Øving 2

2022-04-21

Problem 1

```
library(ISLR)

Auto = subset(Auto, select = -name)  #Removing the name column form the dataset

str(Auto)
```

```
## 'data.frame':    392 obs. of  8 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : num  1 1 1 1 1 1 1 1 1 1 ...
```

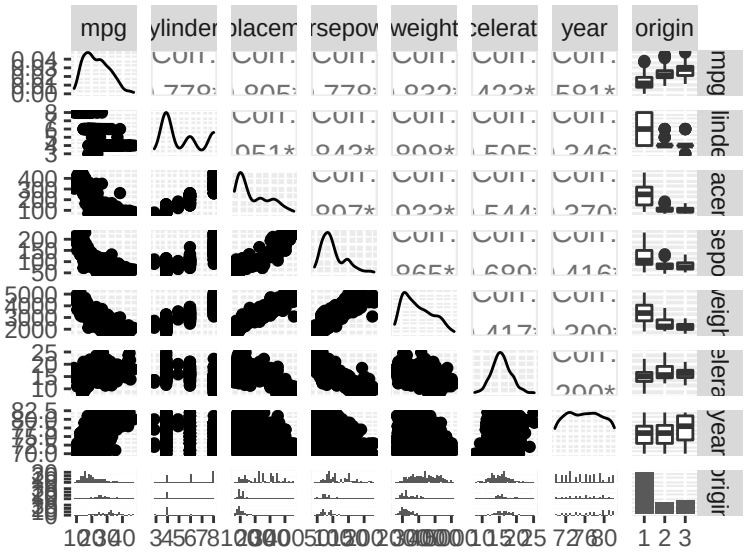
We have 392 samples of 8 variables.

Because origin is not a quantitative variable, but a qualitative encoded with 1, 2, and 3, we need to let R know that these variables are not numerical values so we do not get wrong model fits. We use the factor() function.

```
Auto$origin = factor(Auto$origin)
```

a) Use the function ggpairs() from GGally to produce a scatterplot matrix which includes all of the variables in the data set,

```
library(GGally)

ggpairs(Auto)
```



b)

Compute the correlation matrix between the variables.

```
ReducedAuto = Auto[, -8]   #Removing the 8th column (origin)

corr_matrix = cor(ReducedAuto)

corr_matrix
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
##              acceleration       year
## mpg             0.4233285  0.5805410
## cylinders      -0.5046834 -0.3456474
## displacement   -0.5438005 -0.3698552
## horsepower     -0.6891955 -0.4163615
## weight         -0.4168392 -0.3091199
## acceleration    1.0000000  0.2903161
## year            0.2903161  1.0000000
```

c) Use the lm() function to perform a multiple linear regression with mpg as the response and all other variables (except name) as the predictors.

Comment on:

i) Is there a relationship between the predictors and the response?

ii) Is there evidence that the weight of a car influences mpg? Interpret the regression coefficient β_{weight}.
iii) What does the coefficient for the year variable suggest?

```
model = lm(mpg ~ ., data = Auto)

summary(model)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0095 -2.0785 -0.0982  1.9856 13.3608
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.795e+01  4.677e+00  -3.839 0.000145 ***
## cylinders    -4.897e-01  3.212e-01  -1.524 0.128215
## displacement  2.398e-02  7.653e-03   3.133 0.001863 **
## horsepower   -1.818e-02  1.371e-02  -1.326 0.185488
## weight       -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## acceleration  7.910e-02  9.822e-02   0.805 0.421101
## year          7.770e-01  5.178e-02  15.005  < 2e-16 ***
## origin2       2.630e+00  5.664e-01   4.643 4.72e-06 ***
## origin3       2.853e+00  5.527e-01   5.162 3.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.307 on 383 degrees of freedom
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

i) We observe that the p-value is quite small (2.2e-16), meaning that the probability of observing H_0, that there is no relationship between the predictors and the response is very small, meaning that there definitely is a relationship. The R^2 is also quite high, meaning our model fits well for the data.

ii) The p-value for weight is also very small (2e-16), meaning that this variable has a huge influence on the mpg. The $\beta_{weight} = -6.710 \cdot 10^{-3}$, meaning that for one unit increase in weight, we get a β_{weight} decrease in mpg. So, a car that weights 1000 kg more than another car, would drive 6.7 miles less far per gallon of fuel.

iii) The coefficient $\beta_{year} = 7.77 \cdot 10^{-1}$ suggest that for one unit increase in year, the mpg increases by 0.777. Newer models tend to be able to drive further per gallon of fuel than older models.

iv) Look again at the regression output from question c). Now we want to test whether the origin variable is important. How does this work for a factor variable with more than only two levels?

As we have more than to types of origin, we need to look at the p-value as a whole.

