DSC 423: Data Analysis & Regression

Assignment 9: Advanced Regression Models

```
Name: Kushal Navghare
Student ID: 2116916
Honor Statement: I, Kushal Navghare, assure that I have completed this work independently.
The solutions given are entirely my own work.
```

1. Previously you created a model using the PISA dataset. Build a model again, this time...

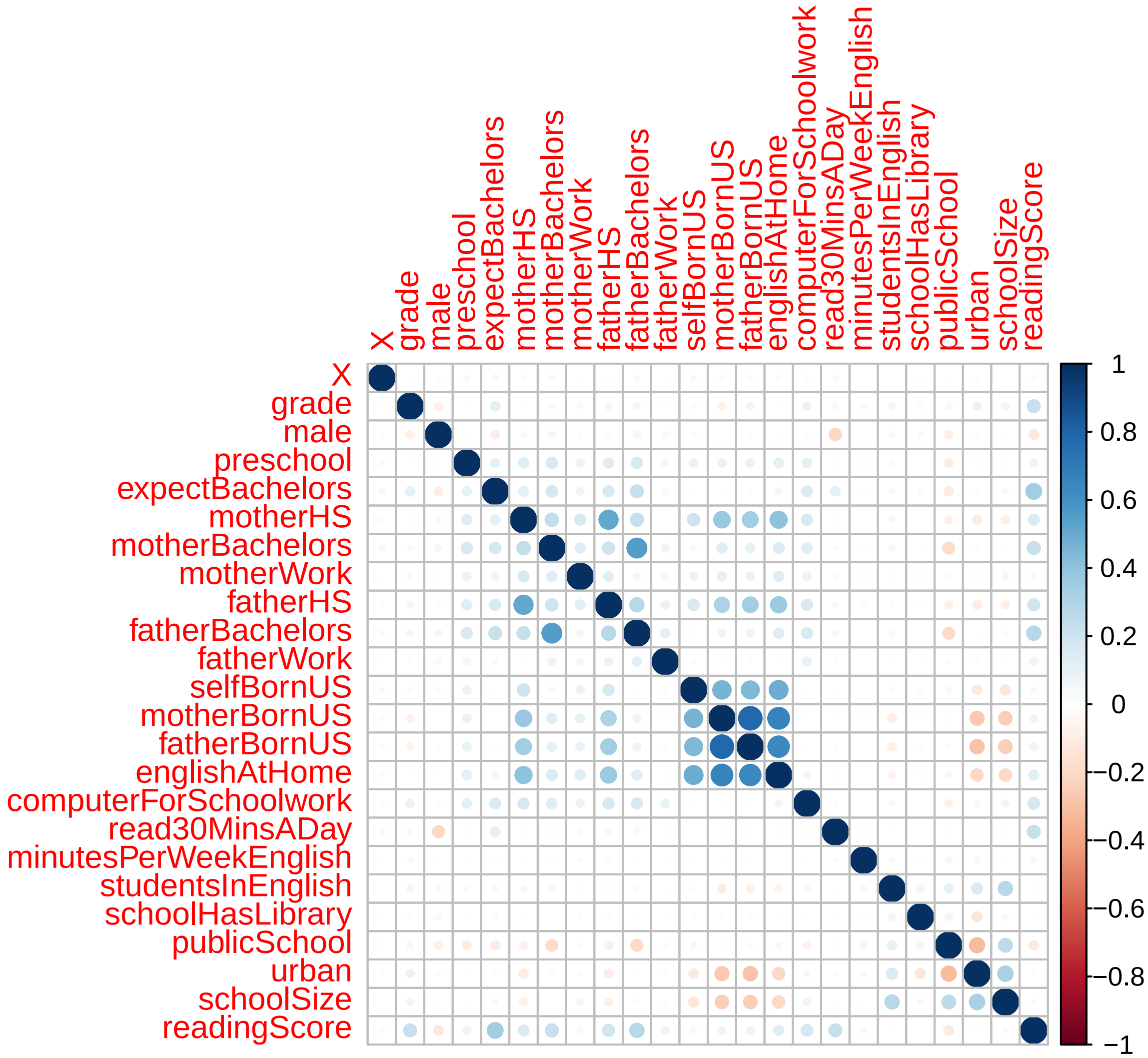
a. (10 points) Use Ridge regression and present your model along with appropriate outputs.

i. Discuss how this technique handles multicollinearity.

```r
# read csv file
raw_df <- read.csv('../data/Pisa2009.csv')

str(raw_df)
```

```
## 'data.frame':    3404 obs. of  25 variables:
##  $ X                    : int  2 4 5 8 10 12 14 15 16 17 ...
##  $ grade                : int  11 10 10 10 10 10 10 10 11 9 ...
##  $ male                 : int  1 0 1 0 1 0 0 0 1 1 ...
##  $ raceeth              : chr  "White" "Black" "Hispanic" "White" ...
##  $ preschool            : int  0 1 1 1 1 1 1 1 1 1 ...
##  $ expectBachelors      : int  0 1 0 1 1 1 1 0 1 1 ...
##  $ motherHS             : int  1 0 1 1 1 1 1 0 1 1 ...
##  $ motherBachelors      : int  1 0 0 0 1 0 0 0 0 1 ...
##  $ motherWork           : int  1 1 1 0 1 1 1 0 0 1 ...
##  $ fatherHS             : int  1 1 1 1 0 1 1 0 1 1 ...
##  $ fatherBachelors      : int  0 0 0 0 0 0 1 0 1 1 ...
##  $ fatherWork           : int  1 1 0 1 1 0 1 1 1 1 ...
##  $ selfBornUS           : int  1 1 1 1 1 0 1 0 1 1 ...
##  $ motherBornUS         : int  1 1 1 1 1 0 1 0 1 1 ...
##  $ fatherBornUS         : int  1 1 0 1 1 0 1 0 1 1 ...
##  $ englishAtHome        : int  1 1 1 1 1 0 1 0 1 1 ...
##  $ computerForSchoolwork: int  1 1 1 1 1 0 1 1 1 1 ...
##  $ read30MinsADay       : int  1 1 1 1 0 1 1 1 0 0 ...
##  $ minutesPerWeekEnglish: int  450 200 250 300 294 232 225 270 275 225 ...
##  $ studentsInEnglish    : int  25 23 35 30 24 14 20 25 30 15 ...
##  $ schoolHasLibrary     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ publicSchool         : int  1 1 1 1 1 1 1 1 1 0 ...
##  $ urban                : int  0 1 1 0 0 0 0 1 1 1 ...
##  $ schoolSize           : int  1173 2640 1095 1913 899 1733 149 1400 1988 915 ...
##  $ readingScore         : num  575 458 614 439 466 ...
```

