Lab 15

Yunting Chiu

2021-04-16

Exercise 1 - Polynomial Regression. Predict US population

a) Load the data USpop.csv and plot Population as a function of Year.

- We can see the curve line in the plot below.

```
USpop <- read_csv("./data/USpop.csv")
```

```
## Parsed with column specification:
## cols(
##   Year = col_double(),
##   Population = col_double()
## )
```

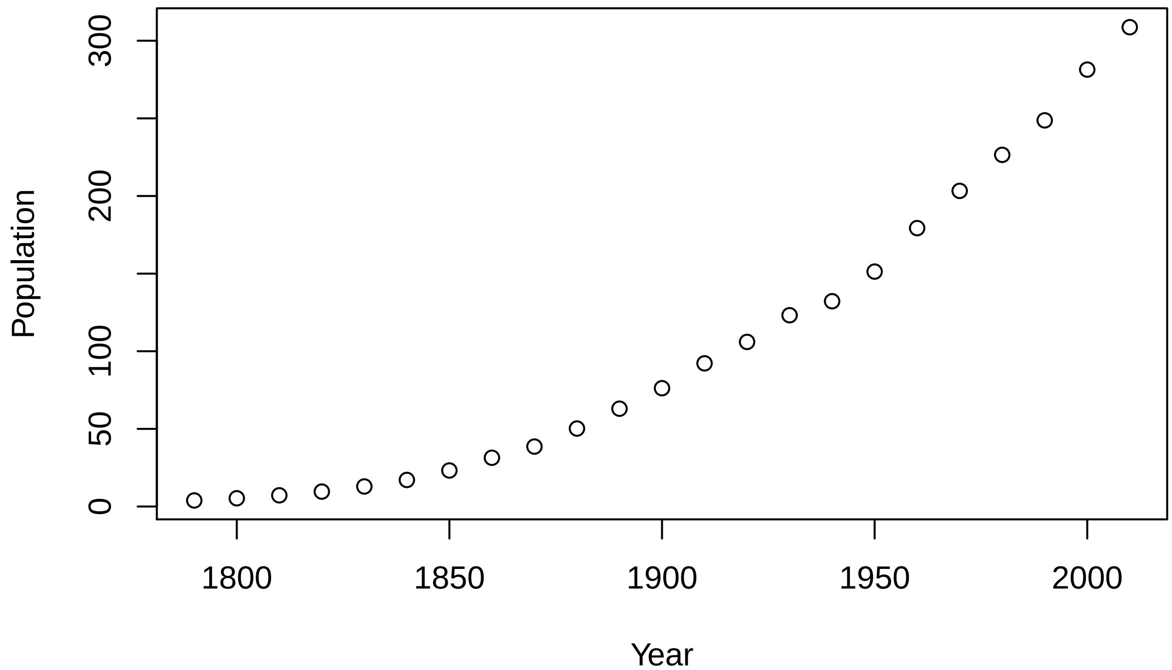
```
USpop
```

```
## # A tibble: 23 x 2
##     Year Population
##    <dbl>      <dbl>
##  1  1790        3.9
##  2  1800        5.3
##  3  1810        7.2
##  4  1820        9.6
##  5  1830       12.9
##  6  1840       17.1
##  7  1850       23.2
##  8  1860       31.4
##  9  1870       38.6
## 10  1880       50.2
## # ... with 13 more rows
```

```
names(USpop)
```

```
## [1] "Year"       "Population"
```

```
attach(USpop)
plot(Year, Population)
```



b) Use a linear model to fit the data. Does a linear model provide a good fit?

