HW 2

Question 8

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
require(ISLR2)
```

```
## Loading required package: ISLR2
```

```


df <- Auto

model <- lm(mpg ~ horsepower, data = df)

summary(model)
```

Creating lm

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
```

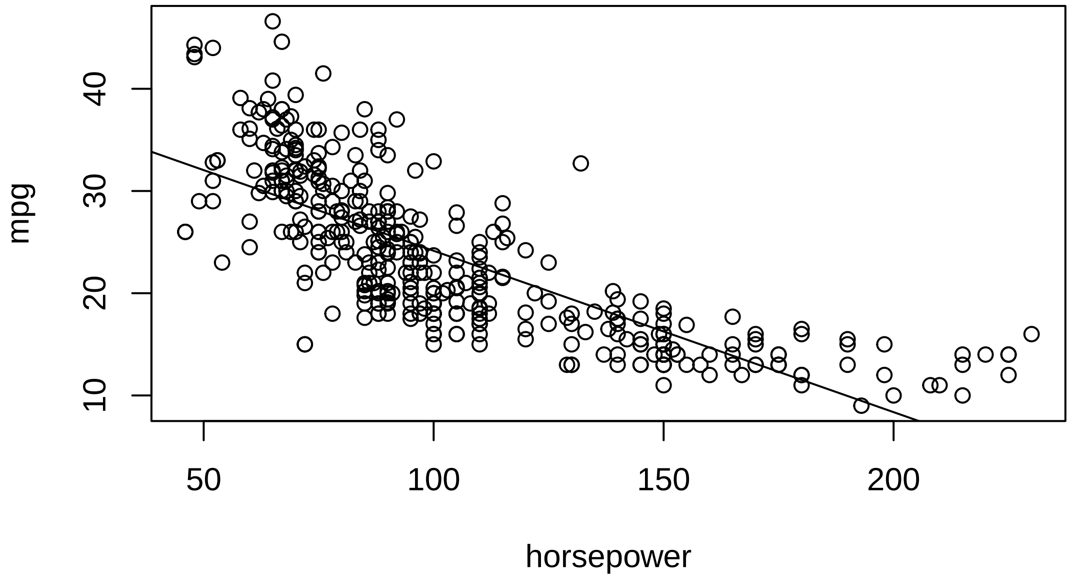
```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

There is a clear relationship between `horsepower` and `mpg` in the `Auto` dataset. The `horsepower` coefficient of `-0.16` indicates that for every one additional unit of `horsepower` corresponds with less `mpg`. The negative relationship is fairly strong, though it is interesting that it’s more of a non-linear relationship at higher `horsepower` values, where the difference between 150 units of `horsepower` and 200 units of `horsepower` is fairly small, whereas the difference between 75 units and 100 units is far more pronounced.

```r
new <- data.frame(horsepower = c(98))