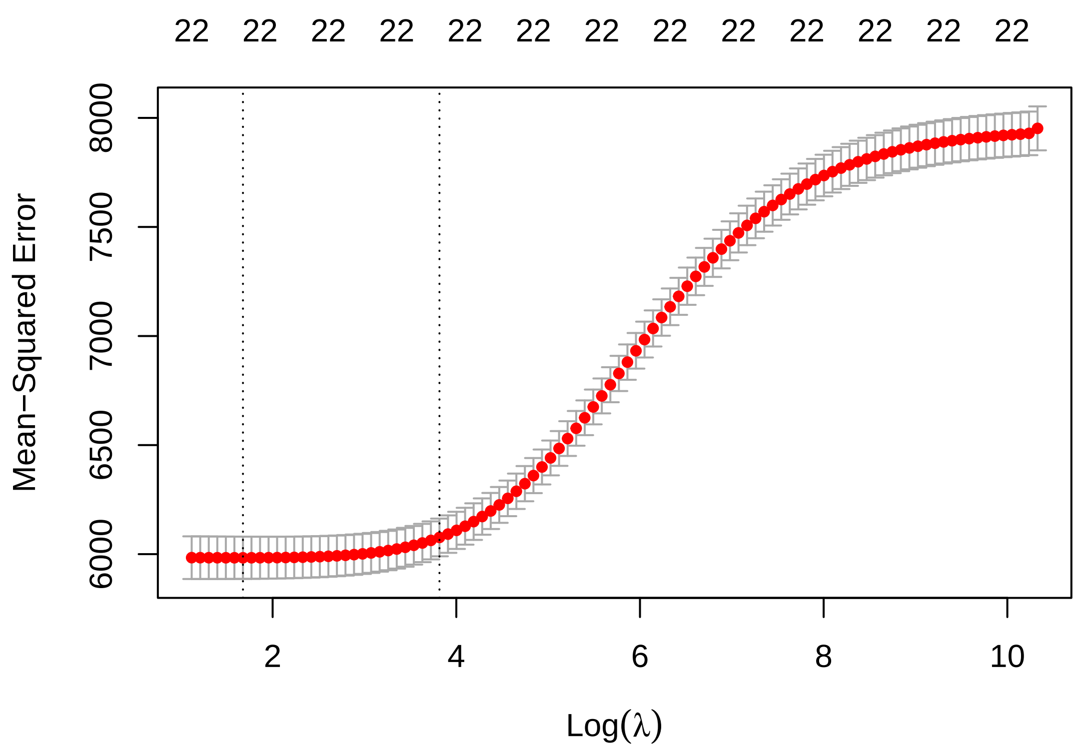
```r
# correlation
corr_df <- cor(raw_df %>% select_if(is.numeric))

# correlation plot
corrplot(corr_df)
```



Ridge regression is a technique used to address the problem of multicollinearity in linear regression models. Ridge regression introduces a penalty term, controlled by the hyperparameter lambda, which helps reduce the impact of multicollinearity. By adding the penalty term, ridge regression shrinks the coefficient estimates towards zero, making them more stable and less sensitive to minor changes in the data. Therefore, ridge regression tends to exhibit stability when considering minor changes in the data used to build the regression.

```r
# data preprocessing
df <- raw_df %>%
  mutate(raceeth = as.factor(raceeth))
```

```r
# predictors
X <- df %>%
  select(-c(X, raceeth, readingScore)) %>%
  data.matrix()

# target
y <- df$readingScore

set.seed(42)

ridge_model <- cv.glmnet(X, y, family='gaussian', alpha=0)

plot(ridge_model)
```




```r
coef(ridge_model, s = ridge_model$lambda.min)
```

```
## 23 x 1 sparse Matrix of class "dgCMatrix"
##                                 s1
## (Intercept)            153.118605567
## grade                   26.378995390
## male                   -12.084388012
## preschool               -1.701141140
## expectBachelors         51.420522656
## motherHS                 3.675078705
```

```
## motherBachelors        12.069607441
## motherWork             -3.195539742
## fatherHS               12.224295298
## fatherBachelors        22.802289022
## fatherWork              8.420586488
## selfBornUS             -0.238029976
## motherBornUS            0.044573521
## fatherBornUS            6.250526433
## englishAtHome          11.532658337
## computerForSchoolwork  25.979627894
## read30MinsADay         31.415017807
## minutesPerWeekEnglish   0.015011896
## studentsInEnglish       0.013651886
## schoolHasLibrary       -3.008442600
## publicSchool          -24.388352147
## urban                  -9.318370773
## schoolSize              0.006092463
```

```
ridge_model$lambda.min
```

```
## [1] 5.348822
```

ii. Evaluate the residual plots. Present the appropriate plots, describe them, and draw appropriate conclusions. Note: to look at the residual plots you can - after selecting variables with ridge regression - build a model using lm and plot the model.