```
anova(model)
```

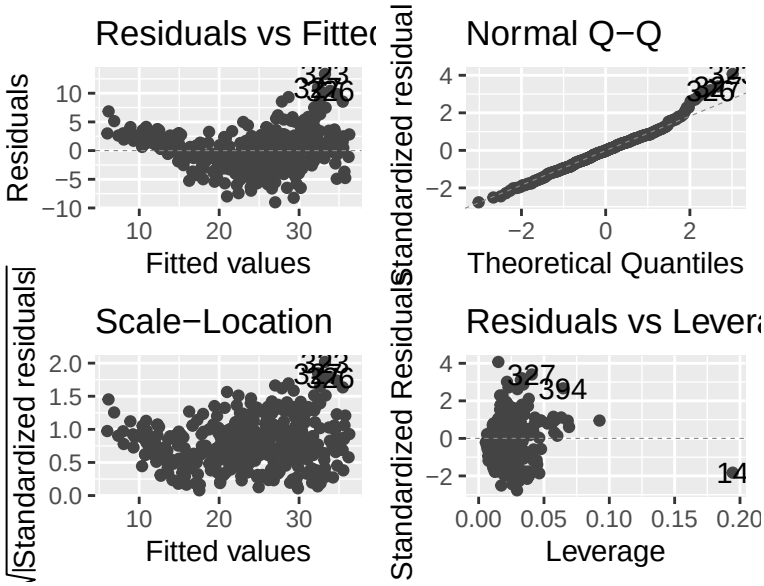
```
## Analysis of Variance Table
```

```
##
## Response: mpg
##                Df  Sum Sq Mean Sq   F value    Pr(>F)
## cylinders       1 14403.1 14403.1 1317.3788 < 2.2e-16 ***
## displacement    1  1073.3  1073.3   98.1735 < 2.2e-16 ***
## horsepower      1   403.4   403.4   36.8977 3.004e-09 ***
## weight          1   975.7   975.7   89.2447 < 2.2e-16 ***
## acceleration    1     1.0     1.0    0.0884    0.7664
## year            1  2419.1  2419.1  221.2650 < 2.2e-16 ***
## origin          2   356.0   178.0   16.2787 1.639e-07 ***
## Residuals     383  4187.4    10.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that the p-value for origin is quite small, $1.639 \cdot 10^{-7}$, which implies that the probability of all the β being zero is quite low. This means that there is a relationship between the origin and the response mpg.

e) Use the autoplot() function from the ggfortify package to produce diagnostic plots of the linear regression fit by setting smooth.colour = NA. Comment on any problems tou see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unuasuallt high leverage?

```
#install.packages("ggfortify")

library(ggfortify)
#?autoplot

autoplot(model, smooth.colour = NA)
```



Residuals vs Leverage: Here we see that observation 14 is closes to the border of Cook's distance. This observation has an unusually high leverage compared to the rest of the data, this observation should be doublechecked to be an outlier.

Outliers are data points that are far from other data points. In other words, they're unusual values in a dataset. Outliers are problematic for many statistical analyses because they can cause tests to either miss significant findings or distort real results.

