Deliverable 3

Numeric and Binary targets Forecasting Models

Jordi Catafal, Lluís Cerdà, Judit Serna, Tomàs Serra

June 8, 2024

Contents

Linear model building

```
df.original <- df[, 1:14]

# Predict duration with the variable age
m1 <- lm(formula = target ~ age, data=df.original)

summary(m1)

##
## Call:
## lm(formula = target ~ age, data = df.original)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54137 -0.26736  0.08543  0.25426  0.50030
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.9066570  0.0530751  17.083  < 2e-16 ***
## age         -0.0068021  0.0009618  -7.072 2.82e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2792 on 1023 degrees of freedom
## Multiple R-squared:  0.04662,    Adjusted R-squared:  0.04568
## F-statistic: 50.02 on 1 and 1023 DF,  p-value: 2.819e-12
```

The low R-squared value (0.04662) tells us that the model does not explain much of the variability in the target variable. This suggests that age alone is not a strong predictor. Given this, the predictive power of age on the target variable is considered "weak."

However, knowing that our data might have particular characteristics, we should consider further analysis. Specifically, we should examine if the relationship between age and the target variable is different for individuals over 45 years old. This subgroup analysis might reveal patterns or relationships that are not apparent when considering the entire dataset.

```
df2.original <- df2[,1:14] #(df2 té age > 45)


# Predict duration with the variable age
m2 <- lm(formula = target ~ age, data=df2.original)

summary(m2)

##
## Call:
## lm(formula = target ~ age, data = df2.original)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5606 -0.2601  0.1130  0.1810  0.2476
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.766750   0.226809   3.381 0.000881 ***
## age         -0.002722   0.005581  -0.488 0.626279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2551 on 186 degrees of freedom
## Multiple R-squared:  0.001278,   Adjusted R-squared:  -0.004092
## F-statistic: 0.2379 on 1 and 186 DF,  p-value: 0.6263
```

We conducted a new linear regression analysis using only the subset of data where the age is greater than 45 (df2). The aim was to see if the relationship between age and the target variable improves in this specific age group. However, the results indicate that there is still not a significant improvement.

The results show that limiting the dataset to individuals older than 45 did not improve the predictive power of age on the target variable. The R-squared value is still very low, and the age coefficient is not significant.

```
mNumAll <- lm(formula = target ~ age + trestbps + chol + thalach +
oldpeak + ca, data=df.original)

summary(mNumAll)
```

```
##
## Call:
## lm(formula = target ~ age + trestbps + chol + thalach + oldpeak +
##     ca, data = df.original)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.60861 -0.17258  0.04945  0.17458  0.57516
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.2222171  0.0945424   2.350  0.01894 *
## age          0.0026758  0.0009653   2.772  0.00567 **
## trestbps    -0.0011537  0.0004414  -2.614  0.00908 **
## chol        -0.0002051  0.0001461  -1.404  0.16063
## thalach      0.0033185  0.0003657   9.075  < 2e-16 ***
## oldpeak     -0.0489060  0.0069172  -7.070 2.87e-12 ***
## ca          -0.1041312  0.0086949 -11.976  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.2335 on 1018 degrees of freedom
## Multiple R-squared:  0.3366, Adjusted R-squared:  0.3327
## F-statistic:  86.1 on 6 and 1018 DF,  p-value: < 2.2e-16
```

The inclusion of multiple predictors has significantly improved the model compared to using age alone. The R-squared value has increased from 0.04662 (when using age alone) to 0.3366, indicating a much better fit. This means that the combined effect of these variables explains about 33.66% of the variability in the target variable, compared to only 4.662% when using age alone.

Although the R-squared value is still relatively low, the model with multiple numeric predictors provides a much better fit than using age alone. This improvement demonstrates the value of incorporating multiple relevant predictors to enhance the explanatory power of the regression model.

```
mNum <- lm(formula = target ~ age + trestbps + chol + thalach + oldpeak +
ca, data=df2.original)

summary(mNum)

##
## Call:
## lm(formula = target ~ age + trestbps + chol + thalach + oldpeak +
##     ca, data = df2.original)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67093 -0.05890  0.03364  0.13070  0.42339
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.455e-01  3.084e-01  -1.769   0.0786 .
## age          7.465e-03  5.033e-03   1.483   0.1397
## trestbps     4.039e-04  1.329e-03   0.304   0.7615
## chol        -3.551e-05  3.587e-04  -0.099   0.9212
## thalach      5.549e-03  8.307e-04   6.679 2.86e-10 ***
## oldpeak     -6.696e-02  1.581e-02  -4.234 3.64e-05 ***
## ca          -5.904e-02  4.413e-02  -1.338   0.1826
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2115 on 181 degrees of freedom
## Multiple R-squared:  0.3317, Adjusted R-squared:  0.3096
## F-statistic: 14.97 on 6 and 181 DF,  p-value: 7.01e-14
```

When we use only numeric variables from the subset of data where age is greater than 45, the R-squared value is 0.3317. This suggests that these predictors explain about 33.17% of the variability in the target variable.

Given that the R-squared values are still relatively low, it indicates that the models, even with multiple predictors, do not capture all the factors affecting the target variable. Therefore, the next step is to explore the impact of the factors directly.

Stepwise method

Let's try Stepwise method in both cases:

```
# Using stepwise methodology to get the most optimize linear regression
model with Akaike criteria
m3 <- lm(formula= target ~ ., data =df.original)

m_step = step(m3)
```

```
## Start:  AIC=-3226.71
## target ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach +
##     exang + oldpeak + slope + ca + thal
##
##             Df Sum of Sq    RSS     AIC
## - thal       2    0.0282 42.437 -3230.0
## - age        1    0.0200 42.429 -3228.2
## - fbs        1    0.0234 42.432 -3228.1
## - restecg    2    0.1442 42.553 -3227.2
## <none>                   42.409 -3226.7
## - chol       1    0.1449 42.554 -3225.2
## - thalach    1    0.3411 42.750 -3220.5
## - oldpeak    1    0.3737 42.783 -3219.7
## - trestbps   1    0.4808 42.890 -3217.2
## - exang      1    0.8074 43.216 -3209.4
## - slope      2    1.2543 43.663 -3200.8
## - cp         3    3.5377 45.947 -3150.6
## - sex        1    3.6760 46.085 -3143.5
## - ca         1    4.3675 46.777 -3128.2
##
## Step:  AIC=-3230.03
## target ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach +
##     exang + oldpeak + slope + ca
##
##             Df Sum of Sq    RSS     AIC
## - fbs        1    0.0199 42.457 -3231.5
## - age        1    0.0201 42.457 -3231.5
## - restecg    2    0.1439 42.581 -3230.6
## <none>                   42.437 -3230.0
## - chol       1    0.1392 42.577 -3228.7
## - thalach    1    0.3507 42.788 -3223.6
## - oldpeak    1    0.3718 42.809 -3223.1
## - trestbps   1    0.4805 42.918 -3220.5
## - exang      1    0.8171 43.254 -3212.5
## - slope      2    1.2583 43.696 -3204.1
## - cp         3    3.5668 46.004 -3153.3
```

```
## - sex       1    3.6959 46.133 -3146.4
## - ca        1    4.3434 46.781 -3132.2
##
## Step:  AIC=-3231.55
## target ~ age + sex + cp + trestbps + chol + restecg + thalach +
##     exang + oldpeak + slope + ca
##
##            Df Sum of Sq    RSS     AIC
## - age       1    0.0220 42.479 -3233.0
## - restecg   2    0.1394 42.597 -3232.2
## <none>                  42.457 -3231.5
## - chol      1    0.1402 42.597 -3230.2
## - thalach   1    0.3551 42.812 -3225.0
## - oldpeak   1    0.3943 42.852 -3224.1
## - trestbps  1    0.4621 42.919 -3222.5
## - exang     1    0.8015 43.259 -3214.4
## - slope     2    1.2744 43.732 -3205.2
## - cp        3    3.6692 46.126 -3152.6
## - sex       1    3.6842 46.141 -3148.3
## - ca        1    4.3560 46.813 -3133.4
##
## Step:  AIC=-3233.02
## target ~ sex + cp + trestbps + chol + restecg + thalach + exang +
##     oldpeak + slope + ca
##
##            Df Sum of Sq    RSS     AIC
## - restecg   2    0.1336 42.613 -3233.8
## <none>                  42.479 -3233.0
## - chol      1    0.1265 42.606 -3232.0
## - thalach   1    0.3364 42.816 -3226.9
## - oldpeak   1    0.3982 42.877 -3225.5
## - trestbps  1    0.4402 42.919 -3224.5
## - exang     1    0.8266 43.306 -3215.3
## - slope     2    1.2824 43.762 -3206.5
## - cp        3    3.7085 46.188 -3153.2
## - sex       1    3.7863 46.265 -3147.5
## - ca        1    4.5381 47.017 -3131.0
##
## Step:  AIC=-3233.8
## target ~ sex + cp + trestbps + chol + thalach + exang + oldpeak +
##     slope + ca
##
##            Df Sum of Sq    RSS     AIC
## <none>                  42.613 -3233.8
## - chol      1    0.1726 42.785 -3231.7
## - thalach   1    0.3405 42.953 -3227.6
## - oldpeak   1    0.3986 43.011 -3226.3
## - trestbps  1    0.5012 43.114 -3223.8
## - exang     1    0.8249 43.438 -3216.1
## - slope     2    1.3415 43.954 -3206.0
```

```
## - cp        3    3.7885 46.401 -3152.5
## - sex       1    3.8797 46.492 -3146.5
## - ca        1    4.7072 47.320 -3128.4
```

```
summary(m_step)
```

```
##
## Call:
## lm(formula = target ~ sex + cp + trestbps + chol + thalach +
##     exang + oldpeak + slope + ca, data = df.original)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58890 -0.12716  0.02328  0.13615  0.62064
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.9523474  0.0867416  10.979  < 2e-16 ***
## sexMale             -0.1404702  0.0146340  -9.599  < 2e-16 ***
## cpAtypical Angina   -0.0254359  0.0294548  -0.864 0.388037
## cpNon-anginal Pain  -0.0044123  0.0271533  -0.162 0.870948
## cpTypical Angina    -0.1547369  0.0268327  -5.767 1.07e-08 ***
## trestbps            -0.0013201  0.0003826  -3.450 0.000584 ***
## chol                -0.0002630  0.0001299  -2.024 0.043190 *
## thalach              0.0009731  0.0003422   2.844 0.004550 **
## exangYes            -0.0721881  0.0163094  -4.426 1.06e-05 ***
## oldpeak             -0.0225551  0.0073310  -3.077 0.002150 **
## slopeFlat           -0.0849292  0.0161481  -5.259 1.76e-07 ***
## slopeUpsloping      -0.0133240  0.0305493  -0.436 0.662823
## ca                  -0.0801848  0.0075839 -10.573  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2052 on 1012 degrees of freedom
## Multiple R-squared:  0.4906, Adjusted R-squared:  0.4846
## F-statistic: 81.23 on 12 and 1012 DF,  p-value: < 2.2e-16
```