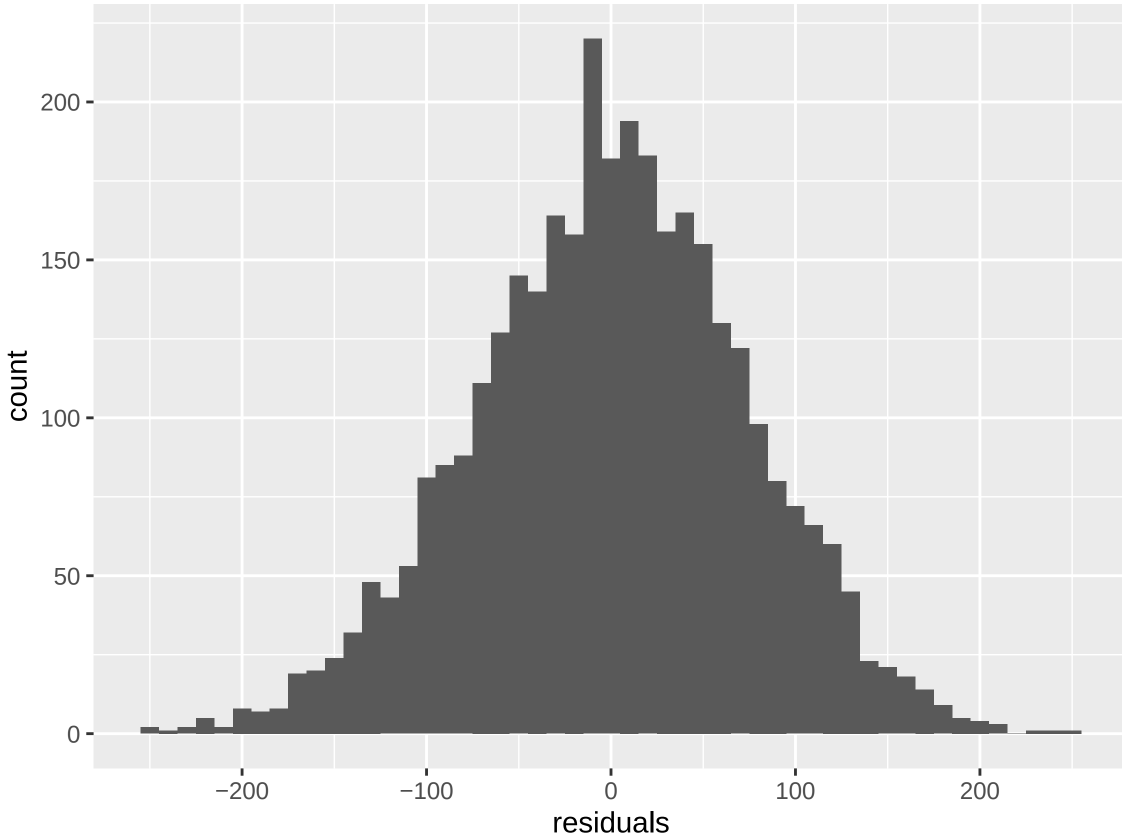
```
# let's build a base model
base_model <- lm(readingScore~ grade+ male +raceeth +expectBachelors
+motherBachelors + fatherHS +fatherBachelors+fatherWork
+motherBornUS +englishAtHome +computerForSchoolwork+read30MinsADay
+minutesPerWeekEnglish +studentsInEnglish  +publicSchool +schoolSize,
data=df)

summary(base_model)
```

```
##
## Call:
## lm(formula = readingScore ~ grade + male + raceeth + expectBachelors +
##     motherBachelors + fatherHS + fatherBachelors + fatherWork +
##     motherBornUS + englishAtHome + computerForSchoolwork + read30MinsADay +
##     minutesPerWeekEnglish + studentsInEnglish + publicSchool +
##     schoolSize, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -247.90  -48.82    0.67   49.48  250.31
##
## Coefficients:
##                                                   Estimate Std. Error t value
## (Intercept)                                     118.586382  29.642958   4.000
## grade                                            26.295997   2.503608  10.503
```