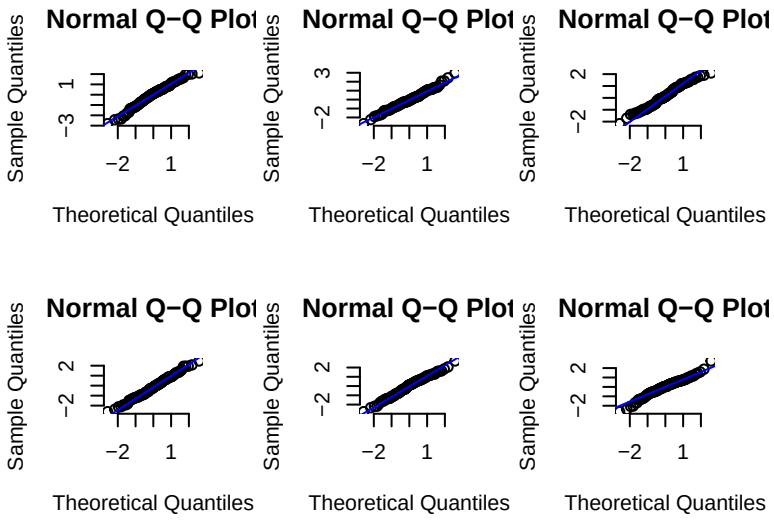
Scale-Location plot: This plot is used to check the assumption of equal variance among the residuals in our regression model. Ideally we want a horizontal line. Observation 326, 327 and 323 seems to be outliers here

Normal Q-Q plot: This plot is used to determine if the residuals of the regression model are normally distributed. If the points fall roughly along a straight diagonal line, then we can assume the residuals are normally distributed. In our plot this fits good for most of the observations. 326, 327, 323 seems to be outliers here. However, this is not enough to declare that the residuals are non-normally distributed.

Residuals vs fitted plot: This plot is used to determine if the residuals exhibit non-linear patterns. Ideally we want a horizontal line. Here we have evidence of non-linearity.

f) For beginners, it can be difficult to decide whether a QQ plot looks good or bad. A way to get a feeling of how bad a QQ plot may look, even when the normality assumptions is perfectly ok, we can use simulations: We can draw from the normal distribution and plot the QQ plot. Use the following code to repeat this six times:

```
set.seed(2332)
n = 100    #100 data points

#par() can be used to set or query graphical parameters.

#Mfrow = A vector of the form c(nr, nc). Subsequent figures will be drawn in an nr-by-nc array on the d

par(mfrow = c(2,3))



for(i in 1:6){
  sim = rnorm(n)     #Generate a random normal distribution of 100 observations
  qqnorm(sim, pch = 1, frame = FALSE)  #Produces a QQ plot
  qqline(sim, col = "blue", lwd = 1)
}
```