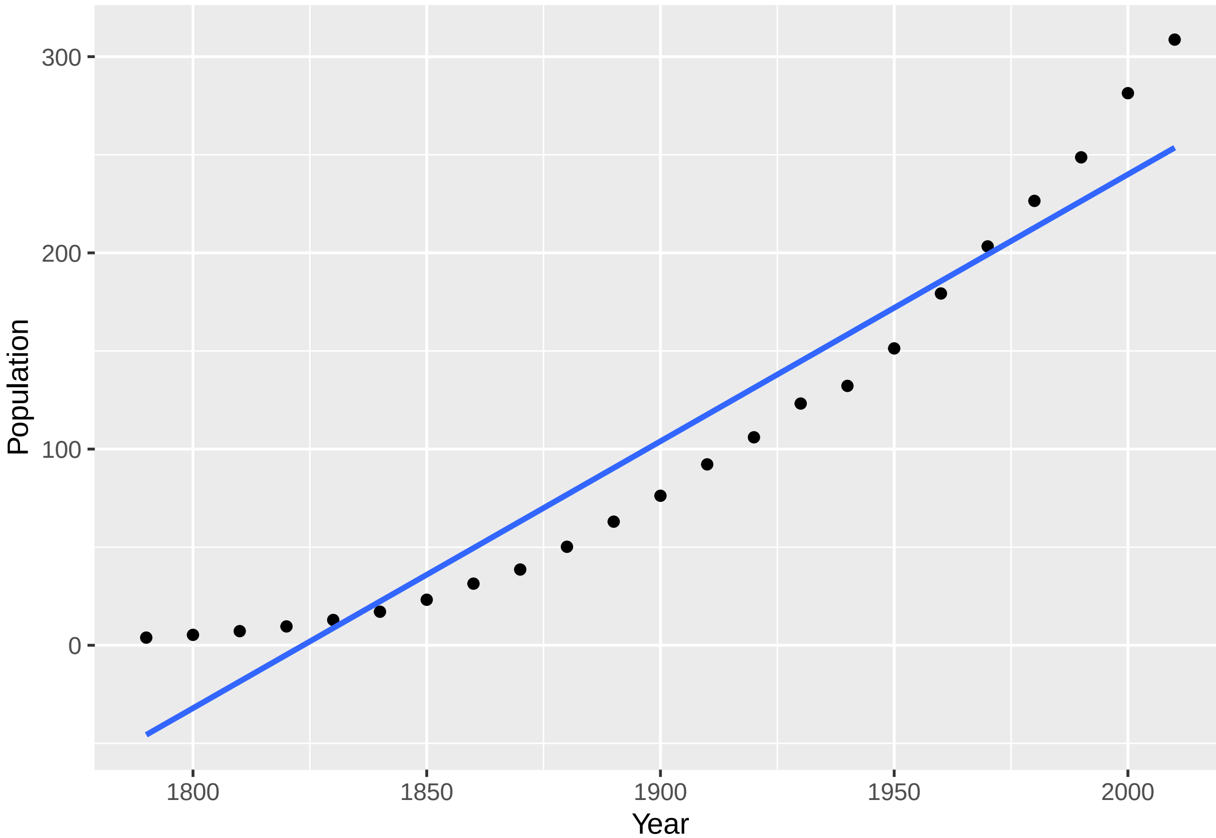
- The p-value and R squared is good, maybe the liner model is a good model. However, according to the plot, the model does not provide a good fit. We considered using the polynomial regression model when comparing the regression line and observed data.
- If n is small, the F-stat should be big in order to reject the null (the F-statistic: 239.3 is large).

```
linearModel <- lm(Population~Year)
summary(linearModel)
```

```
##
## Call:
## lm(formula = Population ~ Year)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.774 -24.872  -6.295  18.374  55.087
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.481e+03  1.672e+02  -14.84 1.33e-12 ***
## Year         1.360e+00  8.794e-02   15.47 5.93e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.97 on 21 degrees of freedom
## Multiple R-squared:  0.9193, Adjusted R-squared:  0.9155
## F-statistic: 239.3 on 1 and 21 DF,  p-value: 5.927e-13
```

```
augment(linearModel) %>%
  ggplot(aes(x = Year, y = Population)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



c) Calculate R^2. What do you observe? Does the value of R2 imply that a linear model is a good choice?

- Although R-squared 0.9193 is excellent, we must constantly compare the independent variable and dependent variable plots (overfitting). Then we'll understand why the linear model isn't the best option.

d) Using the linear model, predict the US population for the year 2030. Is this a good prediction?

- Using linear model to predict `Year` 2030 is not make sense because the `Population` in `Year` 2010 is 308.7 million, but our prediction in `Year` 2030 is only 280.8202 million.