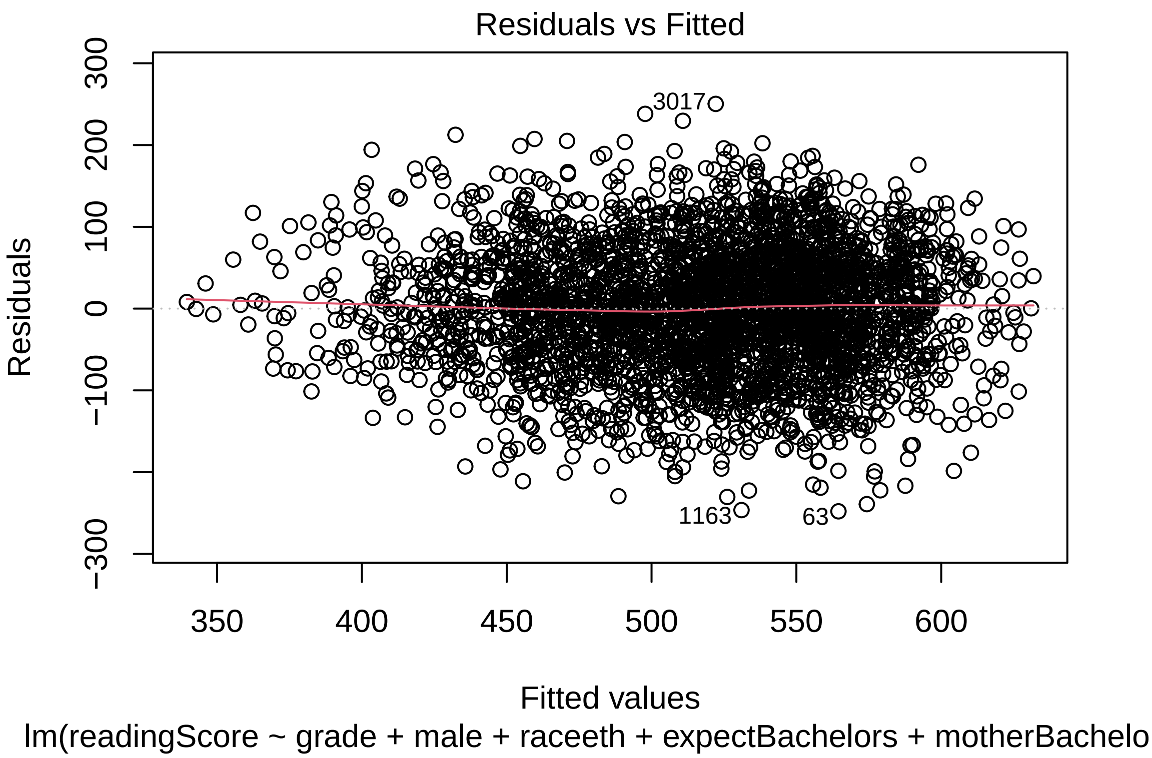
```
## male                                           -12.793127   2.646502  -4.834
## raceethAsian                                    55.510233  15.372982   3.611
## raceethBlack                                    -6.418878  14.111845  -0.455
## raceethHispanic                                 24.845643  13.949071   1.781
## raceethMore than one race                       39.813281  15.123633   2.633
## raceethNative Hawaiian/Other Pacific Islander  50.315869  20.103543   2.503
## raceethWhite                                    61.145535  13.577941   4.503
## expectBachelors                                 53.945503   3.572727  15.099
## motherBachelors                                 11.076522   3.252425   3.406
## fatherHS                                         9.303779   4.326129   2.151
## fatherBachelors                                 17.670920   3.383832   5.222
## fatherWork                                       3.416238   3.690234   0.926
## motherBornUS                                    -5.297113   4.889573  -1.083
## englishAtHome                                   11.419663   5.477132   2.085
## computerForSchoolwork                           21.002870   4.830714   4.348
## read30MinsADay                                  33.038086   2.863029  11.540
## minutesPerWeekEnglish                            0.012654   0.009037   1.400
## studentsInEnglish                               -0.161870   0.191749  -0.844
## publicSchool                                   -17.269190   5.000816  -3.453
## schoolSize                                       0.007152   0.001689   4.234
##                                                           Pr(>|t|)
## (Intercept)                                            0.000064565 ***
## grade                                         < 0.0000000000000002 ***
## male                                                   0.000001398 ***
## raceethAsian                                              0.000310 ***
## raceethBlack                                              0.649241
## raceethHispanic                                           0.074975 .
## raceethMore than one race                                 0.008514 **
## raceethNative Hawaiian/Other Pacific Islander             0.012367 *
## raceethWhite                                           0.000006917 ***
## expectBachelors                               < 0.0000000000000002 ***
## motherBachelors                                           0.000668 ***
## fatherHS                                                  0.031578 *
## fatherBachelors                                        0.000000188 ***
## fatherWork                                                0.354641
## motherBornUS                                              0.278731
## englishAtHome                                             0.037147 *
## computerForSchoolwork                                  0.000014158 ***
## read30MinsADay                                < 0.0000000000000002 ***
## minutesPerWeekEnglish                                     0.161535
## studentsInEnglish                                         0.398630
## publicSchool                                              0.000561 ***
## schoolSize                                             0.000023560 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 74.34 on 3382 degrees of freedom
## Multiple R-squared:  0.3091, Adjusted R-squared:  0.3048
## F-statistic: 72.06 on 21 and 3382 DF,  p-value: < 0.00000000000000022
```