The stepwise regression method has helped us identify a more optimized model with a higher R-squared value, indicating a better fit to the data, this the one with all the variables. However, we notice that some predictors are not statistically significant.

```
# Using stepwise methodology to get the most optimize linear regression
model with Akaike criteria
m4 <- lm(formula= target ~ ., data =df2.original)

m_step = step(m4)
```

```
## Start:  AIC=-688.91
## target ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach +
##     exang + oldpeak + slope + ca + thal
##
```

```
##            Df Sum of Sq    RSS     AIC
## - thalach   1   0.00018 4.0197 -690.90
## - thal      1   0.00172 4.0213 -690.83
## - oldpeak   1   0.00176 4.0213 -690.83
## - restecg   1   0.00803 4.0276 -690.54
## - chol      1   0.01886 4.0384 -690.03
## - fbs       1   0.03057 4.0501 -689.49
## <none>                  4.0196 -688.91
## - slope     2   0.10225 4.1218 -688.19
## - age       1   0.07610 4.0957 -687.39
## - trestbps  1   0.08537 4.1049 -686.96
## - ca        1   0.20375 4.2233 -681.61
## - sex       1   0.37327 4.3928 -674.22
## - exang     1   0.47793 4.4975 -669.79
## - cp        3   1.97513 5.9947 -619.77
##
## Step:  AIC=-690.9
## target ~ age + sex + cp + trestbps + chol + fbs + restecg + exang +
##     oldpeak + slope + ca + thal
##
##            Df Sum of Sq    RSS     AIC
## - oldpeak   1   0.00174 4.0215 -692.82
## - thal      1   0.00209 4.0218 -692.80
## - restecg   1   0.00820 4.0279 -692.52
## - chol      1   0.01901 4.0387 -692.02
## - fbs       1   0.03054 4.0503 -691.48
## <none>                  4.0197 -690.90
## - age       1   0.07669 4.0964 -689.35
## - trestbps  1   0.08859 4.1083 -688.80
## - slope     2   0.14985 4.1696 -688.02
## - ca        1   0.20943 4.2292 -683.35
## - sex       1   0.37545 4.3952 -676.12
## - exang     1   0.52587 4.5456 -669.79
## - cp        3   2.32041 6.3401 -611.23
##
## Step:  AIC=-692.82
## target ~ age + sex + cp + trestbps + chol + fbs + restecg + exang +
##     slope + ca + thal
##
##            Df Sum of Sq    RSS     AIC
## - thal      1   0.00104 4.0225 -694.77
## - restecg   1   0.00784 4.0293 -694.45
## - chol      1   0.01768 4.0392 -694.00
## - fbs       1   0.02881 4.0503 -693.48
## <none>                  4.0215 -692.82
## - age       1   0.07629 4.0978 -691.29
## - trestbps  1   0.08815 4.1096 -690.74
## - slope     2   0.15850 4.1800 -689.55
## - ca        1   0.21068 4.2322 -685.22
## - sex       1   0.37372 4.3952 -678.11
```

```
## - exang     1   0.70106 4.7225 -664.61
## - cp        3   2.37695 6.3984 -611.51
##
## Step:  AIC=-694.77
## target ~ age + sex + cp + trestbps + chol + fbs + restecg + exang +
##     slope + ca
##
##            Df Sum of Sq    RSS     AIC
## - restecg   1   0.00733 4.0299 -696.43
## - chol      1   0.01818 4.0407 -695.92
## - fbs       1   0.02873 4.0512 -695.43
## <none>                  4.0225 -694.77
## - age       1   0.07545 4.0980 -693.28
## - trestbps  1   0.08940 4.1119 -692.64
## - slope     2   0.16805 4.1906 -691.08
## - ca        1   0.21314 4.2357 -687.07
## - sex       1   0.38237 4.4049 -679.70
## - exang     1   0.77586 4.7984 -663.61
## - cp        3   2.40175 6.4243 -612.76
##
## Step:  AIC=-696.43
## target ~ age + sex + cp + trestbps + chol + fbs + exang + slope +
##     ca
##
##            Df Sum of Sq    RSS     AIC
## - chol      1   0.01758 4.0474 -697.61
## - fbs       1   0.03813 4.0680 -696.66
## <none>                  4.0299 -696.43
## - age       1   0.08196 4.1118 -694.64
## - trestbps  1   0.09611 4.1260 -694.00
## - slope     2   0.16845 4.1983 -692.73
## - ca        1   0.23954 4.2694 -687.57
## - sex       1   0.39168 4.4215 -680.99
## - exang     1   0.81251 4.8424 -663.90
## - cp        3   2.50671 6.5366 -611.50
##
## Step:  AIC=-697.61
## target ~ age + sex + cp + trestbps + fbs + exang + slope + ca
##
##            Df Sum of Sq    RSS     AIC
## - fbs       1   0.02762 4.0750 -698.33
## <none>                  4.0474 -697.61
## - age       1   0.09616 4.1436 -695.20
## - trestbps  1   0.11383 4.1613 -694.40
## - slope     2   0.16665 4.2141 -694.03
## - ca        1   0.22479 4.2722 -689.45
## - sex       1   0.40184 4.4493 -681.82
## - exang     1   0.79669 4.8441 -665.83
## - cp        3   2.49281 6.5402 -613.39
##
```

```
## Step:  AIC=-698.33
## target ~ age + sex + cp + trestbps + exang + slope + ca
##
##            Df Sum of Sq    RSS     AIC
## <none>                   4.0750 -698.33
## - age       1   0.08360 4.1586 -696.52
## - trestbps  1   0.09846 4.1735 -695.84
## - slope     2   0.17131 4.2464 -694.59
## - ca        1   0.22106 4.2961 -690.40
## - sex       1   0.37831 4.4534 -683.64
## - exang     1   0.87403 4.9491 -663.80
## - cp        3   2.54692 6.6220 -613.06
```

```
summary(m_step)
```

```
##
## Call:
## lm(formula = target ~ age + sex + cp + trestbps + exang + slope +
##     ca, data = df2.original)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44346 -0.07103  0.00196  0.08962  0.41477
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.3922833  0.1950313   2.011  0.04580 *
## age                 0.0069379  0.0036409   1.906  0.05833 .
## sexMale            -0.1153075  0.0284457  -4.054 7.54e-05 ***
## cpAtypical Angina  -0.0064447  0.0560720  -0.115  0.90863
## cpNon-anginal Pain  0.0017003  0.0546261   0.031  0.97520
## cpTypical Angina   -0.2746960  0.0503768  -5.453 1.65e-07 ***
## trestbps            0.0019978  0.0009661   2.068  0.04009 *
## exangYes           -0.2032154  0.0329817  -6.161 4.73e-09 ***
## slopeFlat          -0.0751278  0.0282839  -2.656  0.00862 **
## slopeUpsloping      0.0009026  0.0525029   0.017  0.98630
## ca                 -0.1081494  0.0349016  -3.099  0.00226 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1517 on 177 degrees of freedom
## Multiple R-squared:  0.6637, Adjusted R-squared:  0.6447
## F-statistic: 34.93 on 10 and 177 DF,  p-value: < 2.2e-16
```

Improvement in R-squared: - The optimized model has a high R-squared value of 0.6637, indicating a strong fit. This means the predictors explain a significant portion of the variability in the target variable.

Significant Predictors: - Variables such as sex, cp (Typical Angina), trestbps, exang, slope (Flat), and ca are significant predictors, indicating their strong relationship with

the target variable. - Some predictors like age are marginally significant, suggesting a potential impact.

Non-significant Predictors: - Variables such as cp (Atypical Angina), cp (Non-anginal Pain), and slope (Upsloping) have high p-values and are not statistically significant. These categories could potentially be removed to simplify the model further without losing much predictive power.

Also we got some observations to mention: - Some coefficients have signs contrary to logical expectations. For example, trestbps is positive, suggesting higher blood pressure is associated with a healthier outcome, which contradicts typical medical understanding. This anomaly might require further investigation. - The ca variable has a negative coefficient, which is not expected, as the number of major vessels colored by fluoroscopy should logically correlate positively with health. - Certain categories like exang (Yes) and cp (Typical Angina) have coefficients and significance levels that align with medical understanding, indicating their importance in predicting the target variable.

Optimization

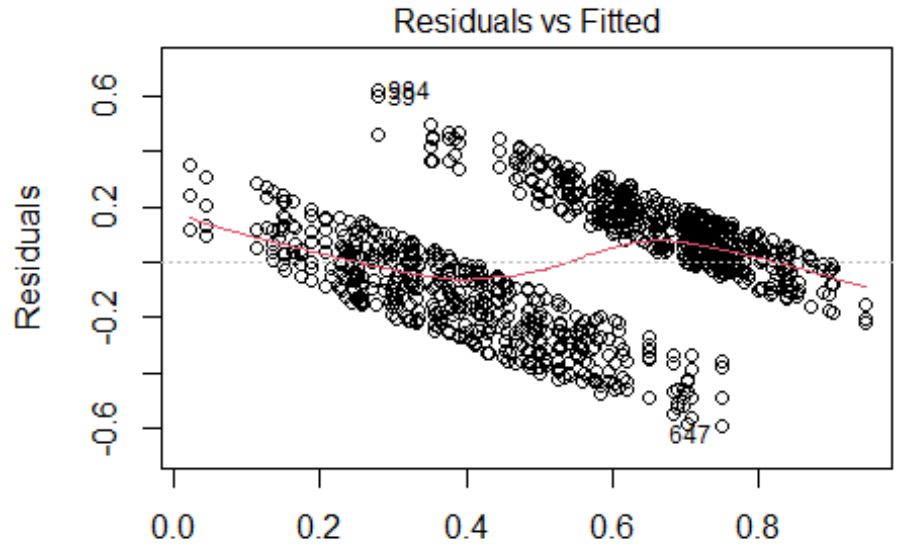
Now we will try to optimize all the categories with a p-value > 0.05, in order to see if we keep them or not.

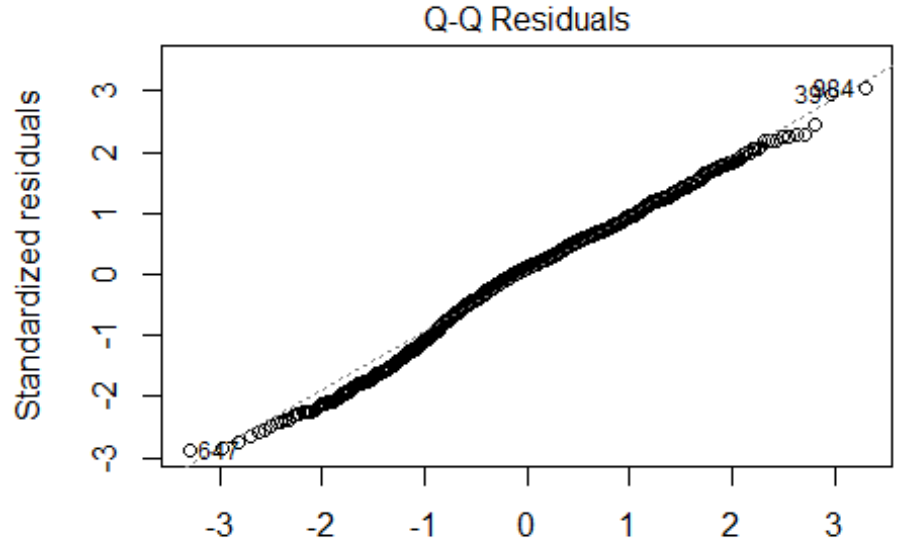
```
optimize = lm(formula = target ~ sex + cp + trestbps + chol + thalach +
    exang + oldpeak + slope + ca, data = df.original)
summary(optimize)
```

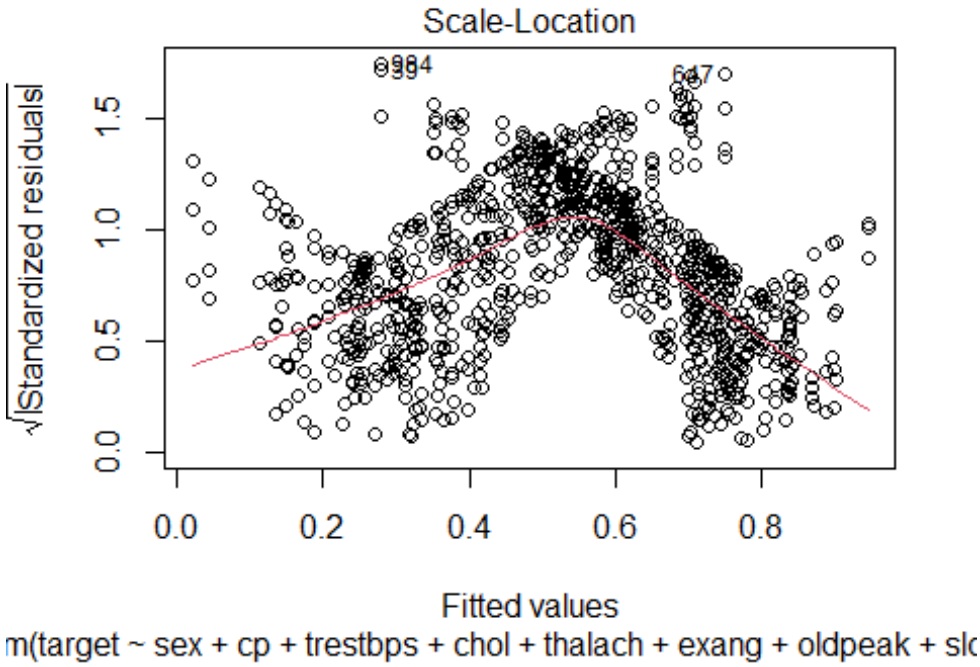
```
##
## Call:
## lm(formula = target ~ sex + cp + trestbps + chol + thalach +
##     exang + oldpeak + slope + ca, data = df.original)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58890 -0.12716  0.02328  0.13615  0.62064
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.9523474  0.0867416  10.979  < 2e-16 ***
## sexMale            -0.1404702  0.0146340  -9.599  < 2e-16 ***
## cpAtypical Angina  -0.0254359  0.0294548  -0.864 0.388037
## cpNon-anginal Pain -0.0044123  0.0271533  -0.162 0.870948
## cpTypical Angina   -0.1547369  0.0268327  -5.767 1.07e-08 ***
## trestbps           -0.0013201  0.0003826  -3.450 0.000584 ***
## chol               -0.0002630  0.0001299  -2.024 0.043190 *
## thalach             0.0009731  0.0003422   2.844 0.004550 **
## exangYes           -0.0721881  0.0163094  -4.426 1.06e-05 ***
## oldpeak            -0.0225551  0.0073310  -3.077 0.002150 **
```

