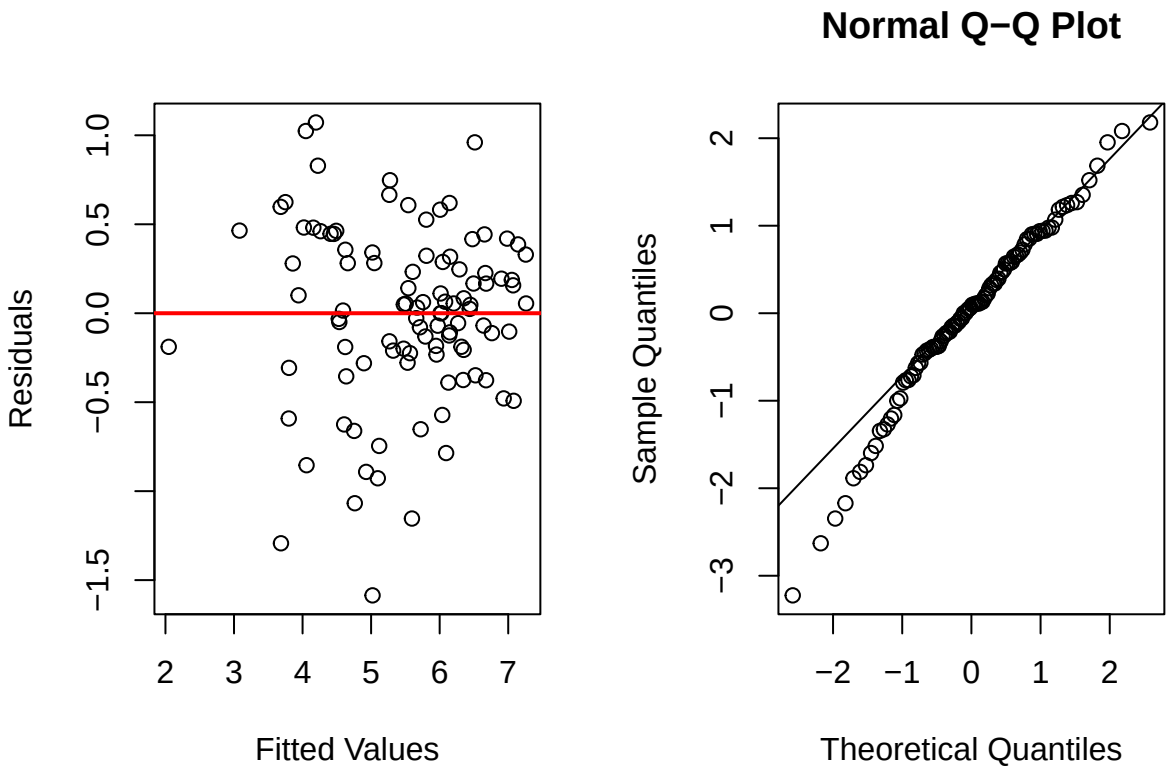
Assumptions

```r
par(mfrow = c(1, 2))

f = base_model$fitted.values
r = base_model$residuals
```

```r
plot(f, r, pch = 1, xlab = "Fitted Values", ylab = "Residuals")
abline(h = 0, col = "red", lwd = 2)

r = scale(r)
qqnorm(r)
qqline(r)
```



Linearity - There is no curve or trend to the residual plot meaning that linearity is not violated Normality - Most points fall on the QQ plot line leaving normality to not be violated Constant Variance - The Vertical spread is uniform over the risdual plot so constant variance is not violated Independence - No obvious correlation in risdual plot so independence holds.

Multi-collinearity Check

```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.4.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.4.3
```

```r
vif(base_model)
```

```
##               gdp_per_capita                  social_support
##                     6.360857                        3.672477
##      healthy_life_expectancy freedom_to_make_life_choices
##                     4.301438                        1.624795
##                   generosity    perceptions_of_corruption
##                     1.308252                        1.609157
```