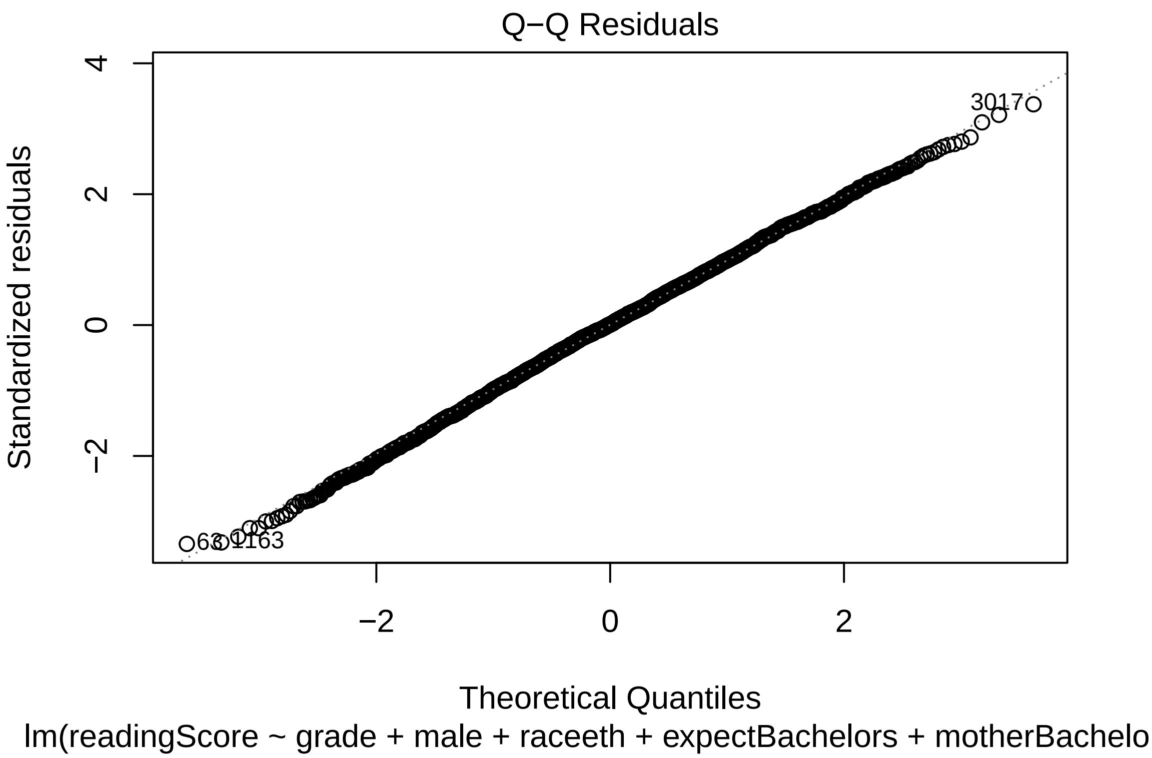
```r
residuals <- base_model$residuals


ggplot() +
```

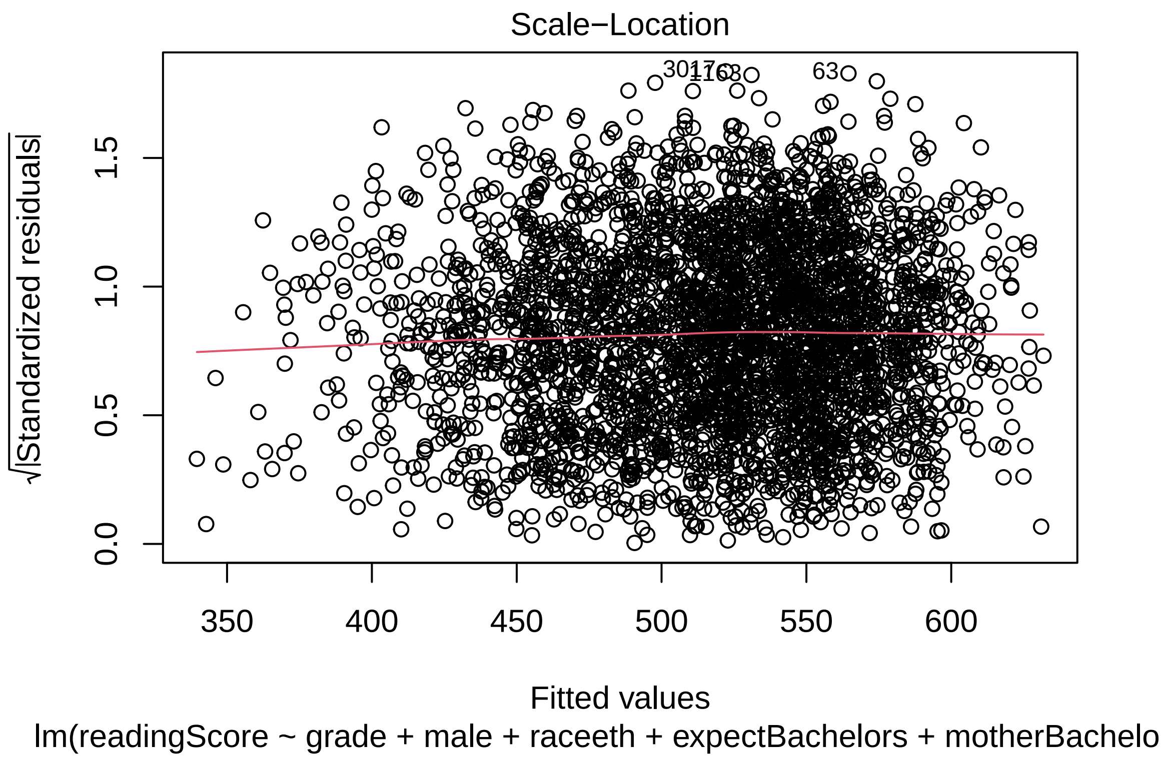
```r
  aes(residuals) +
  geom_histogram(binwidth=10)
```