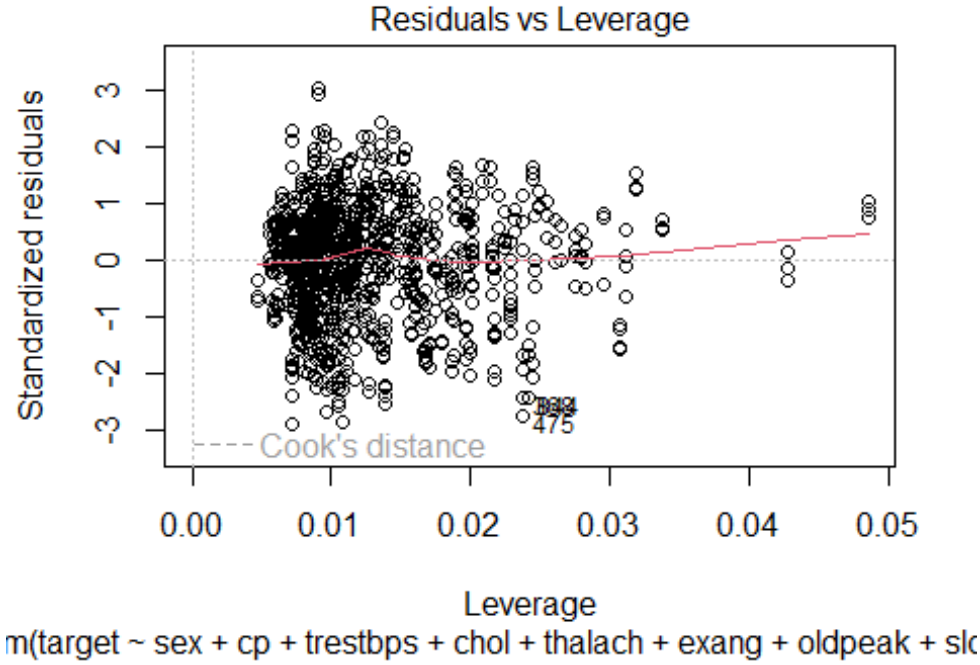
```
## slopeFlat          -0.0849292  0.0161481  -5.259 1.76e-07 ***
## slopeUpsloping     -0.0133240  0.0305493  -0.436 0.662823
## ca                 -0.0801848  0.0075839 -10.573  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2052 on 1012 degrees of freedom
## Multiple R-squared:  0.4906, Adjusted R-squared:  0.4846
## F-statistic: 81.23 on 12 and 1012 DF,  p-value: < 2.2e-16

plot(optimize)
```


Residuals vs Fitted
Residuals
0.6
0.2
-0.2
-0.6
647
0.0
0.2
0.4
0.6
0.8
Fitted values
m(target ~ sex + cp + trestbps + chol + thalach + exang + oldpeak + slc


Q-Q Residuals
Standardized residuals
3
2
1
0
-1
-2
-3
647
-3
-2
-1
0
1
2
3
Theoretical Quantiles
m(target ~ sex + cp + trestbps + chol + thalach + exang + oldpeak + slc





1. Residuals vs Fitted Plot

- Pattern Observed: The plot shows two distinct bands of residuals, one at lower fitted values and one at higher fitted values. There is a noticeable gap between 0.3 and 0.5 in the fitted values.

- Interpretation: This pattern suggests heteroscedasticity, where the variance of the residuals changes with the fitted values. The gap likely indicates that the target variable has no values in that range, leading to a jump in the residuals.

2. Normal Q-Q Plot

- Pattern Observed: The residuals mostly follow the reference line, but there are deviations at both ends.
- Interpretation: The residuals are approximately normally distributed, but there are some outliers. The deviations at the tails suggest potential issues with normality in the extremes.

3. Scale-Location Plot

- Pattern Observed: The spread of the residuals seems to increase with the fitted values, forming a funnel shape.
- Interpretation: This pattern further confirms heteroscedasticity, where the variance of residuals is not constant. It suggests that the model might be less reliable for higher fitted values.

4. Residuals vs Leverage Plot

- Pattern Observed: Most residuals are centered around zero with a few points having high leverage and high standardized
- Interpretation: This plot helps identify influential points that have a significant impact on the model's fit. Points outside the Cook's distance lines are potentially influential observations that may need further investigation.

From those graphics we can get this conclusions: - Gap in Fitted Values: The gap in the first plot suggests that the target variable values are not continuous across the range but jump from 0.4 to 0.6, skipping intermediate values. This may indicate an issue with the distribution of the target variable. - Heteroscedasticity: Both the Residuals vs Fitted and Scale-Location plots indicate heteroscedasticity. This violates one of the key assumptions of linear regression, which assumes constant variance of residuals. - Outliers and Influential Points: The Normal Q-Q plot shows some deviations at the tails, indicating outliers. The Residuals vs Leverage plot shows some points with high leverage, which could disproportionately affect the model.

```
optimize1.1 = lm(formula = target ~ sex + trestbps + chol + thalach +
    exang + oldpeak + slope + ca, data = df.original)
summary(optimize1.1)
```

```
##
## Call:
## lm(formula = target ~ sex + trestbps + chol + thalach + exang +
##     oldpeak + slope + ca, data = df.original)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69669 -0.13941  0.02841  0.14598  0.60752
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)     0.8137025  0.0817816   9.950  < 2e-16 ***
## sexMale        -0.1406972  0.0151385  -9.294  < 2e-16 ***
## trestbps       -0.0011633  0.0003935  -2.956 0.003186 **
## chol           -0.0003018  0.0001352  -2.233 0.025771 *
## thalach         0.0015019  0.0003504   4.287 1.99e-05 ***
## exangYes       -0.1278573  0.0158347  -8.075 1.91e-15 ***
## oldpeak        -0.0273447  0.0075238  -3.634 0.000293 ***
## slopeFlat      -0.0845891  0.0167742  -5.043 5.43e-07 ***
## slopeUpsloping -0.0087801  0.0318038  -0.276 0.782550
## ca             -0.0920263  0.0077917 -11.811  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2138 on 1015 degrees of freedom
## Multiple R-squared:  0.4453, Adjusted R-squared:  0.4404
## F-statistic: 90.55 on 9 and 1015 DF,  p-value: < 2.2e-16
```

Here we remove we have removed cp. The model has a lower value of 0.4453. Predictors have a similar significance when compared to the previous iteration, both strong predictors and weak predictors. So the result is a little worse.

```
optimize1.2 = lm(formula = target ~ sex + trestbps + chol + thalach +
    exang + oldpeak + ca, data = df.original)
summary(optimize1.2)

##
## Call:
## lm(formula = target ~ sex + trestbps + chol + thalach + exang +
##     oldpeak + ca, data = df.original)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67300 -0.14451  0.02658  0.14662  0.58805
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.6880339  0.0793422   8.672  < 2e-16 ***
## sexMale     -0.1358865  0.0153194  -8.870  < 2e-16 ***
## trestbps    -0.0011044  0.0003982  -2.774  0.00564 **
## chol        -0.0003238  0.0001368  -2.367  0.01812 *
## thalach      0.0021029  0.0003357   6.263 5.55e-10 ***
## exangYes    -0.1351571  0.0159940  -8.450  < 2e-16 ***
## oldpeak     -0.0341328  0.0065519  -5.210 2.29e-07 ***
## ca          -0.0924965  0.0077965 -11.864  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2167 on 1017 degrees of freedom
## Multiple R-squared:  0.4289, Adjusted R-squared:  0.425
## F-statistic: 109.1 on 7 and 1017 DF,  p-value: < 2.2e-16
```

Similarly removing slope gives also a slightly worse result, with similar significance of the predictors.

```
optimize1.3 = lm(formula = target ~ sex + trestbps + chol + thalach +
    exang + oldpeak + cp + ca, data = df.original)
summary(optimize1.3)
```

```
##
## Call:
## lm(formula = target ~ sex + trestbps + chol + thalach + exang +
##     oldpeak + cp + ca, data = df.original)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5872 -0.1323  0.0259  0.1440  0.6016
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.8139412  0.0842260   9.664  < 2e-16 ***
## sexMale            -0.1351174  0.0148166  -9.119  < 2e-16 ***
## trestbps           -0.0012382  0.0003874  -3.196  0.00144 **
## chol               -0.0002863  0.0001316  -2.176  0.02982 *
## thalach             0.0015706  0.0003288   4.776 2.05e-06 ***
## exangYes           -0.0790352  0.0165018  -4.789 1.92e-06 ***
## oldpeak            -0.0292329  0.0064251  -4.550 6.02e-06 ***
## cpAtypical Angina  -0.0109759  0.0297700  -0.369  0.71244
## cpNon-anginal Pain  0.0056744  0.0274873   0.206  0.83649
## cpTypical Angina   -0.1461730  0.0271774  -5.378 9.33e-08 ***
## ca                 -0.0804728  0.0076027 -10.585  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2082 on 1014 degrees of freedom
## Multiple R-squared:  0.4746, Adjusted R-squared:  0.4694
## F-statistic: 91.59 on 10 and 1014 DF,  p-value: < 2.2e-16
```

We try a last attempt without slope but adding cp. The result has improved but it keeps much lower than our model with all the variables.

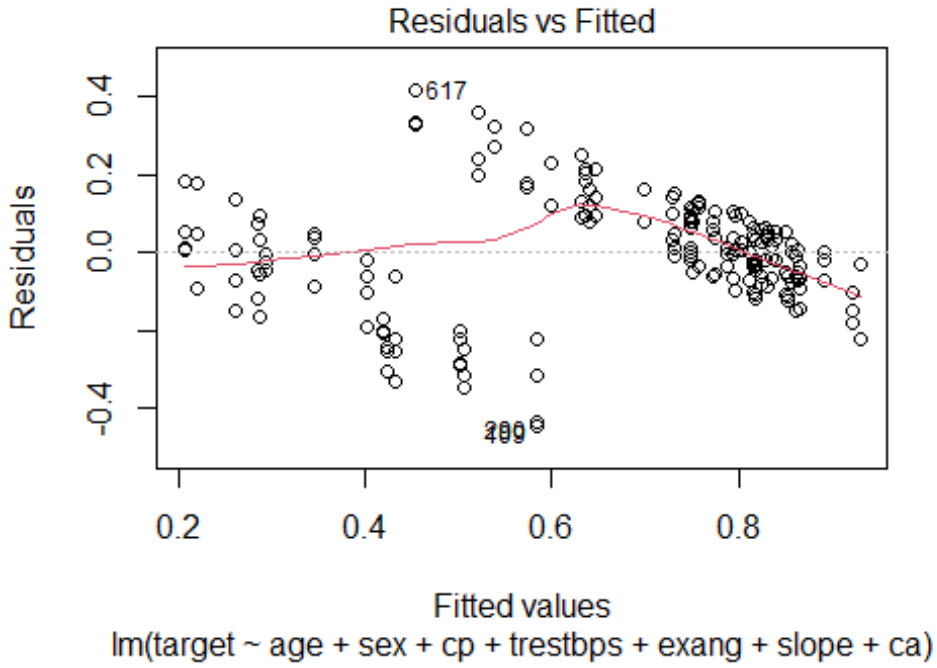
Now it would be interesting to try if our less biased data frame with the individuals with more than 45 years performs better:

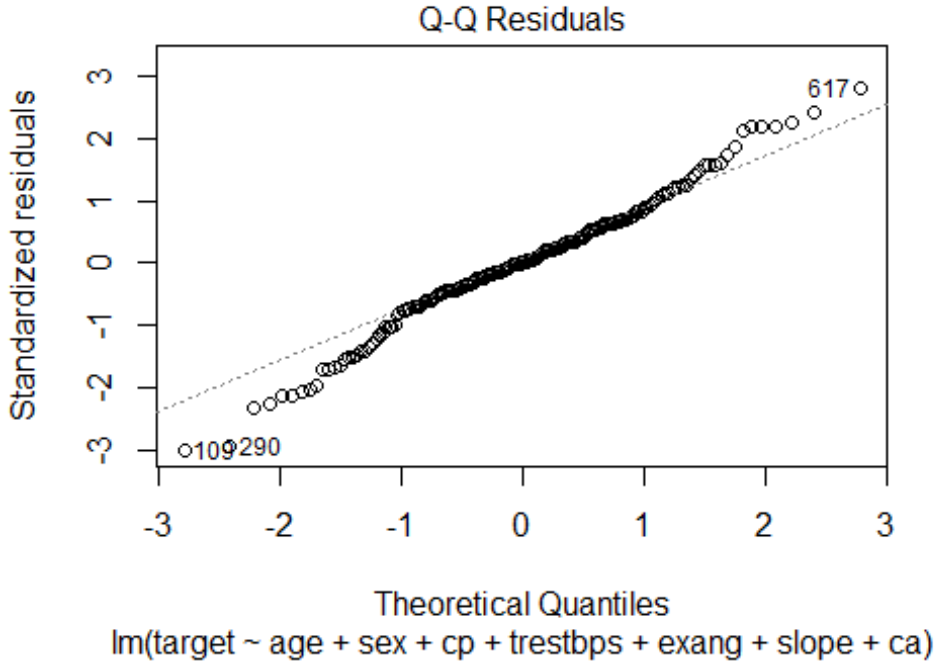
```
optimize2 = lm(formula = target ~ age + sex + cp + trestbps + exang +
slope +
    ca, data = df2.original)
summary(optimize2)
```