```r
predict(linearModel, data.frame(Year = 2030))
```

```
##        1
## 280.8202
```

```r
USpop %>%
  tail(1)
```

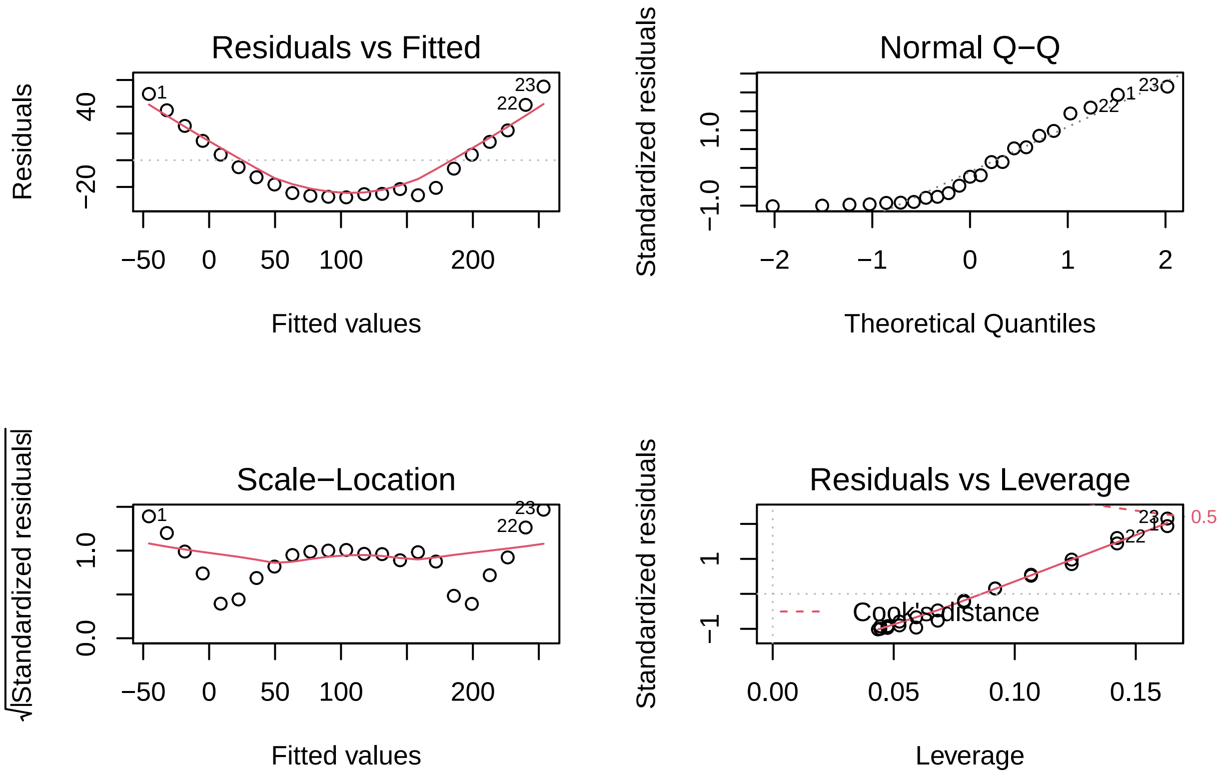
```
## # A tibble: 1 x 2
##    Year Population
##   <dbl>      <dbl>
## 1  2010       309.
```

e) Produce appropriate residual plots and decide whether or not an important predictor has been omitted. What do you observe?

- Residuals vs Fitted plot: strong curve pattern , looks non-linearity
- Normal Q-Q plot: The shape does not appear to follow a normal distribution, particularly the lower tail.