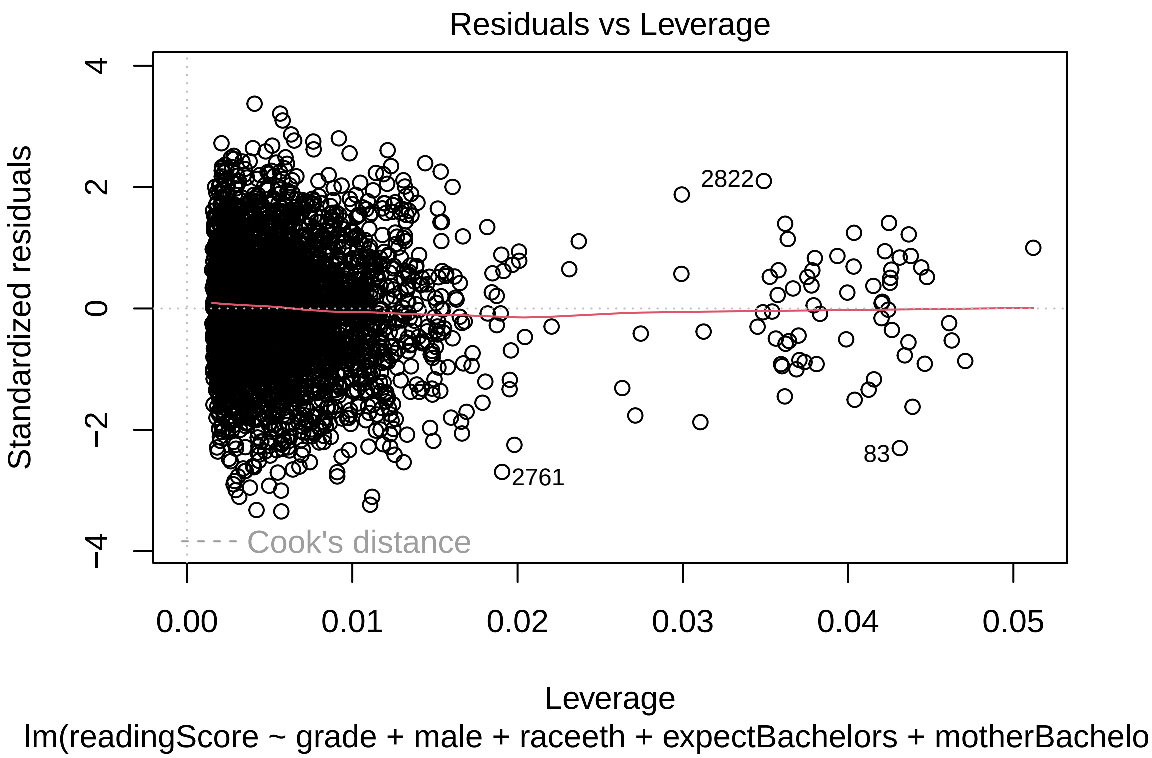



```r
plot(base_model)
```


Residuals vs Fitted
3017
1163
63
Residuals
300
200
100
0
−100
−300
350
400
450
500
550
600
Fitted values
lm(readingScore ~ grade + male + raceeth + expectBachelors + motherBachelo


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(readingScore ~ grade + male + raceeth + expectBachelors + motherBachelo
3017
63
1163
4
2
0
−2
−2
0
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
350
400
450
500
550
600
3017
1168
63
Fitted values
lm(readingScore ~ grade + male + raceeth + expectBachelors + motherBachelo



Plot 1: No non-linear pattern or heteroscedasticity in the residuals.

Plot 2: Residuals are normally distributed except for some outliers.

Plot 3: The points are randomly scattered around a horizontal line without a pattern.

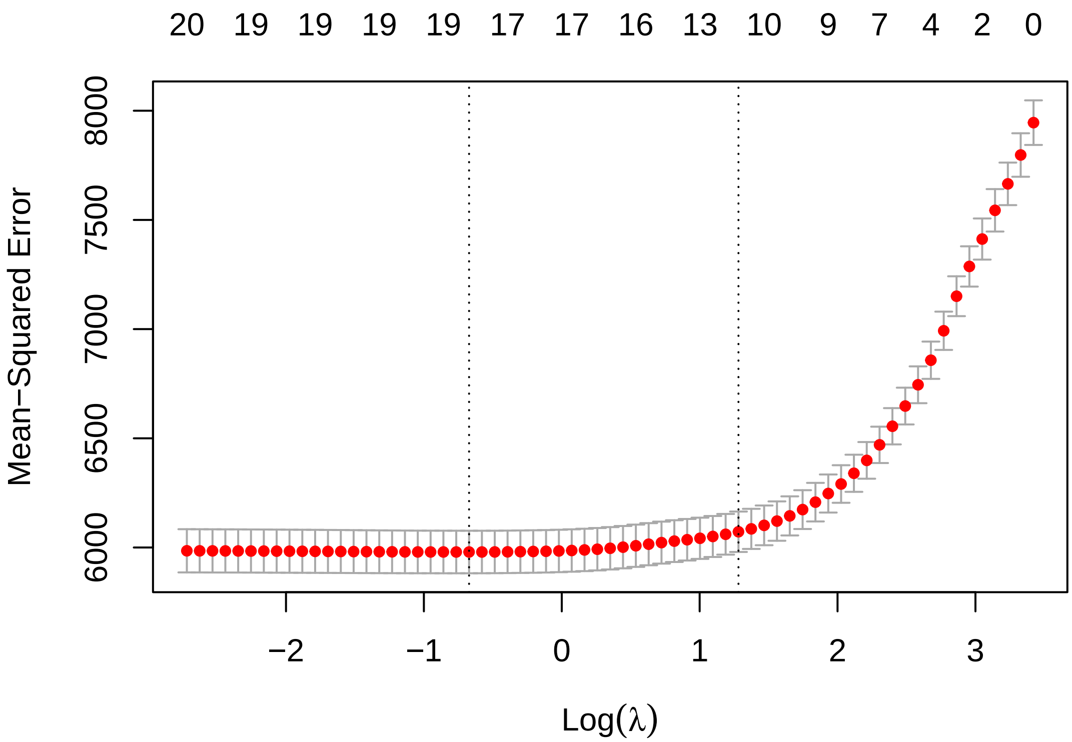
Plot 4: Few observations that have a slight impact on the model's estimates.

b. (10 points) Use LASSO regression and present your model along with appropriate outputs.

i. LASSO is a form of feature selection. Discuss how it reduced the feature space.