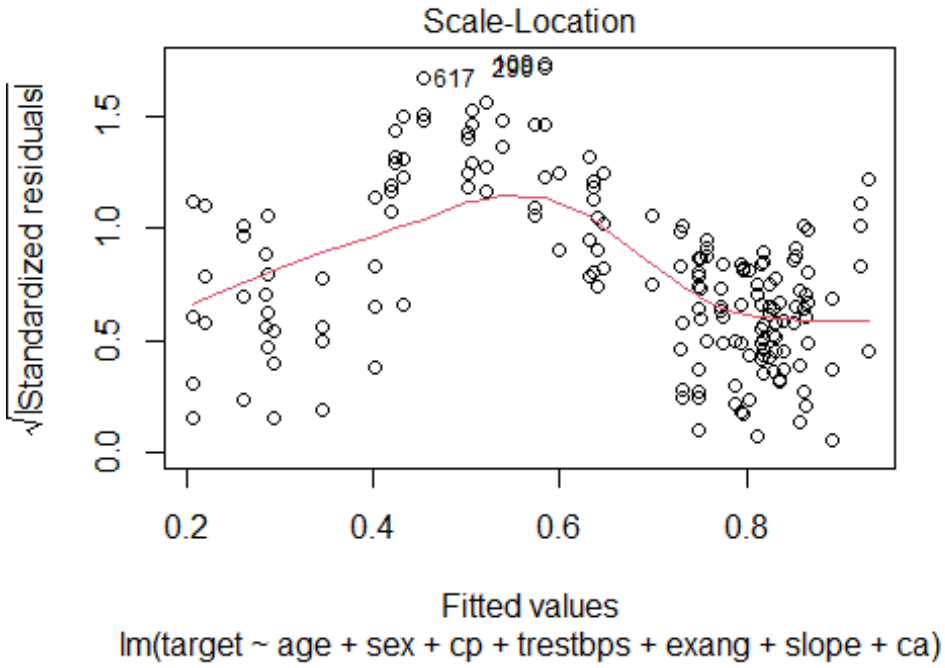
```
##
## Call:
## lm(formula = target ~ age + sex + cp + trestbps + exang + slope +
##     ca, data = df2.original)
```

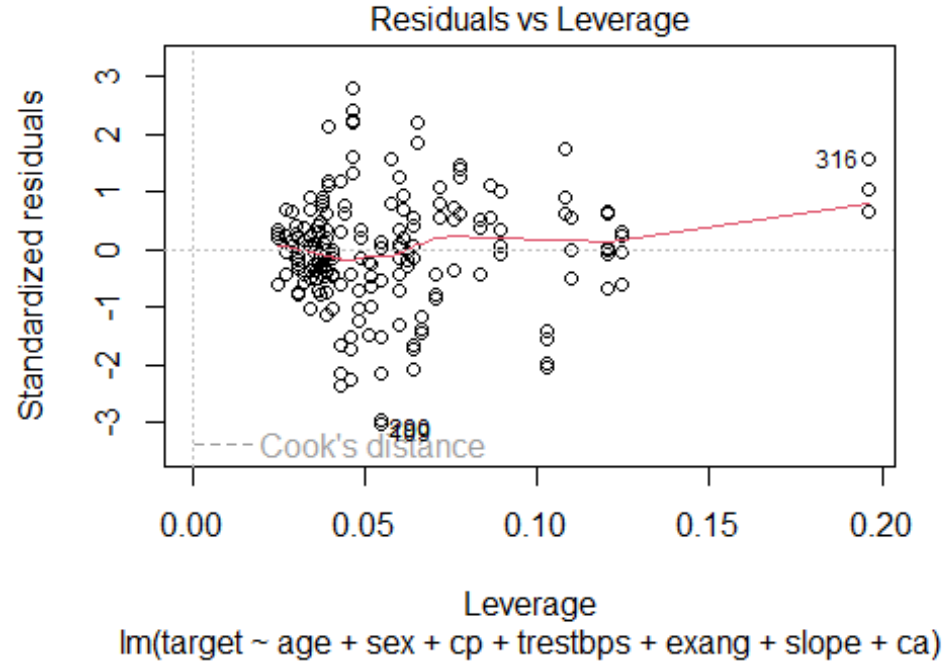
```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44346 -0.07103  0.00196  0.08962  0.41477
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.3922833  0.1950313   2.011  0.04580 *
## age                  0.0069379  0.0036409   1.906  0.05833 .
## sexMale             -0.1153075  0.0284457  -4.054 7.54e-05 ***
## cpAtypical Angina   -0.0064447  0.0560720  -0.115  0.90863
## cpNon-anginal Pain   0.0017003  0.0546261   0.031  0.97520
## cpTypical Angina    -0.2746960  0.0503768  -5.453 1.65e-07 ***
## trestbps             0.0019978  0.0009661   2.068  0.04009 *
## exangYes            -0.2032154  0.0329817  -6.161 4.73e-09 ***
## slopeFlat           -0.0751278  0.0282839  -2.656  0.00862 **
## slopeUpsloping       0.0009026  0.0525029   0.017  0.98630
## ca                  -0.1081494  0.0349016  -3.099  0.00226 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1517 on 177 degrees of freedom
## Multiple R-squared:  0.6637, Adjusted R-squared:  0.6447
## F-statistic: 34.93 on 10 and 177 DF,  p-value: < 2.2e-16

plot(optimize2)
```


Residuals vs Fitted
Residuals
Fitted values
lm(target ~ age + sex + cp + trestbps + exang + slope + ca)


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(target ~ age + sex + cp + trestbps + exang + slope + ca)





This first model with some of the variables gives as already a similar result in terms of R-squared to our complete data frame with all the variables. We will now perform some other tests to see if we can push it a little further.

Both the Residuals vs Fitted and Scale-Location plots keep indicating heteroscedasticity. Even with this second data frame.

```
optimize2.1 = lm(formula = target ~  sex + cp + trestbps + exang + slope
+
    ca, data = df2.original)
summary(optimize2.1)

##
## Call:
## lm(formula = target ~ sex + cp + trestbps + exang + slope + ca,
##     data = df2.original)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43953 -0.07524  0.00152  0.08434  0.42623
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.6597704  0.1363937   4.837 2.84e-06 ***
## sexMale             -0.1079819  0.0283922  -3.803 0.000196 ***
## cpAtypical Angina    0.0138666  0.0554549   0.250 0.802836
## cpNon-anginal Pain   0.0303308  0.0529058   0.573 0.567167
## cpTypical Angina    -0.2508690  0.0491596  -5.103 8.52e-07 ***
## trestbps             0.0018329  0.0009693   1.891 0.060238 .
## exangYes            -0.1996137  0.0331700  -6.018 9.83e-09 ***
## slopeFlat           -0.0679350  0.0282372  -2.406 0.017158 *
## slopeUpsloping       0.0047538  0.0528503   0.090 0.928430
## ca                  -0.0877593  0.0334654  -2.622 0.009488 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1529 on 178 degrees of freedom
## Multiple R-squared:  0.6568, Adjusted R-squared:  0.6394
## F-statistic: 37.85 on 9 and 178 DF,  p-value: < 2.2e-16
```

We have also tried in this new model to exclude some of the less significant variables. Here we can see by removing age that it hasn't much effect but it's a little bit worse.

```
optimize2.1 = lm(formula = target ~  age + sex + trestbps + exang + slope
+
    ca, data = df2.original)
summary(optimize2.1)

##
## Call:
## lm(formula = target ~ age + sex + trestbps + exang + slope +
##     ca, data = df2.original)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.65737 -0.06668  0.02493  0.11839  0.40807
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.421802   0.239163   1.764 0.079486 .
## age             0.004819   0.004403   1.094 0.275205
## sexMale        -0.163150   0.034462  -4.734 4.44e-06 ***
## trestbps        0.002276   0.001202   1.893 0.059941 .
## exangYes       -0.298122   0.037171  -8.020 1.30e-13 ***
## slopeFlat      -0.130040   0.034583  -3.760 0.000229 ***
## slopeUpsloping  0.001620   0.064902   0.025 0.980119
## ca             -0.116106   0.040592  -2.860 0.004733 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1918 on 180 degrees of freedom
## Multiple R-squared:  0.4535, Adjusted R-squared:  0.4322
## F-statistic: 21.34 on 7 and 180 DF,  p-value: < 2.2e-16
```

```
optimize2.1 = lm(formula = target ~ sex + trestbps + exang +
    ca, data = df2.original)
summary(optimize2.1)
```

```
##
## Call:
## lm(formula = target ~ sex + trestbps + exang + ca, data =
df2.original)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.64072 -0.07375  0.03740  0.11325  0.47854
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.542346   0.150602   3.601 0.000408 ***
## sexMale     -0.128543   0.034249  -3.753 0.000234 ***
## trestbps     0.002414   0.001230   1.963 0.051130 .
## exangYes    -0.350293   0.035132  -9.971  < 2e-16 ***
## ca          -0.096248   0.040462  -2.379 0.018402 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.198 on 183 degrees of freedom
## Multiple R-squared:  0.4076, Adjusted R-squared:  0.3946
## F-statistic: 31.48 on 4 and 183 DF,  p-value: < 2.2e-16
```

After all this test we see that the best model is the one obtained with all the variables and optimize, and also that we have better results using df2. By looking at the graphics of df2 we cannot see as many patters, but this is also because it has less individuals.

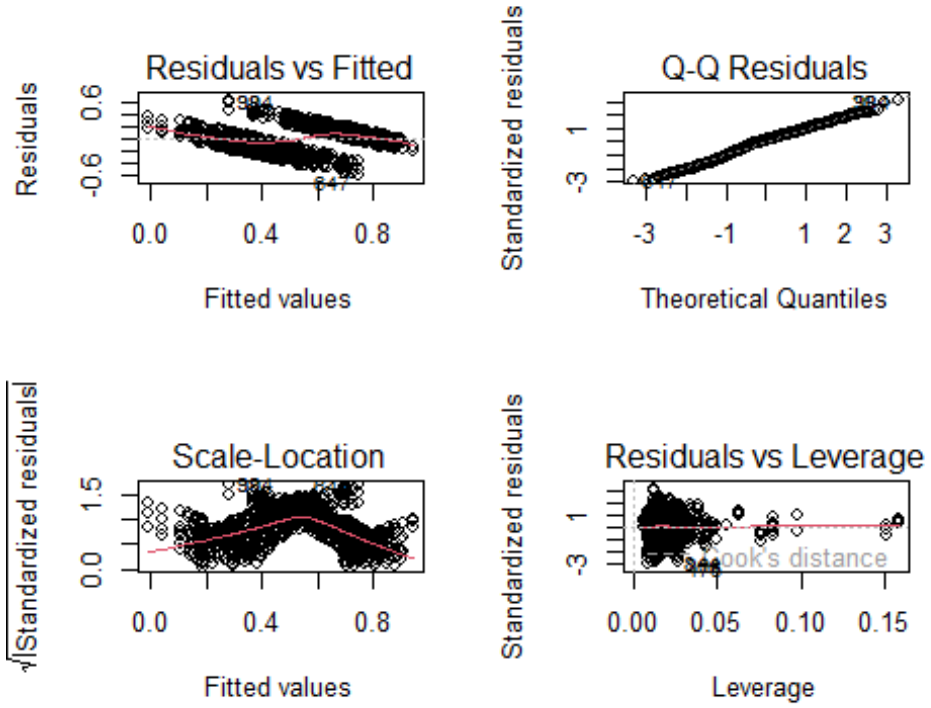
Mout

Finally we introduce a new column indicating the multivariate outliers and we see if they have significance in the model:

```
mMoutAll <- lm(formula= target ~ ., data =df[,1:15])
summary(mMoutAll)
```

```
##
## Call:
## lm(formula = target ~ ., data = df[, 1:15])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58938 -0.12753  0.02231  0.13264  0.61509
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.8982657  0.1044711   8.598  < 2e-16 ***
## age                      0.0003785  0.0008707   0.435  0.66388
## sexMale                 -0.1403903  0.0149217  -9.408  < 2e-16 ***
## cpAtypical Angina       -0.0221363  0.0297077  -0.745  0.45636
## cpNon-anginal Pain      -0.0033626  0.0272883  -0.123  0.90195
## cpTypical Angina        -0.1497929  0.0270419  -5.539 3.88e-08 ***
## trestbps                -0.0012115  0.0004066  -2.979  0.00296 **
## chol                    -0.0002037  0.0001347  -1.513  0.13069
## fbs> 120 mg/dL           0.0130256  0.0193867   0.672  0.50181
## restecgHypertrophy      -0.0068783  0.0561317  -0.123  0.90250
## restecgNormal           -0.0245715  0.0135449  -1.814  0.06997 .
## thalach                  0.0010493  0.0003658   2.869  0.00421 **
## exangYes                -0.0750217  0.0164637  -4.557 5.83e-06 ***
## oldpeak                 -0.0123846  0.0086533  -1.431  0.15268
## slopeFlat               -0.0874589  0.0164159  -5.328 1.23e-07 ***
## slopeUpsloping          -0.0112129  0.0317478  -0.353  0.72402
## ca                      -0.0818918  0.0080328 -10.195  < 2e-16 ***
## thalNormal              -0.0625568  0.0819818  -0.763  0.44561
## thalReversible Defect   -0.0020741  0.0261911  -0.079  0.93690
## moutYesMOut             -0.0697956  0.0316013  -2.209  0.02743 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2049 on 1005 degrees of freedom
## Multiple R-squared:  0.4955, Adjusted R-squared:  0.486
## F-statistic: 51.95 on 19 and 1005 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(mMoutAll)
```