- Scale Location: We could say that the data in the middle is homoscedastic, while the data elsewhere is heteroscedastic.
- Residual vs Leverage: some potential outliers in observation 22 and 23 (high leverage).

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



f) Fit a quadratic model and plot the fitted curve. Is this a good fit?

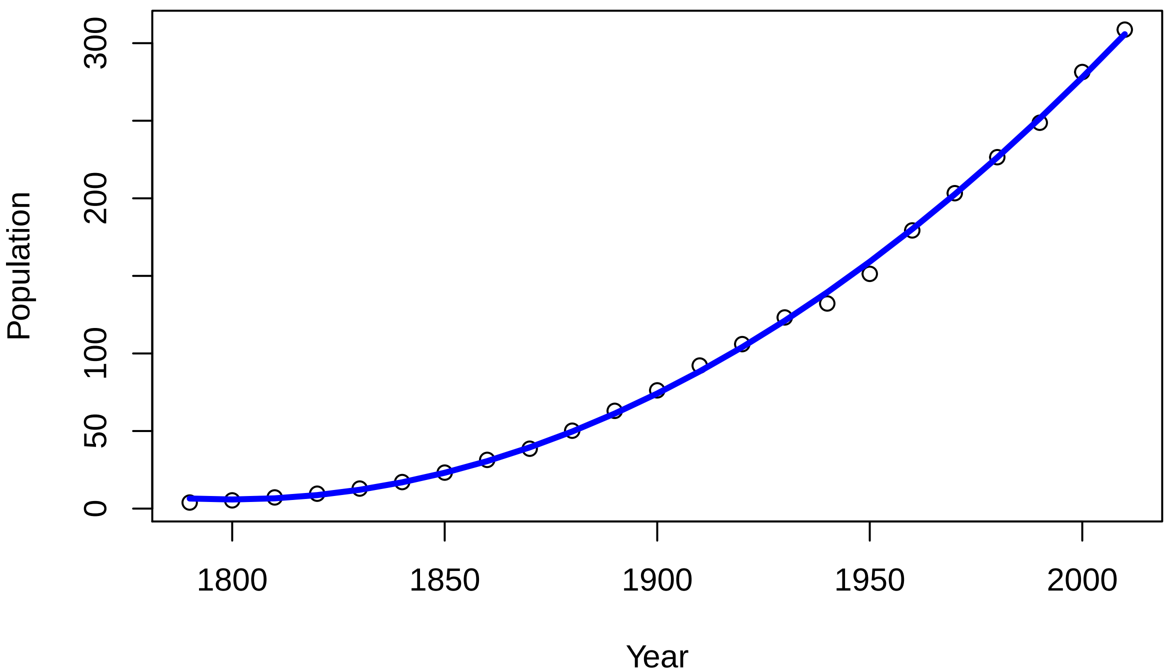
- The plot shows that the curve is well fitted. The R-squared also excellent, and each independent variable is significant.

