assignment4_1.R

ramom

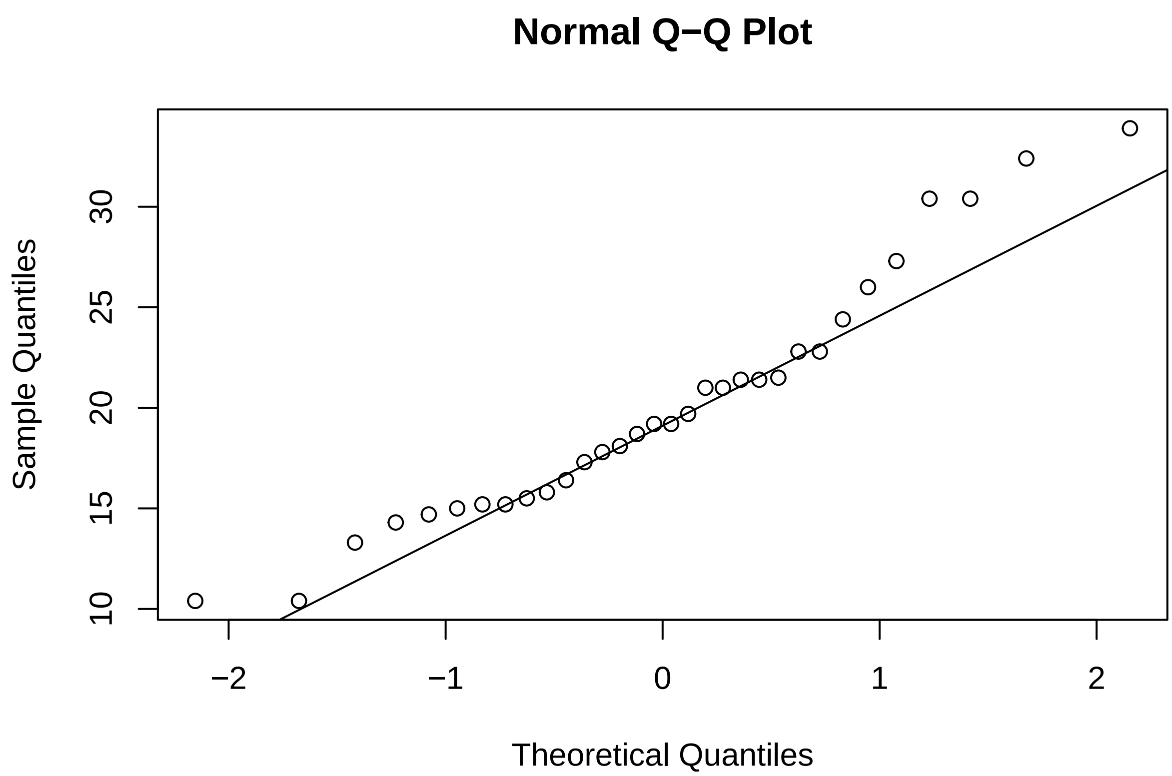
2023-06-05

```
#Use the "mtcars" data and do as follows in R Studio and submit the compiled
#PDF report file with codes and outputs here:

#Before fitting the model first load the data and  confirm that the
#dependent variable is #normally distributed

data <- mtcars  #load the data
#str(data) #check the structure of data

#check the normality of dependent variable i.e  mpg
#suggestive check
qqnorm(data$mpg)
qqline(data$mpg)
```



```r
#looking the graph we are not sure whether the data is normal or not.
#some data points align with the line and some are away
#so we do confirmation test for the normality of dependent variable
#we will use Shapiro-Wilk test

shapiro.test(data$mpg)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$mpg
## W = 0.94756, p-value = 0.1229
```

```r
#the p value(0.12) is greater than 0.05 so we can confirm that the data is
#normally distributed

#Now we can move ahead for modeling
#1. Fit multiple linear regression with mpg as dependent variable and rest
#of the variables in the mtcars data as independent variables and
#save it as mlr object
set.seed(26)
mlr <- lm(mpg ~., data = mtcars)

#2. Get the summary of mlr and interpret the result carefully
summary(mlr)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4506 -1.6044 -0.1196  1.2193  4.6271
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.30337   18.71788   0.657   0.5181
## cyl         -0.11144    1.04502  -0.107   0.9161
## disp         0.01334    0.01786   0.747   0.4635
## hp          -0.02148    0.02177  -0.987   0.3350
## drat         0.78711    1.63537   0.481   0.6353
## wt          -3.71530    1.89441  -1.961   0.0633 .
## qsec         0.82104    0.73084   1.123   0.2739
## vs           0.31776    2.10451   0.151   0.8814
## am           2.52023    2.05665   1.225   0.2340
## gear         0.65541    1.49326   0.439   0.6652
## carb        -0.19942    0.82875  -0.241   0.8122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.65 on 21 degrees of freedom
## Multiple R-squared:  0.869,  Adjusted R-squared:  0.8066
## F-statistic: 13.93 on 10 and 21 DF,  p-value: 3.793e-07
```

```r
#interpretation
#This provide the summary output of the linear regression model
#performance and significant of each predictor variables
#the estimate columns present the slope for each predictor variable. it
#represent the expected changes in response to mpg variable
#for example the coefficient for "cyl" is -0.11144, suggesting that for
#every one-unit increase in the number of cylinders, the expected change
#in mpg is a decrease of 0.11144 units.