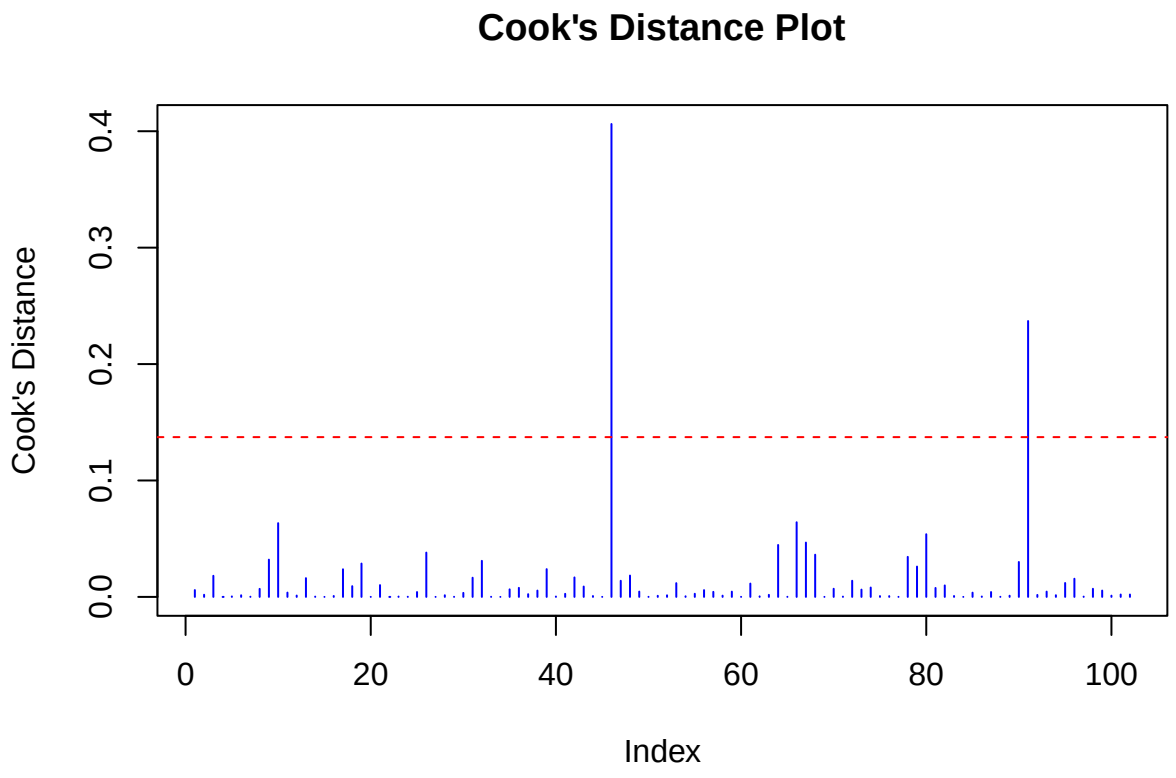
No serious cases of multi-collinearity. GDP is the highest at 6.36, but does not exceed the threshold of 10 so it is okay to include it in the data especially since it has a strong correlation with happiness score.

Outliers

```r
out = 2*(6 + 1)/ nrow(Xtraining1)

cd = cooks.distance(base_model)

plot(cd, type = "h", col = "blue", main = "Cook's Distance Plot", ylab = "Cook's Distance")
abline(h = out, col = "red", lty = 2)
```



```r
ip = which(cd > out)
ip
```

```
## 132 136
##  46  91
```

Influential points of 46 and 91.

```r
outliers = which(abs(rstudent(base_model)) > 1.96)
outliers
```

```
##  91 132  86 129 136 112
##  26  46  79  80  91  96
```

Outliers:

46, 91

79 80 96 26

```r
remove = intersect(ip, outliers)
```

Removes 46 and 91

```r
Xtraining2 = Xtraining1[-remove, ]
Ytraining2  = Ytraining1[-remove]


outlier_model= lm(Ytraining2 ~ ., data = Xtraining2)

summary(outlier_model)
```

```
##
## Call:
## lm(formula = Ytraining2 ~ ., data = Xtraining2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.13950 -0.27498  0.05683  0.26863  0.99906
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.5226     0.2282   6.671 1.80e-09 ***
## gdp_per_capita                  1.2668     0.2825   4.485 2.08e-05 ***
## social_support                  1.0835     0.2603   4.162 7.03e-05 ***
## healthy_life_expectancy        -0.2654     0.6533  -0.406    0.686
## freedom_to_make_life_choices    1.8328     0.3935   4.658 1.06e-05 ***
## generosity                      0.1263     0.7200   0.175    0.861
## perceptions_of_corruption       0.4800     0.4756   1.009    0.316
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4551 on 93 degrees of freedom
## Multiple R-squared:  0.8436, Adjusted R-squared:  0.8335
## F-statistic:  83.6 on 6 and 93 DF,  p-value: < 2.2e-16
```