```r
quadModel <- lm(Population ~ poly(Year, 2))
# or: quadModel2 <- lm(Population ~ Year + I(Year^2))
summary(quadModel)
```

```
##
## Call:
## lm(formula = Population ~ poly(Year, 2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8220 -0.7130  0.5961  1.8344  3.7487
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     103.9739     0.6304  164.94   <2e-16 ***
## poly(Year, 2)1 432.7557     3.0231  143.15   <2e-16 ***
## poly(Year, 2)2 127.4790     3.0231   42.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.023 on 20 degrees of freedom
## Multiple R-squared:  0.9991, Adjusted R-squared:  0.999
## F-statistic: 1.113e+04 on 2 and 20 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(1, 1))
plot(Year, Population)
Yhat = fitted.values(quadModel)
lines(Year, Yhat, col = "blue", lwd = 3)
```



g) Predict the population for the year 2030. Is this a reasonable prediction?

- Compared to linear model: 280.8202 million people, the quadratic model is reasonable.

```
predict(quadModel, data.frame(Year = 2030))
```

```
##        1
## 365.4891
```

Exercise 2 - Regression Diagnostics

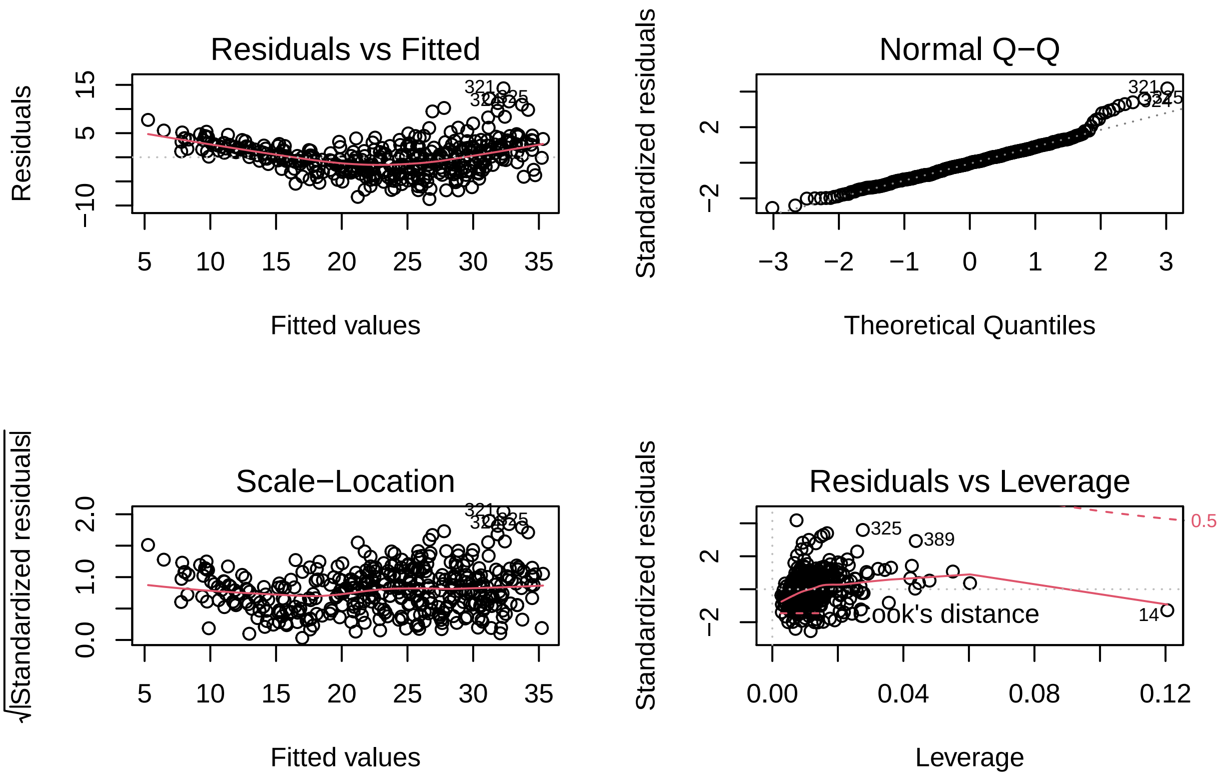
a) Load the Auto.rda dataset. Predict mpg using the predictors year, accelerator, horsepower, weight. Generate different residual plots.

- Residuals vs Fitted plot: strong curve pattern , looks non-linearity
- Normal Q-Q plot: The shape does not follow a normal distribution, especially the higher tail.
- Scale Location: The data has a slight homoscedasticity to it.
- Residual vs Leverage: we focus on some high leverage observations, they may be potential outliers.