#the last column gives the p value associated with each coefficient estimate.
#it provide the significance of each predictors contribution to the model.
#here the p value is greater than the significance level (pvalue > 0.05)
#it means that the predictor variable doesnot contribute more in the model
#expect wt which is significantly close to 0.05

#the residual or error is low

#Multiple R-squared is 0.869 means 86% of variation in mpg can be
#explained by predictor variable in the model

# The adjusted R-squared value of 0.8066 means that about 80.66% of the
#variation in mpg can be explained by the predictors, taking into
#consideration the number of predictors in the model.

#it means that the accuracy is 80% considering all the predictor variables

#3. Get the VIF of mlr model and drop variables with VIF > 10 one-by-one
#until none of the predictors have VIF > 10
library(car)
vif(mlr)
```

```
##       cyl      disp        hp      drat        wt      qsec        vs        am      gear      carb
## 15.373833 21.620241  9.832037  3.374620 15.164887  7.527958  4.965873  4.648487  5.357452  7.908747
```

```r
#dropping the variable disp as it is highest and VIF > 10
mlr1 <- lm(mpg ~ cyl+hp+drat+wt+qsec+vs+am+gear+carb, data = mtcars)
summary(mlr1)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + hp + drat + wt + qsec + vs + am + gear +
##     carb, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7863 -1.4055 -0.2635  1.2029  4.4753
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.55052   18.52585   0.677   0.5052
## cyl          0.09627    0.99715   0.097   0.9240
## hp          -0.01295    0.01834  -0.706   0.4876
## drat         0.92864    1.60794   0.578   0.5694
```

```
## wt          -2.62694    1.19800  -2.193   0.0392 *
## qsec         0.66523    0.69335   0.959   0.3478
## vs           0.16035    2.07277   0.077   0.9390
## am           2.47882    2.03513   1.218   0.2361
## gear         0.74300    1.47360   0.504   0.6191
## carb        -0.61686    0.60566  -1.018   0.3195
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.623 on 22 degrees of freedom
## Multiple R-squared:  0.8655, Adjusted R-squared:  0.8105
## F-statistic: 15.73 on 9 and 22 DF,  p-value: 1.183e-07
```

```
vif(mlr1)
```

```
##       cyl        hp      drat        wt      qsec        vs        am      gear      carb
## 14.284737  7.123361  3.329298  6.189050  6.914423  4.916053  4.645108  5.324402  4.310597
```

```
#dropping the variable cly variable  as it is highest now and VIF > 10
mlr2 <- lm(mpg ~ hp+drat+wt+qsec+vs+am+gear+carb, data = mtcars)
vif(mlr2)
```

```
##       hp     drat       wt     qsec       vs       am     gear     carb
## 6.015788 3.111501 6.051127 5.918682 4.270956 4.285815 4.690187 4.290468
```

```
#4. Fit the mlr model with predictors having VIF <=10, get the summary of mlr
#and interpret the result carefully
summary(mlr2)
```

```
##
## Call:
## lm(formula = mpg ~ hp + drat + wt + qsec + vs + am + gear + carb,
##     data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8187 -1.3903 -0.3045  1.2269  4.5183
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.80810   12.88582   1.072   0.2950
## hp          -0.01225    0.01649  -0.743   0.4650
## drat         0.88894    1.52061   0.585   0.5645
## wt          -2.60968    1.15878  -2.252   0.0342 *
## qsec         0.63983    0.62752   1.020   0.3185
## vs           0.08786    1.88992   0.046   0.9633
## am           2.42418    1.91227   1.268   0.2176
## gear         0.69390    1.35294   0.513   0.6129
## carb        -0.61286    0.59109  -1.037   0.3106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.566 on 23 degrees of freedom
## Multiple R-squared:  0.8655, Adjusted R-squared:  0.8187
## F-statistic:  18.5 on 8 and 23 DF,  p-value: 2.627e-08
```

```r
#Intrepretation
#by dropping two insignificant variable we get the summary of the model again
#and find out that wt variable is significant as the pvalue of wt is 0.03 which
#which is less than 0.05 and also we get the model accuracy as 81%

#5. Fit lasso regression with mpg as dependent variable and rest of the
#variables in the mtcars data as independent variables as cv_model object
#using cv.glmnet model included in the glmnet

# Install and load the required packages
#install.packages("glmnet")
library(glmnet)

# Separate the dependent variable (mpg) and independent variables
mpg <- mtcars$mpg
independent_vars <- mtcars[, -1] # Exclude the first column (mpg)
#print(independent_vars)
# fits a Lasso regression model and performs cross-validation
cv_model <- cv.glmnet(x = as.matrix(independent_vars), y = mpg, alpha = 1)

#cv.glmnet() function used for fitting regularized regression models,
#such as Lasso or Ridge regression, with built-in cross-validation.