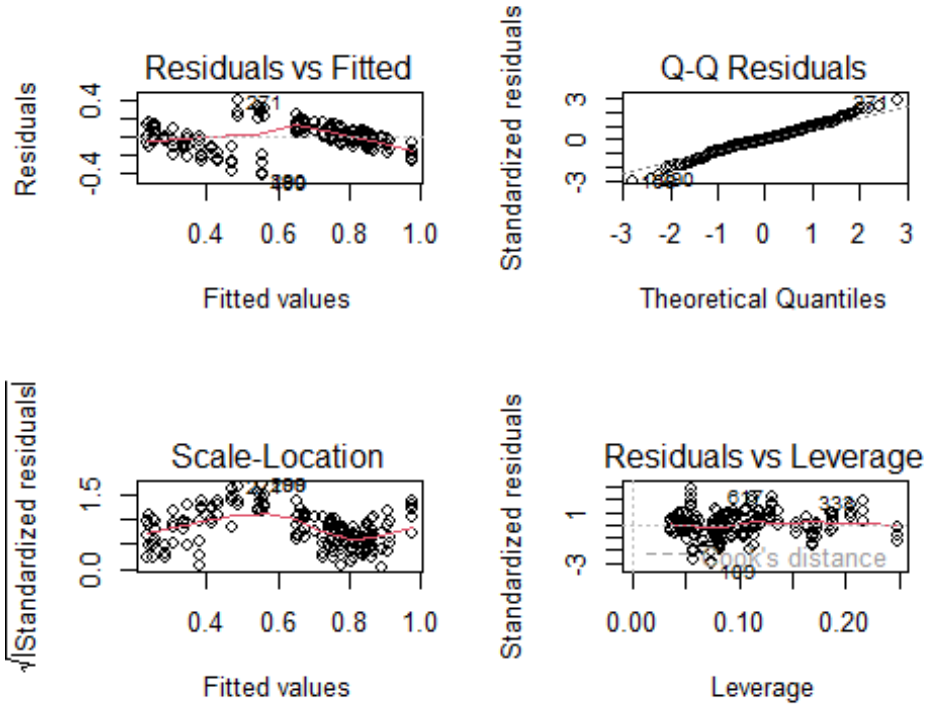



```
mMout <- lm(formula= target ~ ., data =df2[,1:15])
summary(mMout)

##
## Call:
## lm(formula = target ~ ., data = df2[, 1:15])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41505 -0.07084 -0.00183  0.08036  0.39243
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.2821351  0.2772275   1.018 0.310266
## age                     0.0054909  0.0036684   1.497 0.136300
## sexMale                -0.1213204  0.0277961  -4.365 2.20e-05 ***
## cpAtypical Angina      -0.0478359  0.0588600  -0.813 0.417523
## cpNon-anginal Pain     -0.0475636  0.0572611  -0.831 0.407339
## cpTypical Angina       -0.3389279  0.0559511  -6.058 8.62e-09 ***
## trestbps                0.0016013  0.0010375   1.543 0.124592
## chol                    0.0006206  0.0002733   2.271 0.024421 *
## fbs> 120 mg/dL          0.0313351  0.0620594   0.505 0.614268
## restecgNormal          -0.0214525  0.0258593  -0.830 0.407936
## thalach                 0.0004628  0.0008692   0.532 0.595120
## exangYes               -0.2066103  0.0396332  -5.213 5.34e-07 ***
## oldpeak                 0.0532522  0.0214120   2.487 0.013847 *
## slopeFlat              -0.0643248  0.0370906  -1.734 0.084684 .
```

```
## slopeUpsloping          0.0099665  0.0752971   0.132 0.894854
## ca                     -0.1333709  0.0351546  -3.794 0.000206 ***
## thalReversible Defect   0.0479770  0.0555123   0.864 0.388663
## moutYesMOut            -0.3567190  0.0742753  -4.803 3.42e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1443 on 170 degrees of freedom
## Multiple R-squared:  0.7079, Adjusted R-squared:  0.6787
## F-statistic: 24.23 on 17 and 170 DF,  p-value: < 2.2e-16
```

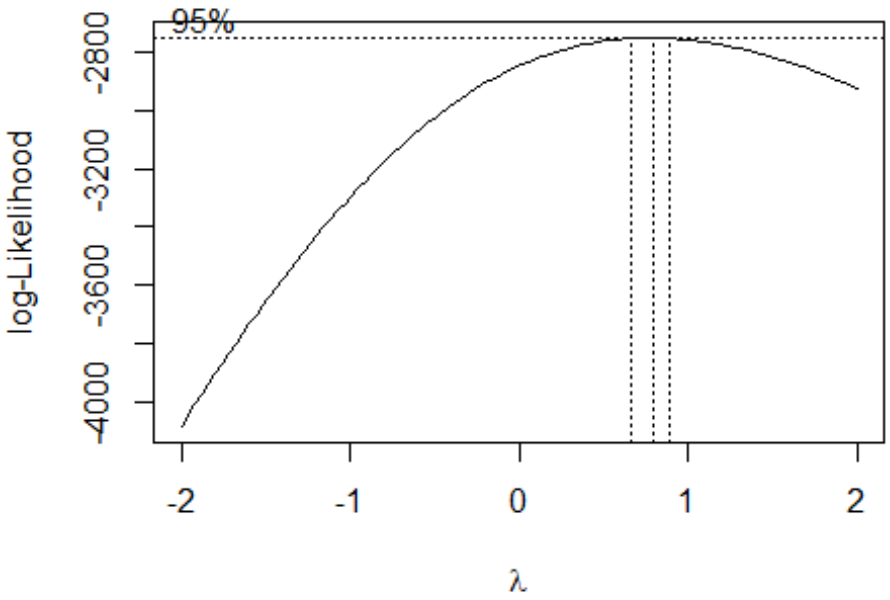
```
par(mfrow = c(2, 2))
plot(mMout)
```



In both the full data frames, df and df2 we can see some improvement on the model by using the variable mOut. This last one is our model with better results.

Target variable transformation?

```
library(MASS)
boxcox(optimize,data=df.original)
```



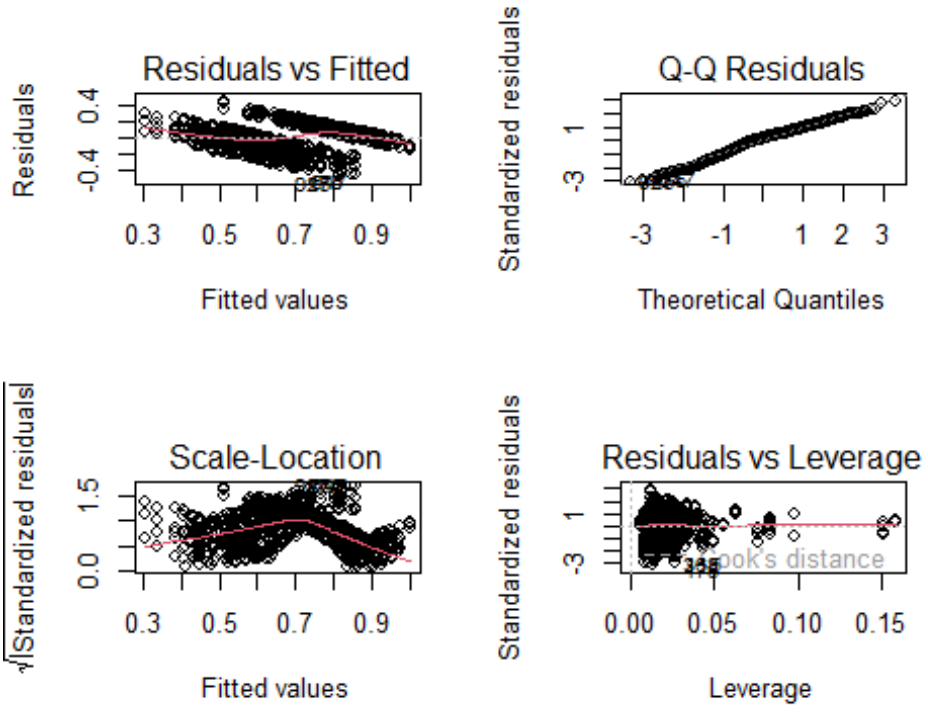
Lambda is lower than one and higher than 0, but being really close to 1 makes us consider trying the square root transformation to the target variable.

```
transm <- lm(formula= sqrt(target) ~ ., data =df[,1:15])
summary(transm)

##
## Call:
## lm(formula = sqrt(target) ~ ., data = df[, 1:15])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46859 -0.09132  0.02049  0.09813  0.43534
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.9678824  0.0788360  12.277  < 2e-16 ***
## age                     0.0001614  0.0006570   0.246  0.80606
## sexMale                -0.1043518  0.0112602  -9.267  < 2e-16 ***
## cpAtypical Angina      -0.0147564  0.0224181  -0.658  0.51054
## cpNon-anginal Pain     -0.0020487  0.0205923  -0.099  0.92077
## cpTypical Angina       -0.1089011  0.0204063  -5.337 1.17e-07 ***
## trestbps               -0.0008863  0.0003068  -2.888  0.00396 **
## chol                   -0.0001561  0.0001016  -1.536  0.12496
## fbs> 120 mg/dL          0.0126442  0.0146296   0.864  0.38764
## restecgHypertrophy      0.0032256  0.0423581   0.076  0.93931
## restecgNormal          -0.0194149  0.0102213  -1.899  0.05779 .
```

```
## thalach                 0.0007671  0.0002760   2.779  0.00555 **
## exangYes               -0.0548551  0.0124239  -4.415 1.12e-05 ***
## oldpeak                -0.0064945  0.0065300  -0.995  0.32019
## slopeFlat              -0.0643895  0.0123877  -5.198 2.44e-07 ***
## slopeUpsloping         -0.0038055  0.0239576  -0.159  0.87382
## ca                     -0.0597964  0.0060617  -9.865  < 2e-16 ***
## thalNormal             -0.0342524  0.0618652  -0.554  0.57993
## thalReversible Defect   0.0026655  0.0197643   0.135  0.89275
## moutYesMOut            -0.0552263  0.0238470  -2.316  0.02077 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1546 on 1005 degrees of freedom
## Multiple R-squared:  0.4786, Adjusted R-squared:  0.4687
## F-statistic: 48.55 on 19 and 1005 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(transm)
```

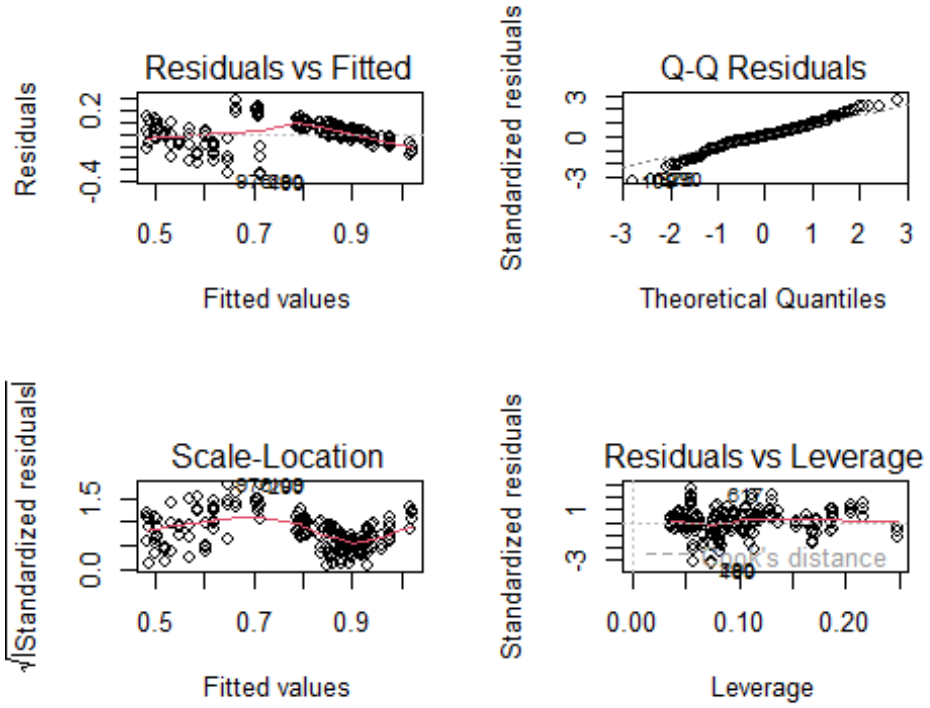



```
transm45 <- lm(formula= sqrt(target) ~ ., data =df2[,1:15])
summary(transm45)
```

```
##
## Call:
## lm(formula = sqrt(target) ~ ., data = df2[, 1:15])
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.33968 -0.04699  0.00134  0.05748  0.27584
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.5226489  0.2092901   2.497  0.01347 *
## age                     0.0039238  0.0027694   1.417  0.15837
## sexMale                -0.0909412  0.0209844  -4.334 2.50e-05 ***
## cpAtypical Angina      -0.0351551  0.0444358  -0.791  0.42996
## cpNon-anginal Pain     -0.0360341  0.0432287  -0.834  0.40569
## cpTypical Angina       -0.2490174  0.0422397  -5.895 1.96e-08 ***
## trestbps                0.0012262  0.0007832   1.566  0.11931
## chol                    0.0004312  0.0002063   2.090  0.03811 *
## fbs> 120 mg/dL          0.0241110  0.0468511   0.515  0.60748
## restecgNormal          -0.0183035  0.0195222  -0.938  0.34979
## thalach                 0.0003352  0.0006562   0.511  0.61011
## exangYes               -0.1396832  0.0299207  -4.668 6.13e-06 ***
## oldpeak                 0.0371698  0.0161648   2.299  0.02270 *
## slopeFlat              -0.0485484  0.0280012  -1.734  0.08477 .
## slopeUpsloping          0.0137102  0.0568448   0.241  0.80970
## ca                     -0.0970618  0.0265396  -3.657  0.00034 ***
## thalReversible Defect  0.0309836  0.0419084   0.739  0.46073
## moutYesMOut            -0.2518596  0.0560734  -4.492 1.30e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1089 on 170 degrees of freedom
## Multiple R-squared:  0.6884, Adjusted R-squared:  0.6572
## F-statistic: 22.09 on 17 and 170 DF,  p-value: < 2.2e-16
```

```r
par(mfrow = c(2, 2))
plot(transm45)
```



Both cases where we apply the transformation are worse, so we confirm that the value lambda being next to 1 makes the transformation not necessary.