LASSO is a powerful technique for feature selection as it can automatically identify and eliminate irrelevant or redundant features from the model. By applying an appropriate regularization parameter, LASSO reduces the feature space by setting the coefficients of irrelevant features to zero, resulting in a more interpretable and potentially more robust model.

```
set.seed(42)

# lasso reg
lasso_model <- cv.glmnet(X,y, family='gaussian', alpha=1)

plot(lasso_model)
```




```r
lasso_model$lambda.min
```

```
## [1] 0.510571
```

```r
coef(lasso_model, s = lasso_model$lambda.min)
```

```
## 23 x 1 sparse Matrix of class "dgCMatrix"
##                                s1
## (Intercept)           143.940901025
## grade                  26.955544796
## male                  -11.378493776
## preschool              -0.497440711
## expectBachelors        53.201212886
## motherHS                2.175011896
## motherBachelors        11.351657118
## motherWork             -2.014794585
## fatherHS               11.784088706
## fatherBachelors        23.582681468
## fatherWork              7.225398562
## selfBornUS              .
## motherBornUS            .
## fatherBornUS            5.711018417
## englishAtHome          11.320729677
## computerForSchoolwork  25.612477579
```

```
## read30MinsADay         32.188583878
## minutesPerWeekEnglish   0.012272085
## studentsInEnglish       .
## schoolHasLibrary       -0.015860505
## publicSchool          -22.390992163
## urban                  -8.274420430
## schoolSize              0.005359265
```

In LASSO, a penalty term proportional to the sum of the absolute values of the coefficients is added to the linear function. The regularization parameter, typically denoted as lambda, controls the amount of shrinkage applied. The main idea behind LASSO is that by increasing the value of lambda, many coefficient estimates can be effectively set to zero, effectively eliminating their corresponding features from the model.

This process leads to sparse solutions where only a subset of the original features are retained, and the coefficients of the remaining features are non-zero. The selection of the features occurs automatically during the optimization process based on the strength of their associations with the response variable.

c. (10 points) Are the two models the same? Explain.

Those are two different models because all explanatory variables remain in the model, ridge regression has the drawback of requiring a separate approach for locating a parsimonious model.

LASSO typically performs better when p is big and few of the predicted betas are practically different from 0, as many of them may actually be equal to 0. Ridge regression typically performs better when the betas do not differ significantly in substantive magnitude. Ridge regression and the lasso will not always prevail over one another. While failing to do feature selection may not affect prediction accuracy, it can make it difficult to comprehend models in situations where there are a lot of variables (p). LASSO produces sparse models, or models that just use a portion of the variables. These models are typically considerably simpler to understand.

2. REMISSION

a. (10 points) Download "remission" and create a logistic model to predict remission.

i. Present your model.

```r
# read csv
raw_remission <- read.csv("../data/remission.csv")

summary(raw_remission)
```