Future Prediction

```
predictions = predict(outlier_model, newdata = Xtest)

MPSE2 = mean((predictions - Ytest)^2)
print(MPSE2)
```

```
## [1] 0.2520164
```

There are a few key ways removing the outliers changed the model. First the R squared and Adjust R^2 increased as well as the F-statistic. Standard error decreased as well. What is most interesting is that the statistically significant variables became much more significant and the insignificant variables became more more insignificant. This leads to me to believe this is an obvious sign of over fitting. The outliers made the model tighter on the data set making model summary nicer, but took out some of the real world messiness, when applying it to the test data. This in turn caused the MPSE to increase from .19 to .25. Because my goal is to predict happiness levels we do don't want to exclude these outliers.

Variabe Selection - AIC

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.4.3
```

```
backwards_model = stepAIC(base_model, direction = "backward")
```

```
## Start:  AIC=-131.77
## Ytraining1 ~ gdp_per_capita + social_support + healthy_life_expectancy +
##     freedom_to_make_life_choices + generosity + perceptions_of_corruption
##
##                                Df Sum of Sq    RSS     AIC
## - generosity                    1    0.1191 24.550 -133.27
## - healthy_life_expectancy       1    0.1939 24.625 -132.96
## - perceptions_of_corruption     1    0.4314 24.863 -131.98
## <none>                                      24.431 -131.77
## - gdp_per_capita                1    1.9407 26.372 -125.97
## - social_support                1    5.3830 29.814 -113.46
## - freedom_to_make_life_choices  1    5.9774 30.409 -111.44
##
## Step:  AIC=-133.27
## Ytraining1 ~ gdp_per_capita + social_support + healthy_life_expectancy +
##     freedom_to_make_life_choices + perceptions_of_corruption
##
```

```
##                                Df Sum of Sq    RSS     AIC
## - healthy_life_expectancy       1    0.1642 24.715 -134.59
## <none>                                      24.550 -133.27
## - perceptions_of_corruption     1    0.6450 25.195 -132.63
## - gdp_per_capita                1    1.8231 26.373 -127.97
## - freedom_to_make_life_choices  1    6.2851 30.835 -112.02
## - social_support                1    6.3697 30.920 -111.74
##
## Step:  AIC=-134.59
## Ytraining1 ~ gdp_per_capita + social_support + freedom_to_make_life_choices +
##     perceptions_of_corruption
##
##                                Df Sum of Sq    RSS     AIC
## <none>                                      24.715 -134.59
## - perceptions_of_corruption     1    0.7020 25.417 -133.74
## - gdp_per_capita                1    4.7072 29.422 -118.81
## - freedom_to_make_life_choices  1    6.1486 30.863 -113.93
## - social_support                1    6.7799 31.494 -111.87
```

```
summary(backwards_model)
```

```
##
## Call:
## lm(formula = Ytraining1 ~ gdp_per_capita + social_support + freedom_to_make_life_choices +
##     perceptions_of_corruption, data = Xtraining1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.79796 -0.23315  0.03334  0.33156  1.02931
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.4114     0.2148   6.569 2.52e-09 ***
## gdp_per_capita                  0.9168     0.2133   4.298 4.10e-05 ***
## social_support                  1.3593     0.2635   5.158 1.32e-06 ***
## freedom_to_make_life_choices    2.0824     0.4239   4.912 3.65e-06 ***
## perceptions_of_corruption       0.8215     0.4949   1.660      0.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5048 on 97 degrees of freedom
## Multiple R-squared:  0.8203, Adjusted R-squared:  0.8129
## F-statistic: 110.7 on 4 and 97 DF,  p-value: < 2.2e-16
```

```
predictions = predict(backwards_model, newdata = Xtest)

MPSE3 = mean((predictions - Ytest)^2)
print(MPSE3)
```

```
## [1] 0.2029312
```

Variable Selection - CP

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.4.3
```