Binary Regression Model

We begin by creating two new datasets: one from the original dataset and another subset where the age is greater than 45. In these datasets, we transform the numerical target variable into a binary variable. Specifically, we assign a value of 1 to the target variable if it is greater than or equal to 0.5, and a value of 0 otherwise.

```
dfb <- df.original[,c(1:14)]
dfb$target[which(dfb$target>=0.5)] <- 1
dfb$target[which(dfb$target<0.5)] <- 0

head(dfb$target, 100)
```

```
##   [1] 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 1 0 1 1 0 1 1 1 1 0 1 0 0 0 0 1
0 0 1 0 1
##  [38] 1 1 0 1 1 0 0 1 1 1 0 1 0 1 0 1 0 0 0 1 1 0 0 1 1 0 1 1 0 1 0 1
0 0 0 0 0
##  [75] 0 1 1 0 1 1 0 0 0 1 1 1 1 0 0 0 1 1 0 0 1 1 1 0 0 1
```

```
dfb45 <- df2[,c(1:14)]
dfb45$target[which(dfb45$target>=0.5)] <- 1
dfb45$target[which(dfb45$target<0.5)] <- 0
```

```r
head(dfb45$target, 100)
```

```
##   [1] 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 0 0 0 1 1 0 1 1 1 0 0 1 1 1
1 1 1 0 1
##  [38] 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1
1 1 1 1 1
##  [75] 1 0 1 0 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0
```

We then proceed to split the datasets into training and testing, with 80-20 split respectively.

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following object is masked from 'package:car':
##
##     recode
```

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

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

```r
# Divide the data set in training and test split (80-20 split)
set.seed(123)  # Ensure the results are repeatable
sample_size <- floor(0.8 * nrow(dfb))
train_index <- sample(seq_len(nrow(dfb)), size = sample_size)
train_data <- dfb[train_index, ]
test_data <- dfb[-train_index, ]
```

```r
library(dplyr)

# Divide the data set in training and test split (80-20 split)
set.seed(123)  # Ensure the results are repeatable
sample_size45 <- floor(0.8 * nrow(dfb45))
train_index45 <- sample(seq_len(nrow(dfb45)), size = sample_size45)
train_data45 <- dfb45[train_index45, ]
test_data45 <- dfb45[-train_index45, ]
```

In this analysis, we begin by building a generalized linear model (GLM) using logistic regression to predict the binary target variable. We use all available predictors in the

training dataset. The binomial family is chosen to perform logistic regression, suitable for binary classification tasks.

After fitting the model (mB), we generate a detailed summary to examine the model's coefficients, their statistical significance, and overall fit. This summary provides insights into how each predictor influences the target variable and helps assess the model's performance through various diagnostic metrics.

Variance Inflation Factor

We then calculate the Variance Inflation Factor (VIF) for each predictor to check for multicollinearity. High VIF values indicate that some predictors are highly correlated with each other, which can affect the model's stability and interpretation. Identifying and addressing high VIF values is crucial for ensuring a robust and reliable model.

```
mB <- glm(formula= target ~ ., data = train_data, family=binomial)
summary(mB)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial, data = train_data)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              6.665860   1.698266   3.925 8.67e-05 ***
## age                      0.011338   0.014584   0.777 0.436906
## sexMale                 -2.282274   0.305698  -7.466 8.28e-14 ***
## cpAtypical Angina       -1.058541   0.436423  -2.425 0.015288 *
## cpNon-anginal Pain      -0.245929   0.382346  -0.643 0.520087
## cpTypical Angina        -2.067178   0.384779  -5.372 7.77e-08 ***
## trestbps                -0.023525   0.006564  -3.584 0.000339 ***
## chol                    -0.006585   0.002396  -2.748 0.005996 **
## fbs> 120 mg/dL           0.426367   0.343832   1.240 0.214959
## restecgHypertrophy      -0.884821   1.456111  -0.608 0.543413
## restecgNormal           -0.203696   0.222750  -0.914 0.360475
## thalach                  0.019888   0.006479   3.070 0.002144 **
## exangYes                -0.970039   0.259671  -3.736 0.000187 ***
## oldpeak                 -0.349499   0.129693  -2.695 0.007043 **
## slopeFlat               -1.350883   0.271742  -4.971 6.65e-07 ***
## slopeUpsloping          -0.791823   0.504129  -1.571 0.116258
## ca                      -1.246640   0.156235  -7.979 1.47e-15 ***
## thalNormal              -1.668368   2.427269  -0.687 0.491866
## thalReversible Defect   -0.558979   0.431338  -1.296 0.195004
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1136.59  on 819  degrees of freedom
## Residual deviance:  566.34  on 801  degrees of freedom
```

```
## AIC: 604.34
##
## Number of Fisher Scoring iterations: 6
```

```
vif(mB)
```

```
##              GVIF Df GVIF^(1/(2*Df))
## age      1.409402  1        1.187182
## sex      1.470315  1        1.212565
## cp       1.617846  3        1.083485
## trestbps 1.272624  1        1.128107
## chol     1.286370  1        1.134183
## fbs      1.293883  1        1.137490
## restecg  1.153860  2        1.036426
## thalach  1.466693  1        1.211071
## exang    1.258715  1        1.121925
## oldpeak  1.491491  1        1.221266
## slope    1.854166  2        1.166909
## ca       1.299452  1        1.139935
## thal     1.363110  2        1.080520
```

Several predictors are highly significant (e.g., sexMale, cpAtypical Angina, trestbps, chol, exangYes, slopeFlat, ca), suggesting they have a strong influence on the target variable.

All VIF values are below 2, indicating that multicollinearity is not a significant concern in this model.

```
mB45 <- glm(formula= target ~ ., data = train_data45, family = binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(mB45)
```

```
##
## Call:
## glm(formula = target ~ ., family = binomial, data = train_data45)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -3.954e+02  1.218e+06   0.000    1.000
## age                    1.604e+01  1.373e+04   0.001    0.999
## sexMale               -1.117e+02  1.277e+05  -0.001    0.999
## cpAtypical Angina      6.626e+00  4.568e+05   0.000    1.000
## cpNon-anginal Pain    -8.036e+01  1.505e+05  -0.001    1.000
## cpTypical Angina      -1.769e+02  1.635e+05  -0.001    0.999
## trestbps              -1.077e+00  2.398e+03   0.000    1.000
## chol                   4.012e-01  1.658e+03   0.000    1.000
## fbs> 120 mg/dL         3.384e+01  2.871e+05   0.000    1.000
## restecgNormal         -9.299e+00  1.876e+05   0.000    1.000
```

```
## thalach                 1.453e+00  6.117e+03   0.000    1.000
## exangYes               -1.437e+02  8.873e+04  -0.002    0.999
## oldpeak                 9.553e+00  7.123e+04   0.000    1.000
## slopeFlat              -7.847e+01  1.252e+05  -0.001    0.999
## slopeUpsloping         -7.886e+01  2.166e+05   0.000    1.000
## ca                     -8.558e+01  1.581e+05  -0.001    1.000
## thalReversible Defect -1.677e+02  3.622e+05   0.000    1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.8652e+02  on 149  degrees of freedom
## Residual deviance: 1.3753e-08  on 133  degrees of freedom
## AIC: 34
##
## Number of Fisher Scoring iterations: 25

vif(mB45)

##                 GVIF Df GVIF^(1/(2*Df))
## age        31.643189  1        5.625228
## sex         8.213396  1        2.865902
## cp       4823.563501  3        4.110500
## trestbps   17.626987  1        4.198450
## chol       19.272888  1        4.390090
## fbs         3.614618  1        1.901215
## restecg   110.620417  1       10.517624
## thalach    59.156941  1        7.691355
## exang      10.284014  1        3.206870
## oldpeak    74.552201  1        8.634362
## slope      54.055559  2        2.711503
## ca         94.144767  1        9.702823
## thal        7.725464  1        2.779472
```

The model shows an extremely low residual deviance and high standard errors for coefficients, suggesting that the model is overfitting the training data. The high number of Fisher Scoring iterations (25) indicates potential issues with model convergence.

None of the predictors are statistically significant, as indicated by their p-values close to 1.

High VIF values for several predictors indicate severe multicollinearity, which can cause instability in the coefficient estimates and make it difficult to determine the individual effect of each predictor.

For these reasons, we'll continue with only the original dataset and its training and test data.

ANOVA tests

As our initial model, we selected an additive model that includes the variables ca, oldpeak, and thalach, based on their strong correlation with the target variable, as seen in the correlation plot of the first deliverable.

```
initial_model <- glm(target ~ ca + oldpeak + thalach, data = train_data,
family = binomial)
```

We then try to build some other models, considering the insights gleaned from summarizing the findings of model mB, which highlighted the most significant variables. Initially, we focus on the significant numerical variables and gradually augment the model complexity by incorporating additional influential factors. Subsequently, we conduct an ANOVA test to determine the superior model among them.

```
initial_model2 <- glm(target ~ trestbps + ca, data = train_data, family =
binomial)

initial_model2.2 <- glm(target ~ ca + oldpeak + thalach + sex + exang ,
data = train_data, family = binomial)

initial_model3 <- glm(target ~ trestbps + ca + cp + sex + exang + slope,
data = train_data, family = binomial)

initial_model3.2 <- glm(target ~ oldpeak + thalach + ca + cp + sex +
exang + slope, data = train_data, family = binomial)

initial_model4 <- glm(target ~ cp + trestbps + slope + ca + sex + exang +
chol + thalach + oldpeak, data = train_data, family = binomial)


anova(initial_model, initial_model2)

## Analysis of Deviance Table
##
## Model 1: target ~ ca + oldpeak + thalach
## Model 2: target ~ trestbps + ca
##   Resid. Df Resid. Dev Df Deviance
## 1       816     779.91
## 2       817     943.91 -1     -164

anova(initial_model, initial_model2.2)

## Analysis of Deviance Table
##
## Model 1: target ~ ca + oldpeak + thalach
## Model 2: target ~ ca + oldpeak + thalach + sex + exang
##   Resid. Df Resid. Dev Df Deviance
```

```
## 1       816     779.91
## 2       814     676.31  2    103.6
```

```
anova(initial_model2.2, initial_model3)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ ca + oldpeak + thalach + sex + exang
## Model 2: target ~ trestbps + ca + cp + sex + exang + slope
##   Resid. Df Resid. Dev Df Deviance
## 1       814     676.31
## 2       810     599.39  4   76.914
```

```
anova(initial_model3.2, initial_model3)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ oldpeak + thalach + ca + cp + sex + exang + slope
## Model 2: target ~ trestbps + ca + cp + sex + exang + slope
##   Resid. Df Resid. Dev Df Deviance
## 1       809     595.35
## 2       810     599.39 -1  -4.0428
```

```
anova(initial_model3.2, initial_model4)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ oldpeak + thalach + ca + cp + sex + exang + slope
## Model 2: target ~ cp + trestbps + slope + ca + sex + exang + chol +
thalach +
##     oldpeak
##   Resid. Df Resid. Dev Df Deviance
## 1       809     595.35
## 2       807     572.66  2    22.69
```

In each ANOVA test, we select the model with the lowest residual deviance to proceed to the next stage. Initial_model4 seems to be the best model. However, despite its superior performance, it demands a larger set of predictors compared to other models, and its improvement in residual deviance isn't substantial. Therefore, we opt to retain initial_model3, which strikes a favorable balance between the number of predictors and the reduction in residual deviance.

Stepwise with Akaike criteria

We use stepwise methodology to get the most optimize linear regression model with Akaike criteria.