pred <- predict(model, newdata = new, interval = "confidence") %>%
  data.frame()
```

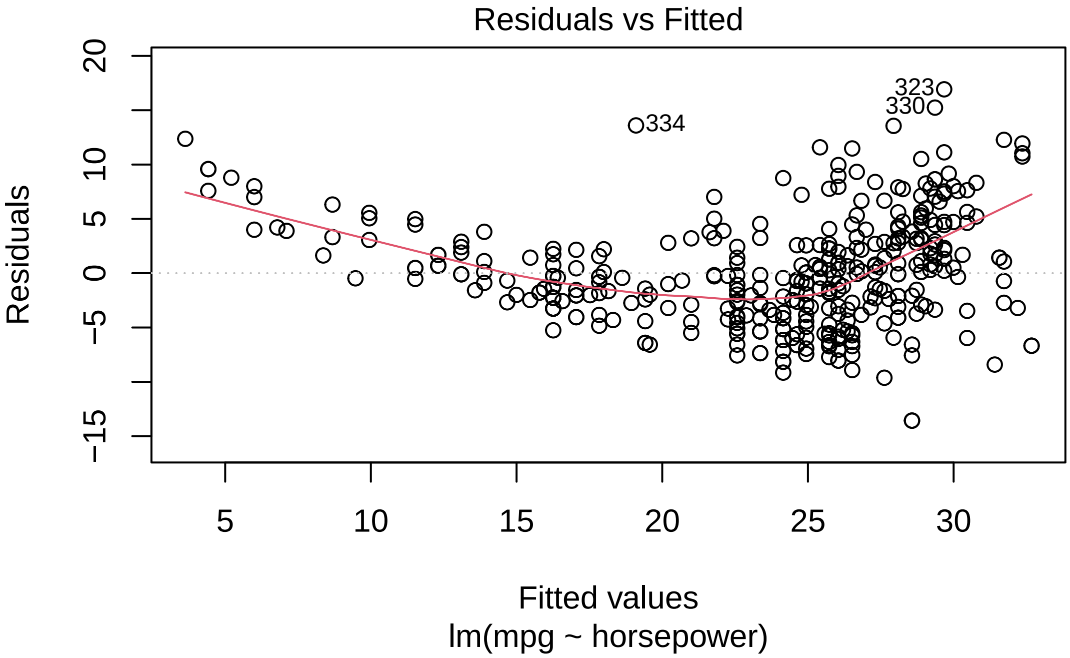
The predicted `mpg` of a 98 `horsepower` is 24.4670772, while the confidence interval ranges from 23.973079 to 24.9610753. Below is a plot of `mpg` as a function of `horsepower`, with the least squares regression line plotted over it.

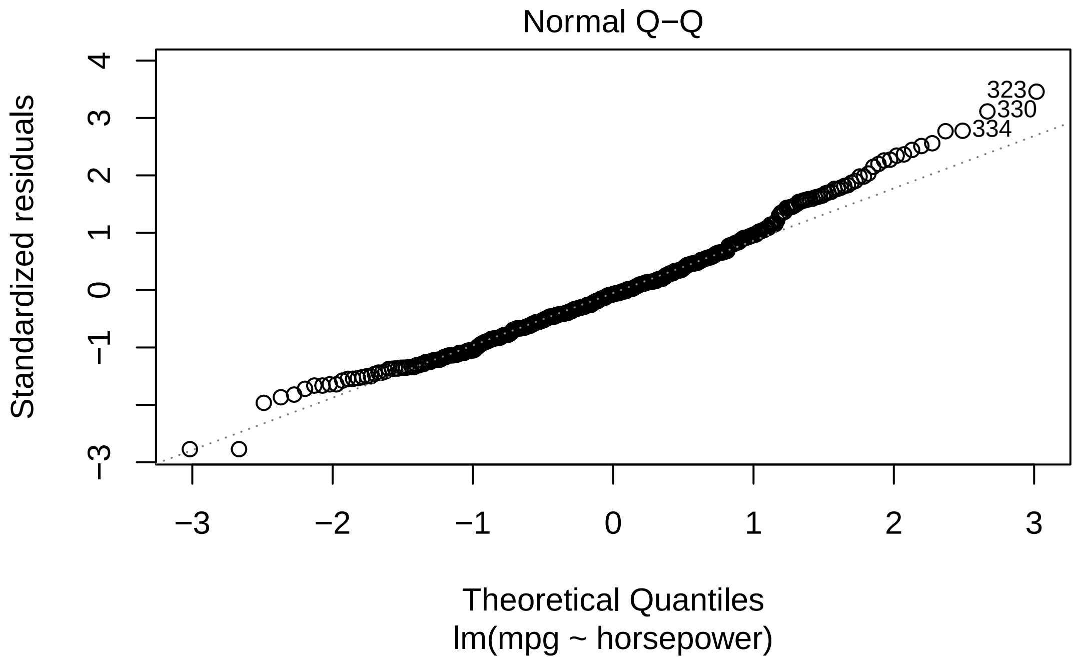
```r
# df %>%
#   ggplot(aes(x = horsepower, y = mpg)) +
#   geom_point() +
#   geom_smooth(method = "lm", formula = mpg ~ horsepower) +
#   theme_minimal() +
#   labs(title = "Horsepower vs MPG") +
#   theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold"))