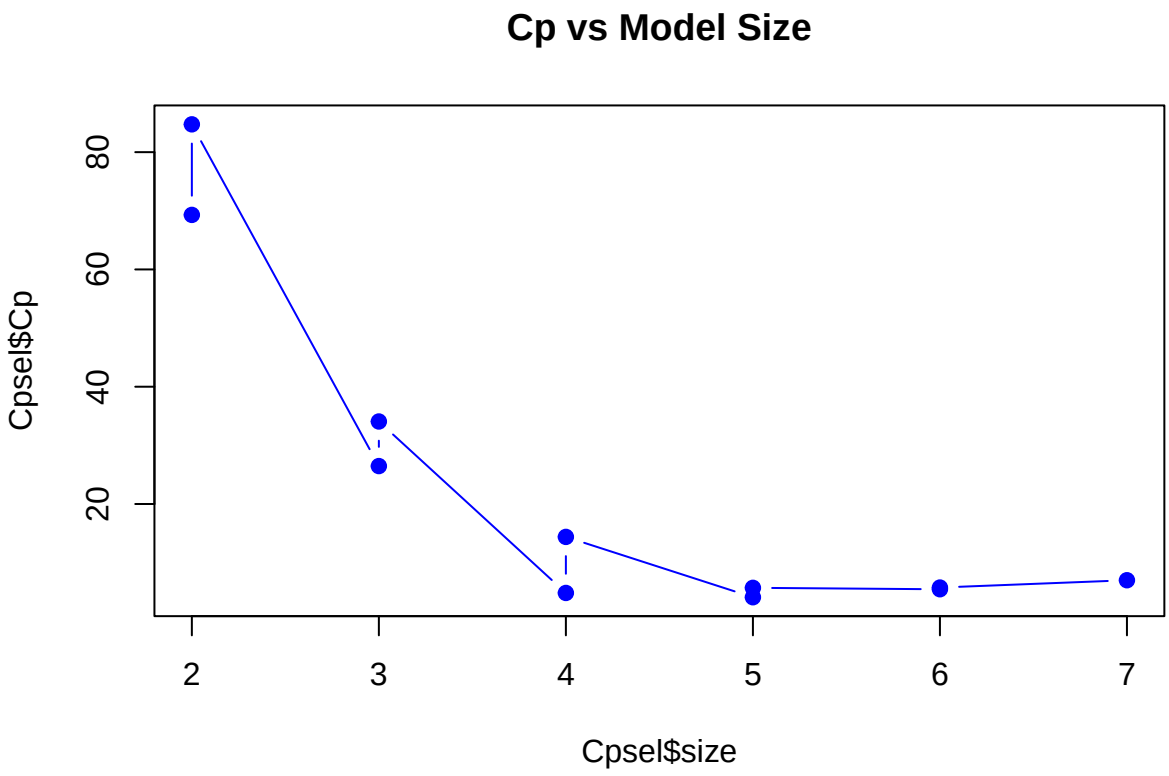
```
Cpsel = leaps(Xtraining1, Ytraining1, method = "Cp", nbest = 2)
cbind(Cpsel$which, Cpsel$size, Cpsel$Cp)
```

```
##   1 2 3 4 5 6
## 1 0 1 0 0 0 0 2 69.300313
## 1 1 0 0 0 0 0 2 84.739853
## 2 1 0 0 1 0 0 3 26.468819
## 2 0 1 0 1 0 0 3 34.069470
## 3 1 1 0 1 0 0 4  4.831083
## 3 0 1 1 1 0 0 4 14.390392
## 4 1 1 0 1 0 1 5  4.101560
## 4 1 1 0 1 1 0 5  5.707461
## 5 1 1 1 1 0 1 6  5.463034
## 5 1 1 0 1 1 1 6  5.754049
## 6 1 1 1 1 1 1 7  7.000000
```

```
index = which(Cpsel$Cp == min(Cpsel$Cp))
Cpsel$which[index, ]
```

```
##     1     2     3     4     5     6
##  TRUE  TRUE FALSE  TRUE FALSE  TRUE
```

```
plot(Cpsel$size, Cpsel$Cp, type = "b", col = "blue", pch = 19, main = "Cp vs Model Size")
```




```
summary(base_model)
```

```
##
## Call:
## lm(formula = Ytraining1 ~ ., data = Xtraining1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.58567 -0.22093  0.03942  0.32797  1.07211
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.3789     0.2507   5.499 3.21e-07 ***
## gdp_per_capita                  0.8199     0.2985   2.747  0.00719 **
## social_support                  1.2759     0.2789   4.575 1.44e-05 ***
## healthy_life_expectancy         0.5789     0.6667   0.868  0.38739
## freedom_to_make_life_choices    2.0787     0.4312   4.821 5.41e-06 ***
## generosity                      0.5344     0.7853   0.680  0.49786
## perceptions_of_corruption       0.6794     0.5246   1.295  0.19839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5071 on 95 degrees of freedom
## Multiple R-squared:  0.8223, Adjusted R-squared:  0.8111
## F-statistic: 73.29 on 6 and 95 DF,  p-value: < 2.2e-16
```

```r
fitselect = lm(Ytraining1 ~ gdp_per_capita + social_support + freedom_to_make_life_choices + perceptions
summary(fitselect)
```

```
##
## Call:
## lm(formula = Ytraining1 ~ gdp_per_capita + social_support + freedom_to_make_life_choices +
##     perceptions_of_corruption, data = Xtraining1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.79796 -0.23315  0.03334  0.33156  1.02931
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.4114     0.2148   6.569 2.52e-09 ***
## gdp_per_capita                  0.9168     0.2133   4.298 4.10e-05 ***
## social_support                  1.3593     0.2635   5.158 1.32e-06 ***
## freedom_to_make_life_choices    2.0824     0.4239   4.912 3.65e-06 ***
## perceptions_of_corruption       0.8215     0.4949   1.660      0.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5048 on 97 degrees of freedom
## Multiple R-squared:  0.8203, Adjusted R-squared:  0.8129
## F-statistic: 110.7 on 4 and 97 DF,  p-value: < 2.2e-16
```

```r
MPSE4 = mean((Ytest - predict(fitselect, Xtest))^2)
print(MPSE4)
```

```
## [1] 0.2029312
```