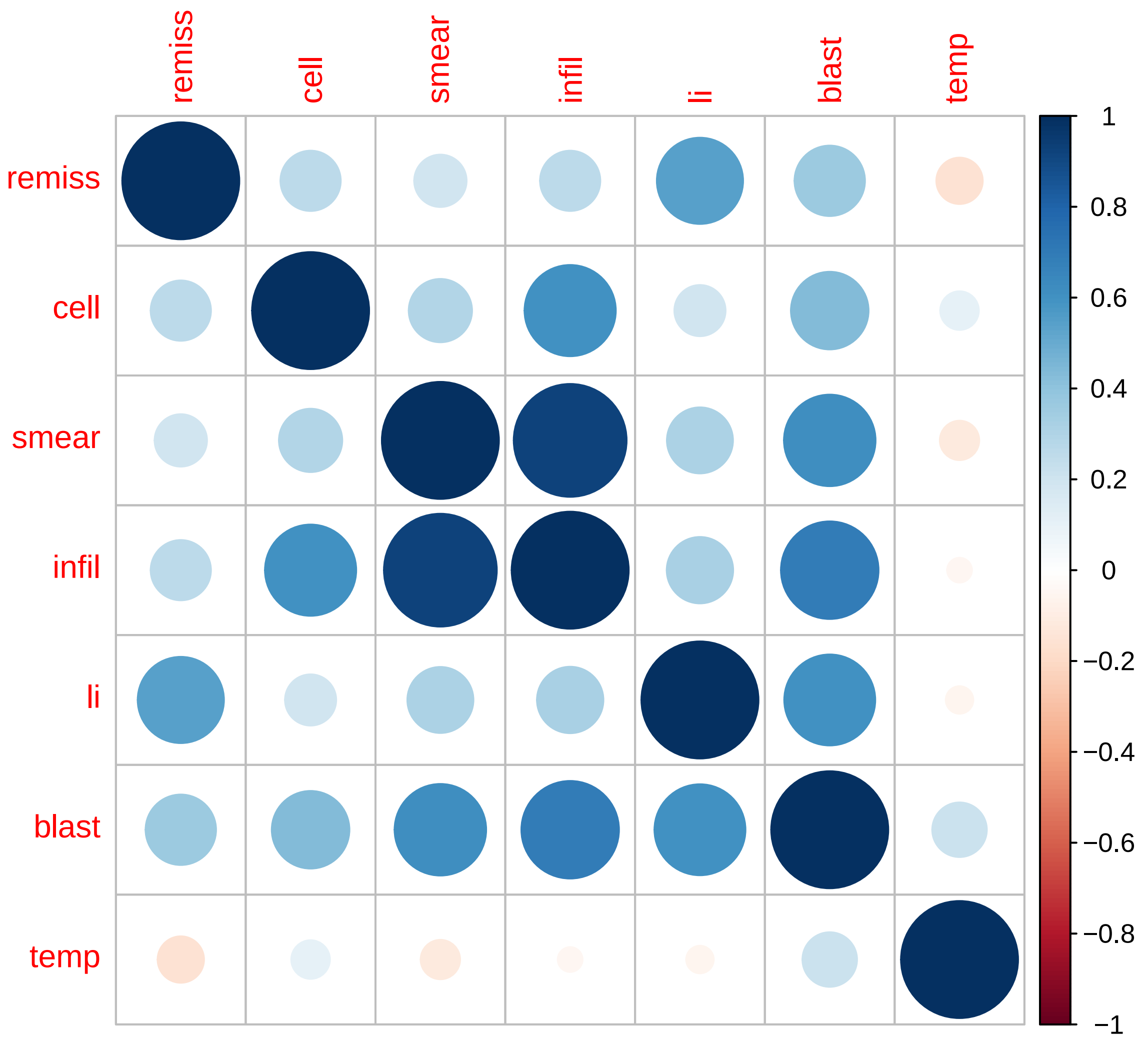
```
##      remiss            cell            smear            infil
##  Min.   :0.0000   Min.   :0.2000   Min.   :0.3200   Min.   :0.0800
##  1st Qu.:0.0000   1st Qu.:0.8250   1st Qu.:0.4300   1st Qu.:0.3350
##  Median :0.0000   Median :0.9500   Median :0.6500   Median :0.6300
##  Mean   :0.3333   Mean   :0.8815   Mean   :0.6352   Mean   :0.5707
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.8350   3rd Qu.:0.7400
##  Max.   :1.0000   Max.   :1.0000   Max.   :0.9700   Max.   :0.9200
##        li            blast             temp
##  Min.   :0.400   Min.   :0.0000   Min.   :0.980
```

```
##  1st Qu.:0.650   1st Qu.:0.2275   1st Qu.:0.986
##  Median :0.900   Median :0.5190   Median :0.990
##  Mean   :1.004   Mean   :0.6889   Mean   :0.997
##  3rd Qu.:1.250   3rd Qu.:1.0625   3rd Qu.:1.005
##  Max.   :1.900   Max.   :2.0640   Max.   :1.038
```

```
corr_df <- cor(raw_remission %>% select(is.numeric))
```

```
## Warning: Use of bare predicate functions was deprecated in tidyselect 1.1.0.
## i Please use wrap predicates in `where()` instead.
##   # Was:
##   data %>% select(is.numeric)
##
##   # Now:
##   data %>% select(where(is.numeric))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
corrplot(corr_df)
```




```r
# data preprocessing
df <- raw_remission %>%
  mutate(remiss = as.factor(remiss))


glm_model <- glm(remiss~., data= df, family= binomial)

summary(glm_model)
```

```
##
## Call:
## glm(formula = remiss ~ ., family = binomial, data = df)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   58.0385    71.2364   0.815   0.4152
```

```
## cell         24.6615    47.8377   0.516   0.6062
## smear        19.2936    57.9500   0.333   0.7392
## infil       -19.6013    61.6815  -0.318   0.7507
## li            3.8960     2.3371   1.667   0.0955 .
## blast         0.1511     2.2786   0.066   0.9471
## temp        -87.4339    67.5735  -1.294   0.1957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 34.372  on 26  degrees of freedom
## Residual deviance: 21.751  on 20  degrees of freedom
## AIC: 35.751
##
## Number of Fisher Scoring iterations: 8
```

```
# final model
final_model <- glm(remiss~li, data = df, family = binomial)

summary(final_model)
```

```
##
## Call:
## glm(formula = remiss ~ li, family = binomial, data = df)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.777      1.379  -2.740  0.00615 **
## li             2.897      1.187   2.441  0.01464 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 34.372  on 26  degrees of freedom
## Residual deviance: 26.073  on 25  degrees of freedom
## AIC: 30.073
##
## Number of Fisher Scoring iterations: 4
```

b. Notice that you are using the glm function.

i. Explain how this differs from lm.

GLM offers more flexibility by accommodating a wider range of response variable types and allowing for different error distributions and link functions. GLM is particularly useful when the assumptions of linearity and normality are not met, which makes it a powerful tool for various regression scenarios.

c. Evaluate the model particularly the independent variables.

```
summary(final_model)
```

```
##
## Call:
## glm(formula = remiss ~ li, family = binomial, data = df)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.777      1.379  -2.740  0.00615 **
## li             2.897      1.187   2.441  0.01464 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 34.372  on 26  degrees of freedom
## Residual deviance: 26.073  on 25  degrees of freedom
## AIC: 30.073
##
## Number of Fisher Scoring iterations: 4
```

```
confint(final_model)
```

```
## Waiting for profiling to be done...
```

```
##                  2.5 %    97.5 %
## (Intercept) -6.9951909 -1.409844
## li           0.8504641  5.693335
```

```
exp(coef(final_model))-1
```

```
## (Intercept)          li
##  -0.9771119  17.1244863
```

The Intercept is -3.777 with a standard error of 1.379. It indicates the estimated log-odds of the dependent variable when the independent variable (li) is 0.

The coefficient for li (independent variable) is 2.897 with a standard error of 1.187. It represents the estimated change in the log-odds of the dependent variable for a one-unit increase in the independent variable (li).

AIC is 30.073, dropped from 35 with full model.

The Null deviance is 34.372 with 26 degrees of freedom, and the Residual deviance is 26.073 with 25 degrees of freedom. These represent goodness-of-fit measures for the model.