```
load("./data/Auto.rda")
attach(Auto)
```

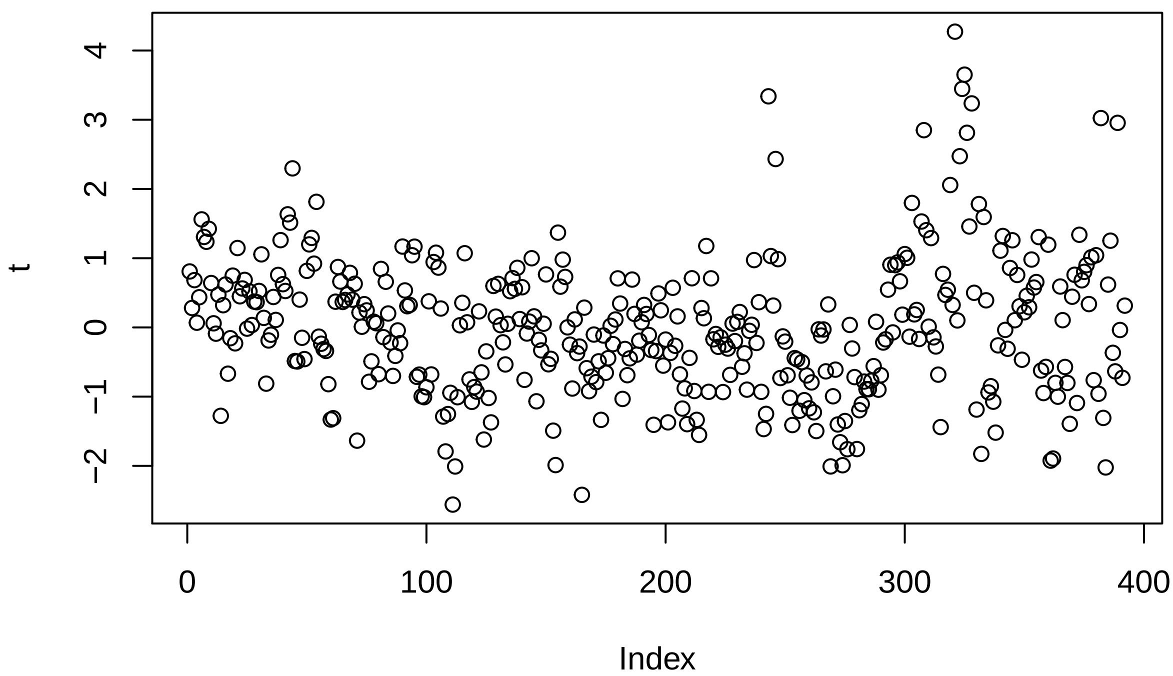
```
## The following object is masked from package:ggplot2:
##
##     mpg
```

```r
# model
reg <- lm(mpg ~ year + acceleration + horsepower + weight)
par(mfrow = c(2, 2))
plot(reg)
```



b) Which of these residuals can be considered as outliers? Compare with the Bonferroni-adjusted quantile from t-distribution.

```r
t <- rstudent(reg)
plot(t)
```



#

Look the summary first

```
summary(t)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -2.559162 -0.690663 -0.028693  0.001774  0.589435  4.272284
```

```
t[abs(t) > 3]
```

```
##      243      321      324      325      328      382
## 3.338459 4.272284 3.446234 3.651403 3.236226 3.024362
```

compute the upper quantile of the t-distribution.

```
qt(0.05/2/392, 387) # qt(alpha/ x/ , n-predictors-1)
```

```
## [1] -3.870293
```

compare the t-student residual vs critical value

- There is one outlier

```
t[abs(t) > abs(qt(0.05/2/392, 387))]
```

```
##      321
## 4.272284
```

c) Test Normality using Shapiro-Wilk normality test. Also look at the Normal Q-Q plot above. Shapiro-Wilk statistic W measures how close the graph is to a straight line.

- The small p-value indicates that there is non-normality (rejected the null).