g) Fit another model for mpg, using only displacement, weight, year and origin as predictors, plus an interaction between year and origin (interactions can be included as year*origin, this ads the main effects and the interaction at once). Is there evidence that the interactions term is relevant? Give an interpretation of the result.

```
spec_mod = lm(mpg ~ year*origin + weight + displacement, data = Auto)

summary(spec_mod)
```

```
##
## Call:
## lm(formula = mpg ~ year * origin + weight + displacement, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7710 -2.0204 -0.0207  1.7045 13.0017
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.117e+00  5.259e+00  -0.973 0.331220
## year          6.152e-01  6.614e-02   9.302  < 2e-16 ***
## origin2      -3.735e+01  1.026e+01  -3.642 0.000307 ***
## origin3      -2.532e+01  9.441e+00  -2.682 0.007631 **
## weight       -6.685e-03  5.543e-04 -12.060  < 2e-16 ***
## displacement  4.803e-03  5.032e-03   0.955 0.340420
## year:origin2  5.187e-01  1.342e-01   3.865 0.000130 ***
## year:origin3  3.564e-01  1.213e-01   2.937 0.003514 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.257 on 384 degrees of freedom
## Multiple R-squared:  0.829,  Adjusted R-squared:  0.8259
## F-statistic: 265.9 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
anova(spec_mod)
```

```
## Analysis of Variance Table
##
## Response: mpg
##               Df Sum Sq Mean Sq  F value    Pr(>F)
## year           1 8027.7  8027.7 756.8331 < 2.2e-16 ***
## origin         2 5768.2  2884.1 271.9061 < 2.2e-16 ***
## weight         1 5712.8  5712.8 538.5876 < 2.2e-16 ***
## displacement   1   38.4    38.4   3.6245 0.0576820 .
## year:origin    2  198.9    99.5   9.3764 0.0001057 ***
## Residuals    384 4073.1    10.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As last time, we need to use the anova() (F-test) function to interpret because of origin having more than two levels.

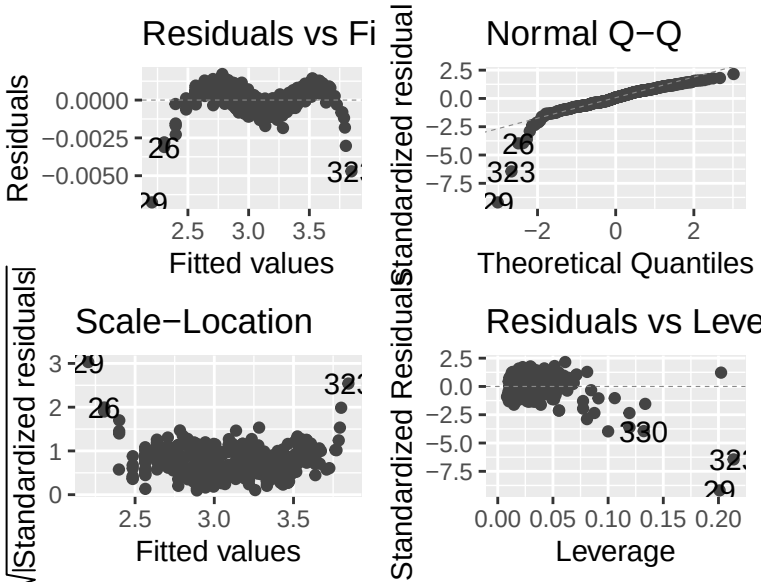
We see that the p-value for year:origin is quite low, 0.0001057, and therefore we can conclude that the year-effect depends on the origin of the car. We see that for origin2 and origin3 cars, the mpg has a steeper

slope for year, $\beta_{year} = 0.6152$ for reference value (origin1). But $\beta_{year} = 0.6152 + 0.356$ for origin 3 and $\beta_{year} = 0.6152 + 0.5187$ for origin 2.

We can conclude that cars from outside America has a bigger mpg.

h) Try a few different transformations of the variables $(\log(X), X^2, \sqrt{X})$. Perhaps you manage to improve the residual plots that you got in e)? Comment on your findings.

```
#Preprocessing data

Auto$sqrtmpg <- sqrt(Auto$mpg)
Auto$mpg2 <- (Auto$mpg)^2
Auto$logmpg <- log(Auto$mpg)


transformed_model = lm(logmpg ~ ., data = Auto)

autoplot(transformed_model, smooth.colour = NA)
```



Problem 2

a) A core finding for the least squares estimator $\hat{\beta}$ of linear regression models is

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

with $\hat{\beta} \sim N_p(\beta, \sigma^2 (X^T X)^{-1})$

i) Show that $\hat{\beta}$ has the distribution with the given mean and covariance matrix

1. $$\begin{aligned} E[\hat{\beta}] &= E[(X^TX)^{-1}X^TY] \\ &= (X^TX)^{-1}E[X^TY] \\ &= (X^TX)^{-1}E[X^T(\beta X + \epsilon)] \\ &= (X^TX)^{-1}E[X^TX\beta + X^T\epsilon)] \\ &= (X^TX)^{-1}E[X^TX\beta] \\ &= E[\beta] \\ &= \beta \end{aligned}$$

2. $$\begin{aligned} Cov[\hat{\beta}] &= Cov[(X^TX)^{-1}X^TY] \\ &= (X^TX)^{-1}X^TCov[Y]((X^TX)^{-1}X^T)^T \\ &= (X^TX)^{-1}X^T\sigma^2I((X^TX)^{-1}X^T)^T \\ &= (X^TX)^{-1}X^T\sigma^2((X^TX)^{-1}X^T)^T \\ &= \sigma^2(X^TX)^{-1}X^TX((X^TX)^{-1})^T \\ &= \sigma^2(X^TX)^{-1} \end{aligned}$$

ii) What do you need to assume to get to this result?

We need to assume that Y is multivariate normal, the same holds for $\hat{\beta}$, because it is a linear transformation of Y.

All components of a multivariate normal vector are themselves univariate normal. This means that $\hat{\beta}_j$ is normally distributed with expected valye given the $\hat{\beta}_j$ and the variance given by the jth diagonal element of $\sigma^2(X^TX)^{-1}$.

b) What is the interpretation of a 95% confidence interval?

A 95% CI for an estimator means that 95% of the time the true value of what is estimated lies in this interval.

Find the 95% CI for $Y = 1 + 3X + \epsilon \sim N(0, 1)$