Interaction

```r
set.seed(100)
train_data = sample(1:nrow(data_2023), size = 0.75 * nrow(data_2023))

Xtraining1 = X[train_data, ]
Ytraining1 = Y[train_data]
Xtest = X[-train_data, ]
Ytest = Y[-train_data]
```

```r
cor(Xtraining1[, c("generosity", "social_support",
                   "perceptions_of_corruption",
                   "freedom_to_make_life_choices")],
    use = "complete.obs")
```

```
##                               generosity social_support
## generosity                   1.000000000    0.006973008
## social_support               0.006973008    1.000000000
```

```
## perceptions_of_corruption    0.159749758    0.204544393
## freedom_to_make_life_choices 0.172805275    0.540285803
##                              perceptions_of_corruption
## generosity                                   0.1597498
## social_support                               0.2045444
## perceptions_of_corruption                    1.0000000
## freedom_to_make_life_choices                 0.3732436
##                              freedom_to_make_life_choices
## generosity                                      0.1728053
## social_support                                  0.5402858
## perceptions_of_corruption                       0.3732436
## freedom_to_make_life_choices                    1.0000000
```

```
interactions_model = lm(Ytraining1 ~ gdp_per_capita + social_support +
                                     healthy_life_expectancy + freedom_to_make_life_choices +
                                     generosity + perceptions_of_corruption +
                                     social_support:freedom_to_make_life_choices+
                                     freedom_to_make_life_choices:generosity+
                                     generosity:perceptions_of_corruption+
                                     social_support:perceptions_of_corruption,
                                   data = Xtraining1)


summary(interactions_model)
```

```
##
## Call:
## lm(formula = Ytraining1 ~ gdp_per_capita + social_support + healthy_life_expectancy +
##     freedom_to_make_life_choices + generosity + perceptions_of_corruption +
##     social_support:freedom_to_make_life_choices + freedom_to_make_life_choices:generosity +
##     generosity:perceptions_of_corruption + social_support:perceptions_of_corruption,
##     data = Xtraining1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.59677 -0.21593  0.02518  0.31356  1.07345
##
## Coefficients:
##                                           Estimate Std. Error t value
## (Intercept)                                1.03650    0.57751   1.795
## gdp_per_capita                             0.86346    0.30999   2.785
## social_support                             1.16550    0.45372   2.569
## healthy_life_expectancy                    0.60624    0.67760   0.895
## freedom_to_make_life_choices               2.93771    1.14192   2.573
## generosity                                 3.79753    2.84010   1.337
## perceptions_of_corruption                 -0.25029    2.75978  -0.091
## social_support:freedom_to_make_life_choices  0.06112    0.87668   0.070
## freedom_to_make_life_choices:generosity   -7.32330    5.78552  -1.266
## generosity:perceptions_of_corruption       3.67828    8.44434   0.436
## social_support:perceptions_of_corruption   0.33549    1.99533   0.168
##                                           Pr(>|t|)
## (Intercept)                                 0.0760 .
## gdp_per_capita                              0.0065 **
## social_support                              0.0118 *
```

```
## healthy_life_expectancy                         0.3733
## freedom_to_make_life_choices                    0.0117 *
## generosity                                      0.1845
## perceptions_of_corruption                       0.9279
## social_support:freedom_to_make_life_choices     0.9446
## freedom_to_make_life_choices:generosity         0.2088
## generosity:perceptions_of_corruption            0.6642
## social_support:perceptions_of_corruption        0.8668
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5135 on 91 degrees of freedom
## Multiple R-squared:  0.8255, Adjusted R-squared:  0.8064
## F-statistic: 43.06 on 10 and 91 DF,  p-value: < 2.2e-16
```

```
predictions = predict(interactions_model, newdata = Xtest)

MPSE5= mean((predictions - Ytest)^2)
print(MPSE5)
```

```
## [1] 0.1841892
```