```
shapiro.test(t)
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  t
## W = 0.97109, p-value = 5.101e-07
```

d) Test HOMOSCEDASTICITY (constant variance) using the Breausch-Pagan test.

- A significantly different variance could overshadow the differences between means and lead to incorrect conclusions.
- HOMOSCEDASTICITY is rejected, meaning that there is evidence of equal variance (Heteroskedasticity).

```r
library(car)
```

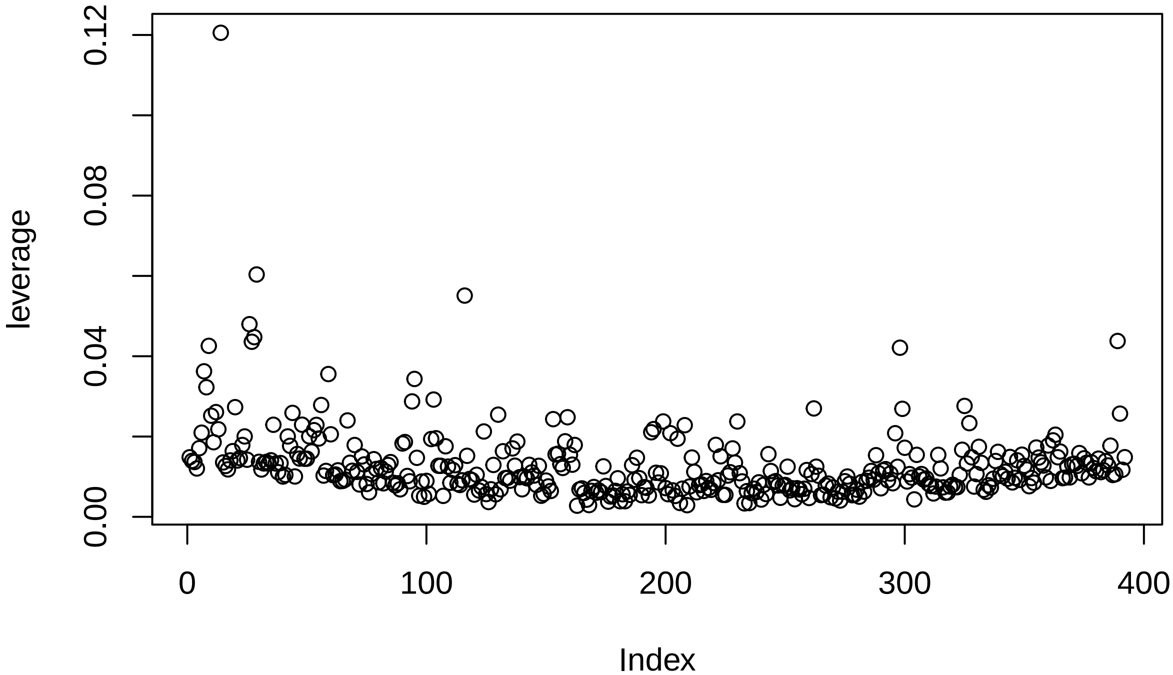
```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some
```

```r
ncvTest(reg)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 22.04621, Df = 1, p = 2.6616e-06
```

e) Check for INFLUENTIAL DATA

```r
infl <- influence(reg)
leverage <- infl$hat
plot(leverage)
```



```r
5/length(mpg) # average leverage: mean value
```

```
## [1] 0.0127551
```

```r
summary(infl$hat)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.002781 0.007543 0.010638 0.012755 0.014735 0.120544
```

```r
check_leverage <- leverage[leverage > 0.03] # 0.03 is between the third quantile and the max value
check_leverage
```

```
##          7          8          9         14         26         27         28
## 0.03624109 0.03226743 0.04258253 0.12054403 0.04797419 0.04360256 0.04475796
##         29         59         95        116        298        389
## 0.06035105 0.03555978 0.03434446 0.05510052 0.04212120 0.04379524
```