with(df, plot(horsepower, mpg)) +
abline(model)
```

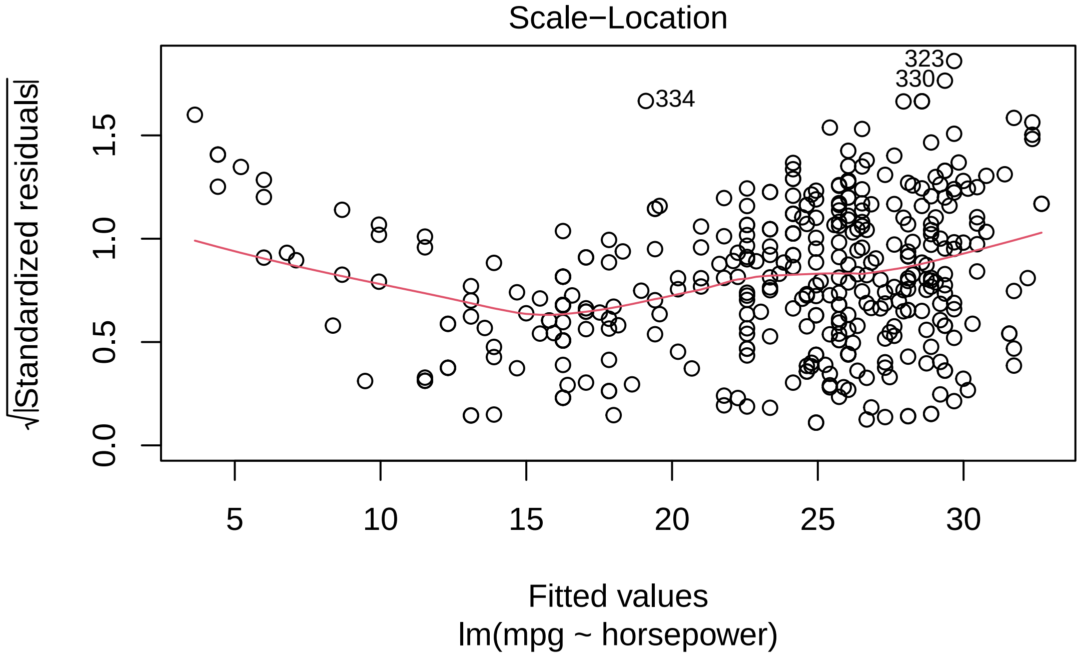


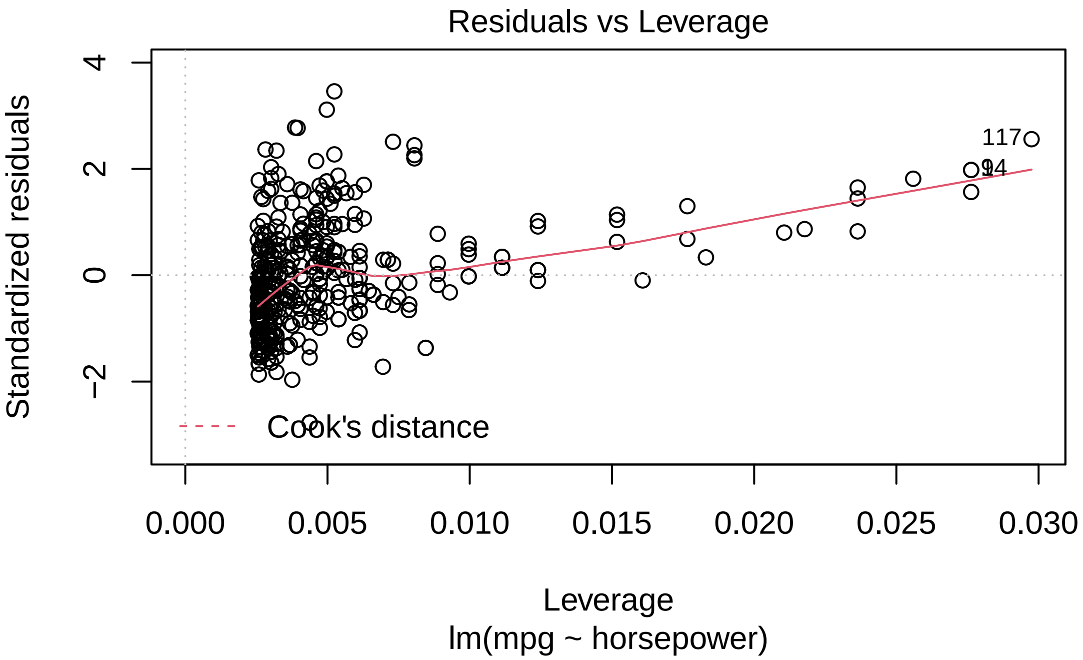
```
## integer(0)
```

```
plot(model)
```




Normal Q−Q
Standardized residuals
4
3
2
1
0
−1
−3
−3
−2
−1
0
1
2
3
323
330
334
Theoretical Quantiles
lm(mpg ~ horsepower)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
5
10
15
20
25
30
323
330
334
Fitted values
lm(mpg ~ horsepower)



From evaluating the diagnostic plots, it's clear that the linear model is not a good fit for the current data set. The residuals are not evenly or consistently distributed over the distribution of fitted values. For example, the model consistently under predicts low values of `mpg`. The large positive residual shows that the actual values are well above the prediction. A similar phenomenon occurs at higher values too; given the `horsepower vs mpg` and line of best fit plot, the non-linear relationship between the two variables ends up under-predicting values at both extremes.