```
beta0 = 1
beta1 = 3

true_beta = c(beta0,beta1)
true_sd = 1

X = runif(100,0,1)   #100 values between 0 and 1, X is predictor values

Xmat = model.matrix(~X, data = data.frame(X))  #Making X a matrix

ci_int = ci_x = 0 #Counts how many times the true value is within the CI

nsim = 1000 #Number of simulations

for(i in 1:nsim){
  y = rnorm(n=100, mean = Xmat %*% true_beta, sd = rep(true_sd, 100))
  mod = lm(y ~ x, data = data.frame(y=y, x= X))
  ci = confint(mod)
```

```
  ci_int[i] = ifelse(true_beta[1] >= ci[1,1] && true_beta[1] <= ci[1,2],1,0)
  ci_x[i] = ifelse(true_beta[2] >= ci[2,1] && true_beta[2] <= ci[2,2],1,0)
}

c(mean(ci_int), mean(ci_x))
```

```
## [1] 0.955 0.947
```

c) What is the interpretation of a 95% prediction interval? Write R code that shows the interpretation of a 95% PI.

95% of the time, the predicted value will lie in the prediction interval.

```
beta0 = 1
beta1 = 3

true_beta = c(beta0,beta1)
true_sd = 1

x0 = c(1, 0.4)

X = runif(100,0,1)   #100 values between 0 and 1, X is predictor values

Xmat = model.matrix(~X, data = data.frame(X))  #Making X a matrix

pi_y0 = 0 #Counts how many times the true value is within the CI

nsim = 1000 #Number of simulations

for(i in 1:nsim){
  y = rnorm(n=100, mean = Xmat %*% true_beta, sd = rep(true_sd, 100))
  mod = lm(y ~ x, data = data.frame(y=y, x= X))
  y0 = rnorm(n=1, mean = x0 %*% true_beta, sd = true_sd)
  pi = predict(mod, newdata = data.frame( x = x0[2]), interval = "predict")[2:3]
  pi_y0[i] = ifelse(y0 >= pi[1] && y0 <= pi[2], 1, 0)
}

mean(pi_y0)
```

```
## [1] 0.945
```

e) What is the difference between error and residual? What are the properties of the raw residuals? Why don´t we want to use the raw residuals for model check? What is our solution to this?

The error is the natural error from your data (ϵ), that can never be reduced or observed. Residual is calculated after running the regression model and is the differences between the observed values and the estimated values.The residuals is an estimation of the error. The residuals is the difference between the true response and the predicted value:

$$\hat{\epsilon} = Y - \hat{Y} = (I - X(X^T X)^{-1} X^T)Y$$

Properties of the raw residuals: Normally distributed with mean 0 and covariance $Cov(\hat{\epsilon}) = \sigma^2(I - X(X^TX)^{-1}X^T))$. This means that the residuals mat have different variance (depending on X) and may also be correlated.

We want to check if our errors are independent, homoscedastic (same variance for each observation) and not dependent on covariates. We don´t want to use the raw residuals for model check, as these are predictors and mat have different variances and may be correlated.

We standardize the residuals so they at least have equal variances.