#x specifies the predictor variables (independent_vars)
#as the input matrix x for the model.

#y specifies the response variable (mpg) as the input vector y for the model.

#The alpha argument controls the type of regularization used in the model.
#A value of 1 indicates Lasso regression, which applies L1 regularization to
#encourage sparsity in the coefficient estimates. L1 regularization can set
#some coefficients to exactly zero, effectively performing feature selection
#by eliminating irrelevant predictors.

print(cv_model)
```

```
##
## Call:  cv.glmnet(x = as.matrix(independent_vars), y = mpg, alpha = 1)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure    SE Nonzero
## min 0.5519    25   9.104 2.767       5
## 1se 1.5357    14  11.393 4.836       3
```

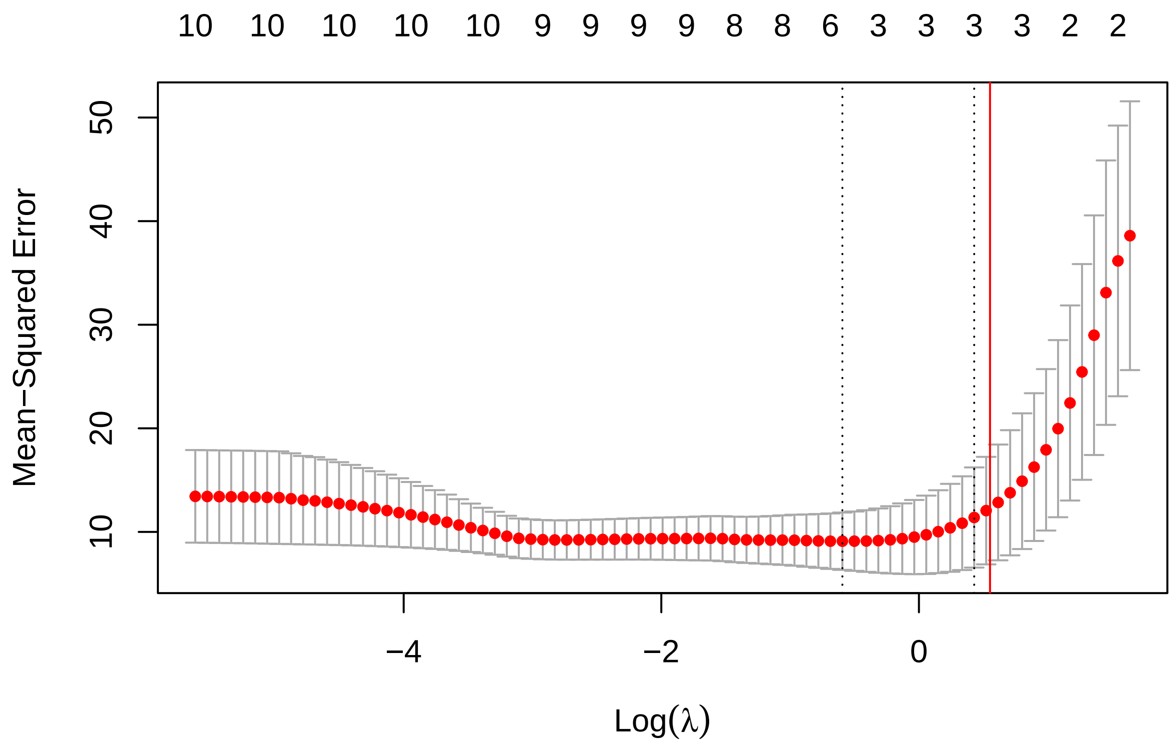
```r
#The number of non zero column is 5 when lambda is 0.55 and when the lamda
#with the standard error of mean square error(MSE), the the non zero column is 3

#6. Get the best lambda value from the lasso regression fitted above, plot
```

```
#the cv_model and interpret them carefully
# Get the best lambda value
best_lambda <- cv_model$lambda.min
#the lambda value with minimum mean squared error (MSE) during cross-validation

# Plot the cv_model
plot(cv_model)
#the plot provide cross-validated performance of the Lasso regression model
#across different lambda values.