```
model <- glm(target ~ .,  data = train_data, family=binomial)

m_step_mB = step(model)
```

```
## Start:  AIC=604.34
## target ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach +
##     exang + oldpeak + slope + ca + thal
##
##            Df Deviance    AIC
## - restecg   2   567.49 601.49
## - thal      2   568.28 602.28
## - age       1   566.95 602.95
## - fbs       1   567.90 603.90
## <none>          566.34 604.34
## - chol      1   573.61 609.61
## - oldpeak   1   573.89 609.89
## - thalach   1   576.24 612.24
## - trestbps  1   579.69 615.69
## - exang     1   580.21 616.21
## - slope     2   592.59 626.59
## - cp        3   623.27 655.27
## - sex       1   636.46 672.46
## - ca        1   649.61 685.61
##
## Step:  AIC=601.49
## target ~ age + sex + cp + trestbps + chol + fbs + thalach + exang +
##     oldpeak + slope + ca + thal
##
##            Df Deviance    AIC
## - thal      2   569.42 599.42
## - age       1   568.08 600.08
## - fbs       1   568.96 600.96
## <none>          567.49 601.49
## - oldpeak   1   575.19 607.19
## - chol      1   575.78 607.78
## - thalach   1   577.82 609.82
## - exang     1   581.01 613.01
## - trestbps  1   582.26 614.26
## - slope     2   595.35 625.35
## - cp        3   624.14 652.14
## - sex       1   638.70 670.70
## - ca        1   651.99 683.99
##
## Step:  AIC=599.42
## target ~ age + sex + cp + trestbps + chol + fbs + thalach + exang +
##     oldpeak + slope + ca
##
##            Df Deviance    AIC
## - age       1   569.99 597.99
## <none>          569.42 599.42
## - fbs       1   572.03 600.03
## - oldpeak   1   576.80 604.80
## - chol      1   578.30 606.30
## - thalach   1   579.00 607.00
```

```
## - exang     1   582.87 610.87
## - trestbps  1   583.58 611.58
## - slope     2   596.37 622.37
## - cp        3   624.51 648.51
## - sex       1   638.81 666.81
## - ca        1   654.73 682.73
##
## Step:  AIC=597.99
## target ~ sex + cp + trestbps + chol + fbs + thalach + exang +
##     oldpeak + slope + ca
##
##            Df Deviance    AIC
## <none>          569.99 597.99
## - fbs       1   572.66 598.66
## - oldpeak   1   577.79 603.79
## - chol      1   578.50 604.50
## - thalach   1   579.10 605.10
## - trestbps  1   583.78 609.78
## - exang     1   583.93 609.93
## - slope     2   596.58 620.58
## - cp        3   626.06 648.06
## - sex       1   641.79 667.79
## - ca        1   657.43 683.43
```

```
anova(m_step_mB, initial_model3)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ sex + cp + trestbps + chol + fbs + thalach + exang +
##     oldpeak + slope + ca
## Model 2: target ~ trestbps + ca + cp + sex + exang + slope
##   Resid. Df Resid. Dev Df Deviance
## 1       806     569.99
## 2       810     599.39 -4  -29.404
```

```
summary(m_step_mB)
```

```
##
## Call:
## glm(formula = target ~ sex + cp + trestbps + chol + fbs + thalach +
##     exang + oldpeak + slope + ca, family = binomial, data =
train_data)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         6.748112   1.443955   4.673 2.96e-06 ***
## sexMale            -2.227561   0.293829  -7.581 3.43e-14 ***
## cpAtypical Angina  -0.996955   0.428145  -2.329 0.019883 *
## cpNon-anginal Pain -0.212480   0.376960  -0.564 0.572980
## cpTypical Angina   -1.990601   0.372594  -5.343 9.17e-08 ***
## trestbps           -0.022277   0.006114  -3.644 0.000269 ***
```

```
## chol               -0.006805   0.002279  -2.986 0.002824 **
## fbs> 120 mg/dL      0.532635   0.327677   1.625 0.104059
## thalach             0.017678   0.005997   2.948 0.003202 **
## exangYes           -0.966729   0.258096  -3.746 0.000180 ***
## oldpeak            -0.348286   0.127150  -2.739 0.006159 **
## slopeFlat          -1.330648   0.266683  -4.990 6.05e-07 ***
## slopeUpsloping     -0.665368   0.495616  -1.343 0.179431
## ca                 -1.232290   0.150315  -8.198 2.44e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1136.59  on 819  degrees of freedom
## Residual deviance:  569.99  on 806  degrees of freedom
## AIC: 597.99
##
## Number of Fisher Scoring iterations: 6
```

The top-performing model identified through stepwise selection employs twice the number of predictors compared to our chosen initial_model3. However, upon examining its summary, we observe that the variables selected in initial_model3 remain the most significant, denoted by their significance level (***). Despite the doubling of predictors, these variables maintain their importance. Therefore, to enhance the model further while avoiding excessive predictor inclusion, we aim to refine initial_model3 by exploring potential interactions between variables.

```
final_model <- initial_model3

vif(final_model)
```

```
##              GVIF Df GVIF^(1/(2*Df))
## trestbps 1.085623  1        1.041933
## ca       1.109936  1        1.053535
## cp       1.352857  3        1.051660
## sex      1.200808  1        1.095814
## exang    1.179108  1        1.085868
## slope    1.199385  2        1.046501
```

Once again, we see low VIF values which confirm that multicollinearity is not a significant issue, ensuring the stability and reliability of the final_model coefficients.

Interactions between factors and interactions between factors and covariate

We now proceed to consider interactions between factors and between factors and covariate.

```
# Initial model without interaction
model_no_interaction <- glm(formula = target ~ sex + exang + cp +
trestbps + slope + ca, family = binomial, data = train_data)
```

```r
# Model with interaction
model_with_interaction <- glm(formula = target ~ sex * exang + cp +
trestbps + slope + ca, family = binomial, data = train_data)

# Compare models using ANOVA
anova(model_no_interaction, model_with_interaction)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ sex + exang + cp + trestbps + slope + ca
## Model 2: target ~ sex * exang + cp + trestbps + slope + ca
##   Resid. Df Resid. Dev Df Deviance
## 1       810     599.39
## 2       809     594.65  1    4.742
```

We see a slight improvement when considering the interaction between factors sex and exang.

```r
# Initial model without interaction
model_no_interaction2 <- glm(formula = target ~ sex + cp + exang +
trestbps + slope + ca, family = binomial, data = train_data)

# Model with interaction
model_with_interaction2 <- glm(formula = target ~ sex * cp + exang +
trestbps + slope + ca, family = binomial, data = train_data)

# Compare models using ANOVA
anova(model_no_interaction2, model_with_interaction2)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ sex + cp + exang + trestbps + slope + ca
## Model 2: target ~ sex * cp + exang + trestbps + slope + ca
##   Resid. Df Resid. Dev Df Deviance
## 1       810     599.39
## 2       807     597.69  3    1.698
```

The enhancement observed by incorporating the interaction between the factors sex and cp is minimal.

```r
# Initial model without interaction
model_no_interaction3 <- glm(formula = target ~ sex + slope + cp +
trestbps + exang + ca, family = binomial, data = train_data)

# Model with interaction
model_with_interaction3 <- glm(formula = target ~ sex * slope + cp +
trestbps + exang + ca, family = binomial, data = train_data)

# Compare models using ANOVA
anova(model_no_interaction3, model_with_interaction3)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ sex + slope + cp + trestbps + exang + ca
## Model 2: target ~ sex * slope + cp + trestbps + exang + ca
##   Resid. Df Resid. Dev Df Deviance
## 1       810     599.39
## 2       808     597.79  2   1.5992
```

We see the same minimal improvement as in the previous example.

```
# Initial model without interaction
model_no_interaction_covariate <- glm(formula = target ~ sex + ca + exang
+ trestbps + slope + cp, family = binomial, data = train_data)

# Model with interaction
model_with_interaction_covariate <- glm(formula = target ~ sex * ca +
exang + trestbps + slope + cp, family = binomial, data = train_data)

# Compare models using ANOVA
anova(model_no_interaction_covariate, model_with_interaction_covariate)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ sex + ca + exang + trestbps + slope + cp
## Model 2: target ~ sex * ca + exang + trestbps + slope + cp
##   Resid. Df Resid. Dev Df Deviance
## 1       810     599.39
## 2       809     598.64  1  0.74887
```

We see the same minimal improvement as in the previous example. We see even less minimal improvement than in the two previous example.

```
# Initial model without interaction
model_no_interaction_covariate2 <- glm(formula = target ~ sex + trestbps
+ exang + ca + slope + cp, family = binomial, data = train_data)

# Model with interaction
model_with_interaction_covariate2 <- glm(formula = target ~ sex *
trestbps + exang + ca + slope + cp, family = binomial, data = train_data)

# Compare models using ANOVA
anova(model_no_interaction_covariate2, model_with_interaction_covariate2)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ sex + trestbps + exang + ca + slope + cp
## Model 2: target ~ sex * trestbps + exang + ca + slope + cp
##   Resid. Df Resid. Dev Df Deviance
## 1       810     599.39
## 2       809     598.61  1  0.78616
```

We see the same minimal improvement as in the previous example.

```
# Initial model without interaction
model_no_interaction_covariate3 <- glm(formula = target ~ exang +
trestbps + sex + ca + slope + cp, family = binomial, data = train_data)

# Model with interaction
model_with_interaction_covariate3 <- glm(formula = target ~ exang *
trestbps + sex + ca + slope + cp, family = binomial, data = train_data)

# Compare models using ANOVA
anova(model_no_interaction_covariate3, model_with_interaction_covariate3)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ exang + trestbps + sex + ca + slope + cp
## Model 2: target ~ exang * trestbps + sex + ca + slope + cp
##   Resid. Df Resid. Dev Df Deviance
## 1       810     599.39
## 2       809     593.45  1   5.9379
```

We see a slight improvement when considering the interaction between factor exang and covariate trestbps.

```
# Initial model without interaction
model_no_interaction_covariate4 <- glm(formula = target ~ exang + cp +
sex + ca + slope + trestbps, family = binomial, data = train_data)

# Model with interaction
model_with_interaction_covariate4 <- glm(formula = target ~ exang * cp +
sex + ca + slope + trestbps, family = binomial, data = train_data)

# Compare models using ANOVA
anova(model_no_interaction_covariate4, model_with_interaction_covariate4)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ exang + cp + sex + ca + slope + trestbps
## Model 2: target ~ exang * cp + sex + ca + slope + trestbps
##   Resid. Df Resid. Dev Df Deviance
## 1       810     599.39
## 2       807     590.81  3    8.579
```

We see a slight improvement, better than the previous one, when considering the interaction between factor exang and covariate cp.

After seeing the improvement of single interactions, we want to see if the models improve when having two interactions.

```
# Model with interaction
model_with_interaction_double <- glm(formula = target ~ exang * (cp +
```

```
trestbps) + ca + slope + sex, family = binomial, data = train_data)

# Compare models using ANOVA
anova(model_with_interaction_covariate3, model_with_interaction_double)

## Analysis of Deviance Table
##
## Model 1: target ~ exang * trestbps + sex + ca + slope + cp
## Model 2: target ~ exang * (cp + trestbps) + ca + slope + sex
##   Resid. Df Resid. Dev Df Deviance
## 1       809     593.45
## 2       806     585.20  3   8.2516

anova(model_with_interaction_covariate4, model_with_interaction_double)

## Analysis of Deviance Table
##
## Model 1: target ~ exang * cp + sex + ca + slope + trestbps
## Model 2: target ~ exang * (cp + trestbps) + ca + slope + sex
##   Resid. Df Resid. Dev Df Deviance
## 1       807     590.81
## 2       806     585.20  1   5.6105

anova(model_no_interaction, model_with_interaction_double)

## Analysis of Deviance Table
##
## Model 1: target ~ sex + exang + cp + trestbps + slope + ca
## Model 2: target ~ exang * (cp + trestbps) + ca + slope + sex
##   Resid. Df Resid. Dev Df Deviance
## 1       810     599.39
## 2       806     585.20  4    14.19
```

We see that considering the interactions between factor exang and factors trestbps and cp the model improves much more compared to a single interaction.

We now try different models adding other interactions with exang.

```
# Model with interaction
model_with_interaction_triple <- glm(formula = target ~ exang * (cp +
trestbps + sex) + ca + slope, family = binomial, data = train_data)

model_with_interaction_q <- glm(formula = target ~ exang * (cp + trestbps
+ sex + ca) + slope, family = binomial, data = train_data)

model_with_interaction_c <- glm(formula = target ~ exang * (cp + trestbps
+ sex + ca + slope), family = binomial, data = train_data)

# Compare models using ANOVA
anova(model_with_interaction_double, model_with_interaction_triple)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ exang * (cp + trestbps) + ca + slope + sex
## Model 2: target ~ exang * (cp + trestbps + sex) + ca + slope
##   Resid. Df Resid. Dev Df Deviance
## 1       806     585.20
## 2       805     583.65  1   1.5571
```

```
anova(model_with_interaction_triple, model_with_interaction_q)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ exang * (cp + trestbps + sex) + ca + slope
## Model 2: target ~ exang * (cp + trestbps + sex + ca) + slope
##   Resid. Df Resid. Dev Df Deviance
## 1       805     583.65
## 2       804     579.41  1   4.2371
```

```
anova(model_with_interaction_q, model_with_interaction_c)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ exang * (cp + trestbps + sex + ca) + slope
## Model 2: target ~ exang * (cp + trestbps + sex + ca + slope)
##   Resid. Df Resid. Dev Df Deviance
## 1       804     579.41
## 2       802     565.76  2   13.646
```

```
anova(final_model, model_with_interaction_c)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ trestbps + ca + cp + sex + exang + slope
## Model 2: target ~ exang * (cp + trestbps + sex + ca + slope)
##   Resid. Df Resid. Dev Df Deviance
## 1       810     599.39
## 2       802     565.76  8   33.629
```

```
anova(m_step_mB, model_with_interaction_c)
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ sex + cp + trestbps + chol + fbs + thalach + exang +
##     oldpeak + slope + ca
## Model 2: target ~ exang * (cp + trestbps + sex + ca + slope)
##   Resid. Df Resid. Dev Df Deviance
## 1       806     569.99
## 2       802     565.76  4    4.225
```

We see that model_with_interaction_c improves much, not only with respect to final_model but also to the best found model m_step_mB, which had 10 predictors.

As model_with_interaction_c has 5 interactions, we want to see how this affects multicollinearity.