Cross Validation Over Different Datasets

```
data_2015 = read.csv("C:\\Users\\wrnaf\\Downloads\\Stats_Final_Data\\WHR_2015.csv")

headers_2015 = data_2015[sapply(data_2015, is.numeric)]

Xtest = headers_2015[, -which(names(headers_2015) == "happiness_score")]
Ytest = headers_2015$happiness_score


base_predictions = predict(base_model, newdata = Xtest)

base_MPSE = mean((base_predictions - Ytest)^2)
print(base_MPSE)
```

```
## [1] 0.621148
```

```
outlier_predictions = predict(outlier_model, newdata = Xtest)

outlier_MPSE = mean((outlier_predictions - Ytest)^2)
print(outlier_MPSE)
```

```
## [1] 1.324196
```

```
backwards_predictions = predict(backwards_model, newdata = Xtest)

backwards_MPSE = mean((backwards_predictions - Ytest)^2)
print(backwards_MPSE)
```

```
## [1] 1.028072
```

```
fitselect_predictions = predict(fitselect, newdata = Xtest)

fitselect_MPSE = mean((fitselect_predictions - Ytest)^2)
print(fitselect_MPSE)
```

```
## [1] 1.028072
```

```
interaction_predictions = predict(interactions_model, newdata = Xtest)

interaction_MPSE = mean((interaction_predictions - Ytest)^2)
print(interaction_MPSE)
```

```
## [1] 0.5986021
```

```
MPSE_table_2015 = data.frame(
  Model_2015 = c("Base Model", "Outlier Model", "Backward Stepwise Model", "Cp-Selected Model", "Intera
  MPSE = c(base_MPSE, outlier_MPSE, backwards_MPSE, fitselect_MPSE, interaction_MPSE)
)

print(MPSE_table_2015)
```

```
##                Model_2015      MPSE
## 1              Base Model 0.6211480
## 2           Outlier Model 1.3241960
## 3 Backward Stepwise Model 1.0280716
## 4       Cp-Selected Model 1.0280716
## 5       Interaction Model 0.5986021
```

```
MPSE_table_2023 = data.frame(
  Model_2023 = c("Base Model", "Outlier Model", "Backward Stepwise", "Cp-Selected Model", "Interaction M
  MPSE = c(MPSE1, MPSE2, MPSE3, MPSE4, MPSE5)
)

print(MPSE_table_2023)
```

```
##          Model_2023      MPSE
## 1        Base Model 0.1968347
## 2     Outlier Model 0.2520164
## 3 Backward Stepwise 0.2029312
## 4 Cp-Selected Model 0.2029312
## 5 Interaction Model 0.1841892
```