# Add a vertical line at the best lambda value
abline(v = best_lambda, col = "red")
```




```
#7 . Fit the best lasso regression model as best_model using the
#best_lambda value obtained above
# Fit the best Lasso regression model using the best lambda value
best_model <- glmnet(x = independent_vars, y = mpg, alpha = 1,
                     lambda = best_lambda)

#8. Get the coefficients of the best_model and identify the
#important variables with s0 non-missing values
# Get the coefficients of the best model
coefficients <- coef(best_model, s = best_lambda)
print(coefficients)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) 36.30278840
## cyl         -0.87099529
## disp         .
## hp          -0.01396623
## drat         .
## wt          -2.71976202
## qsec         .
## vs           .
## am           0.34304119
## gear         .
## carb        -0.05810295
```

```
# Identify important variables with non-missing values
important_variables <- rownames(coefficients)[coefficients[, 1] != 0][-1]
print(important_variables)
```

```
## [1] "cyl"  "hp"   "wt"   "am"   "carb"
```

```
# Step 9: Fit multiple linear regression using independent variables
#from best_model
mlr_final <- lm(mpg ~ ., data = mtcars[, c("mpg", important_variables)])
#now we have fitted the multiple linear regression model using the
#independent variables obtained from Lasso Regression


#10. Compare the statistically significant variables obtained
#from step 4 and step 9
summary(mlr2)
```

```
##
## Call:
## lm(formula = mpg ~ hp + drat + wt + qsec + vs + am + gear + carb,
##     data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8187 -1.3903 -0.3045  1.2269  4.5183
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.80810   12.88582   1.072   0.2950
## hp          -0.01225    0.01649  -0.743   0.4650
## drat         0.88894    1.52061   0.585   0.5645
## wt          -2.60968    1.15878  -2.252   0.0342 *
## qsec         0.63983    0.62752   1.020   0.3185
## vs           0.08786    1.88992   0.046   0.9633
## am           2.42418    1.91227   1.268   0.2176
## gear         0.69390    1.35294   0.513   0.6129
## carb        -0.61286    0.59109  -1.037   0.3106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.566 on 23 degrees of freedom
## Multiple R-squared:  0.8655, Adjusted R-squared:  0.8187
## F-statistic:  18.5 on 8 and 23 DF,  p-value: 2.627e-08
```

```
summary(mlr_final)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = mtcars[, c("mpg", important_variables)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1890 -1.3760 -0.5532  1.5119  5.3251
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.62507    3.13296  11.371 1.37e-11 ***
## cyl         -0.81680    0.58482  -1.397    0.174
## hp          -0.01572    0.01607  -0.978    0.337
## wt          -2.36223    0.94461  -2.501    0.019 *
## am           2.07807    1.54075   1.349    0.189
## carb        -0.50441    0.46766  -1.079    0.291
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.502 on 26 degrees of freedom
## Multiple R-squared:  0.8555, Adjusted R-squared:  0.8277
## F-statistic: 30.79 on 5 and 26 DF,  p-value: 3.904e-10
```

```
#for mlr2
#Residual standard error: 2.566
#Multiple R-squared:  0.8655
#Adjusted R-squared:  0.8187 (accuracy of 82%)

#for mlr_final using Lasso regression
#Residual standard error: 2.502
#Multiple R-squared:  0.8555
#Adjusted R-squared:  0.8277 (Accuracy of 83%)

#also checking the summary based on p value
summary(mlr2)$coefficients[summary(mlr2)$coefficients[, "Pr(>|t|)"] < 0.05, ]
```

```
##    Estimate  Std. Error     t value    Pr(>|t|)
## -2.60967758  1.15878333 -2.25208415  0.03416499
```

```
#Analysis of first summary
#each unit increase in the predictor variable, the expected change in the
#response variable (mpg) is -2.60967758 units.
#here  p value (0.03416499 )is less 0.05 so we can conclude that the predictor
#variable is statistically significant in predicting the response variable.
```

```r
summary(mlr_final)$coefficients[summary
                                (mlr_final)$coefficients[, "Pr(>|t|)"] < 0.05, ]
```

```
##              Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 35.625068  3.1329609 11.37105 1.368954e-11
## wt          -2.362226  0.9446146 -2.50073 1.902602e-02
```

```r
#Analysis of final model
#the weight variable has a significant effect on mpg, with higher weights
#leading to lower mpg values.

#model using the lasso regression is best for now

#11. Write a summary for handling multicollinearity with VIF
#dropouts and LASSO regression

#Both VIF dropouts and LASSO regression are useful techniques for handling
#multicollinearity. VIF dropouts allow for a direct assessment of collinearity
#by examining the VIF values, and removing highly correlated variables based
#on a predetermined threshold. On the other hand, LASSO regression provides a
#more automated approach to variable selection, by estimating the importance
#of variables and shrinking less important ones towards zero
```