```
vif(model_with_interaction_c)

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

##                      GVIF Df GVIF^(1/(2*Df))
## exang          147.267627  1       12.135387
## cp               2.167466  3        1.137606
## trestbps         1.432665  1        1.196940
## sex              1.588246  1        1.260256
## ca               1.216816  1        1.103094
## slope            1.695345  2        1.141076
## exang:cp        14.301011  3        1.557977
## exang:trestbps  91.624358  1        9.572061
## exang:sex        6.667194  1        2.582091
## exang:ca         1.604391  1        1.266646
## exang:slope      6.164812  2        1.575723
```

We see that exang has now a value over 10 caused by the interactions with trestbps and sex. We now proceed to substract this variables and check again for its VIF values.

```
# Model with interaction
model_with_interaction_c2 <- glm(formula = target ~ exang * (cp + ca +
slope) + trestbps + sex, family = binomial, data = train_data)

anova(final_model, model_with_interaction_c2)

## Analysis of Deviance Table
##
## Model 1: target ~ trestbps + ca + cp + sex + exang + slope
## Model 2: target ~ exang * (cp + ca + slope) + trestbps + sex
##   Resid. Df Resid. Dev Df Deviance
## 1       810     599.39
## 2       804     575.60  6   23.787
```

We see that removing the two interactions does not worsen the model much, and still performs better than the final model.

```
vif(model_with_interaction_c2)

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

##                  GVIF Df GVIF^(1/(2*Df))
## exang       12.346984  1        3.513828
## cp           2.207359  3        1.141070
## ca           1.197713  1        1.094401
## slope        1.679938  2        1.138475
## trestbps     1.158835  1        1.076492
```

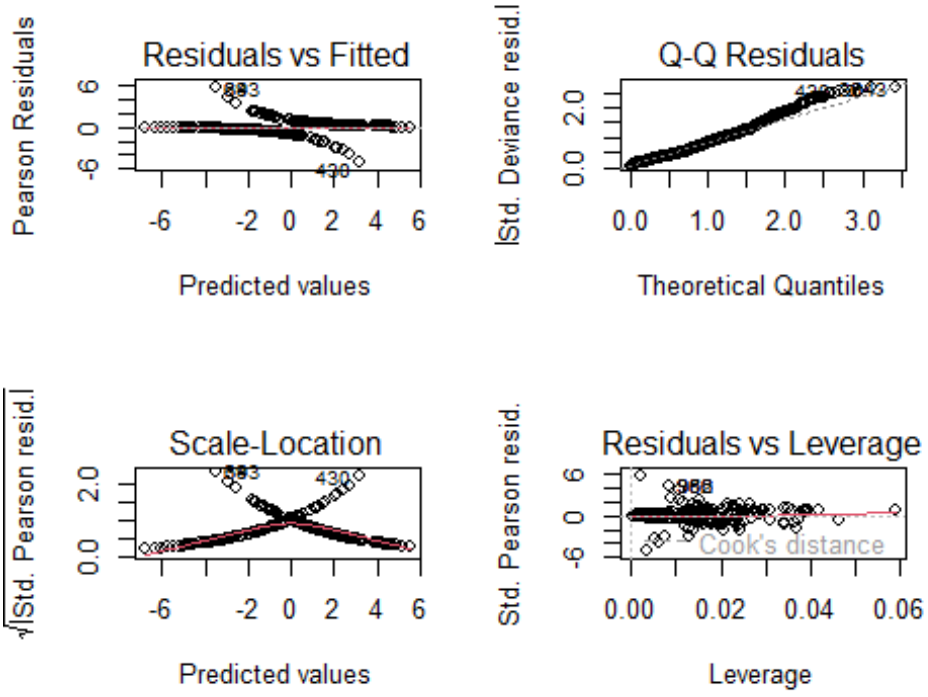
```
## sex           1.254269  1        1.119942
## exang:cp     14.475258  3        1.561125
## exang:ca      1.516720  1        1.231552
## exang:slope   4.270621  2        1.437549
```

The VIF values observed are consistently low across all predictors. Although exang shows a slightly higher value of three, this doesn't necessarily imply that multicollinearity is a concern for the model.

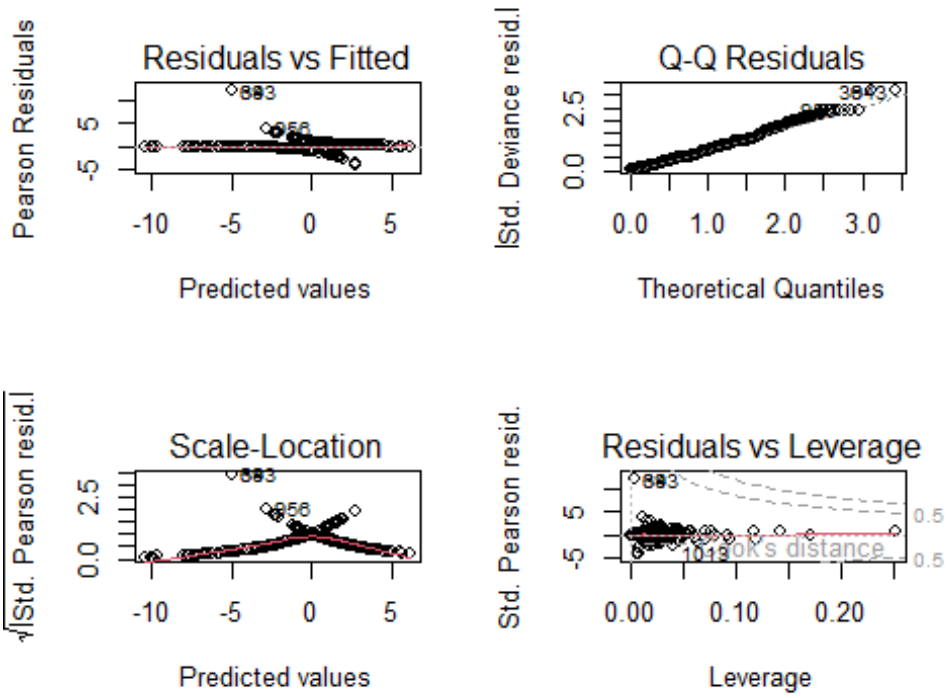
Diagnostic plots for the final model

We've opted to assess both the final_model and the model_with_interaction_c2. Despite the latter's superior performance, we're conducting this comparison to ensure that introducing interactions doesn't introduce any unforeseen issues.

```
# Diagnostic plots for the final model with two three interaction terms
par(mfrow = c(2, 2))
plot(final_model)
```




```
plot(model_with_interaction_c2)
```



Residuals vs Fitted: The residuals display a funnel shape, indicating heteroscedasticity. This means the variance of the residuals is not constant across levels of the fitted values, violating the homoscedasticity assumption.

Q-Q Plot: The residuals deviate from the 45-degree line, particularly in the tails, suggesting that they do not follow a normal distribution. This is an indication that the normality assumption of the residuals is violated.

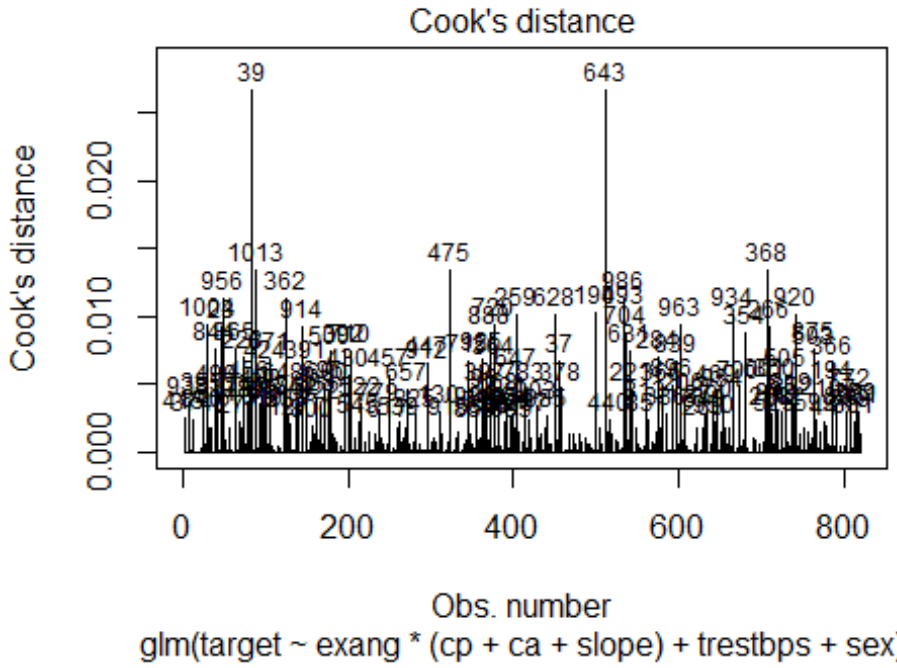
Scale-Location: The spread of the residuals increases with the predicted values, reinforcing the presence of heteroscedasticity. The residuals are not evenly spread, indicating issues with the model's assumptions.

Residuals vs. Leverage Plot: This plot shows the relationship between standardized residuals and leverage. Most data points are clustered with low leverage and residuals near zero. A few points have higher leverage or residuals, indicating potential influential observations, but no points exceed Cook's distance threshold significantly.

Influential points

```
# Identify specific influential observations
influential_points <-
which(influence.measures(model_with_interaction_c2)$is.inf)

# Highlight influential points in plots
plot(model_with_interaction_c2, which = 4, id.n =
length(influential_points))
```



Cook's Distance Plot: This plot identifies influential data points in the model. Observations such as 39, 643, and 475 have relatively higher Cook's distances, suggesting these points have a greater influence on the model's coefficients. Overall, most points have low Cook's distances, indicating minimal individual influence.

```
# Avaluació del model amb les dades de test
predictions <- predict(final_model, newdata = test_data, type =
"response")
predicted_classes <- ifelse(predictions > 0.5, 1, 0)
conf_matrix <- table(Predicted = predicted_classes, Actual =
test_data$target)

# Calcular mètriques d'avaluació (Accuracy, Sensitivity, Specificity)
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
sensitivity <- conf_matrix[2, 2] / sum(conf_matrix[2, ])
specificity <- conf_matrix[1, 1] / sum(conf_matrix[1, ])

# Mostrar la matriu de confusió i les mètriques d'avaluació
print(conf_matrix)

##          Actual
## Predicted  0  1
##         0 79 11
##         1 16 99

cat("Accuracy: ", accuracy, "\n")

## Accuracy:  0.8682927
```

```r
cat("Sensitivity: ", sensitivity, "\n")
```

```
## Sensitivity:  0.8608696
```

```r
cat("Specificity: ", specificity, "\n")
```

```
## Specificity:  0.8777778
```

The confusion matrix presents the model's performance. Accuracy: 86.83% Sensitivity (True Positive Rate): 86.09% Specificity (True Negative Rate): 87.78% The model shows balanced performance in predicting both classes with high accuracy, sensitivity, and specificity.

```r
# Avaluació del model amb les dades de test
predictions <- predict(model_with_interaction_c2, newdata = test_data,
type = "response")
predicted_classes <- ifelse(predictions > 0.5, 1, 0)
conf_matrix <- table(Predicted = predicted_classes, Actual =
test_data$target)

# Calcular mètriques d'avaluació (Accuracy, Sensitivity, Specificity)
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
sensitivity <- conf_matrix[2, 2] / sum(conf_matrix[2, ])
specificity <- conf_matrix[1, 1] / sum(conf_matrix[1, ])

# Mostrar la matriu de confusió i les mètriques d'avaluació
print(conf_matrix)
```

```
##          Actual
## Predicted   0   1
##         0  79   9
##         1  16 101
```

```r
cat("Accuracy: ", accuracy, "\n")
```

```
## Accuracy:  0.8780488
```

```r
cat("Sensitivity: ", sensitivity, "\n")
```

```
## Sensitivity:  0.8632479
```

```r
cat("Specificity: ", specificity, "\n")
```

```
## Specificity:  0.8977273
```

The confusion matrix presents the model's performance. Accuracy: 87.80% Sensitivity (True Positive Rate): 86.32% Specificity (True Negative Rate): 89.77% The model shows balanced performance in predicting both classes with high accuracy, sensitivity, and specificity.