Homework 2

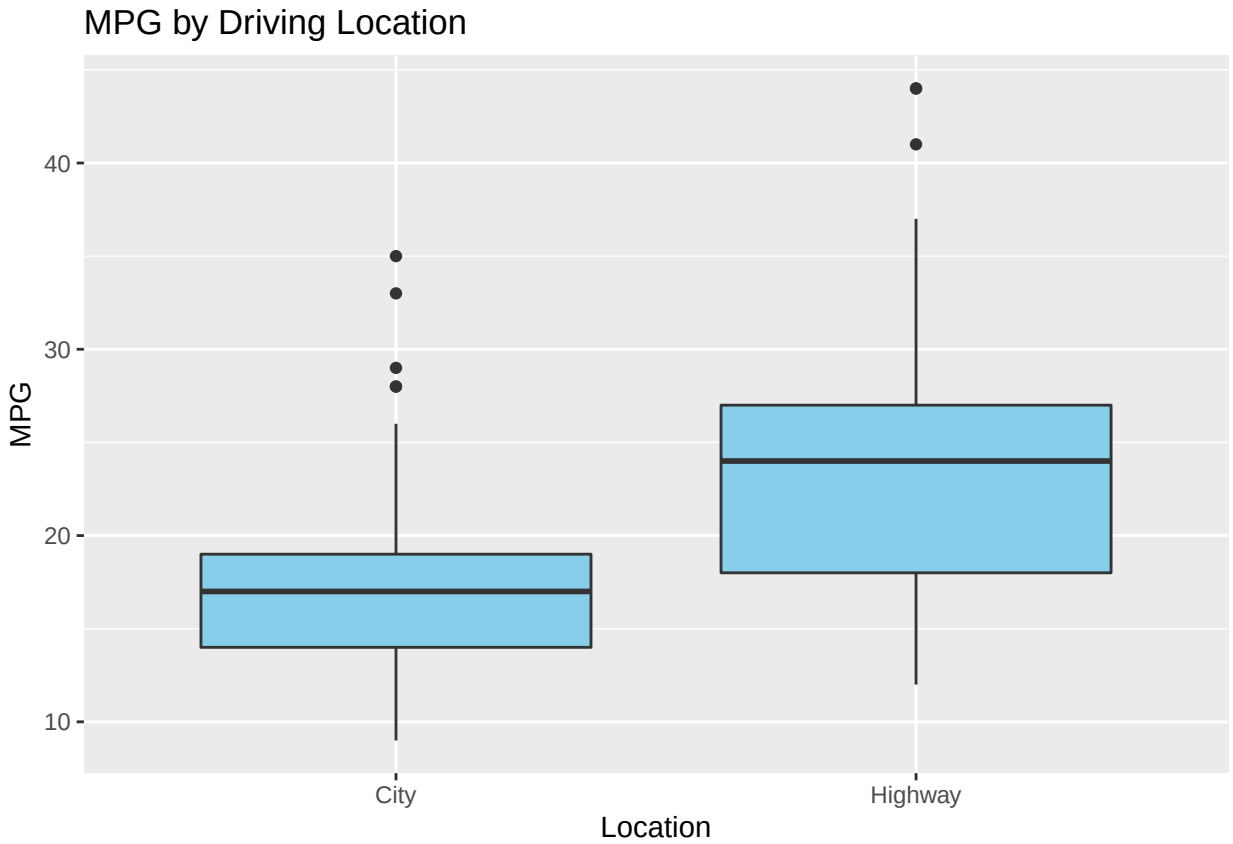
Weeks 4 and 5

```
library(tidyverse)

# Load up the mpg dataset (from the ggplot2 package)
data("mpg")
```

1. Create a plot comparing the miles per gallon by different driving location (`cty` or `hwy`). You may find this easier to do if you transform the data set using a method we learned in a previous lesson. What do you see in this plot/what should be the main takeaways?
2. Take the plot you created in 1 and make it publication ready however you see fit (scale, labels, color, theme, etc.).

```
# 1 and 2
mpg %>%
  gather(cty, hwy, key = "location", value = "mpg") %>%
  ggplot(aes(x = location, y = mpg)) +
  geom_boxplot(fill = "skyblue") +
  scale_x_discrete(labels = c("City", "Highway")) +
  labs(x = "Location", y = "MPG", title = "MPG by Driving Location")
```



3. Create another plot on your own using the `mpg` data set. Explain why you chose to create the plot that you did, why you chose the variables you did, and why you think it is an important relationship to look

at. Explain what you see in your plot.

4. Calculate each of the following and tell me what we can take away from each statistic:

- Count of `drv`
- Quartiles of `hwy`
- Mean and median of `cty`

```
table(mpg$drv)
```

```
##
##   4   f   r
## 103 106  25
```

```
quantile(mpg$hwy)
```

```
##   0%  25%  50%  75% 100%
##   12   18   24   27   44
```

```
mean(mpg$cty)
```

```
## [1] 16.85897
```

```
median(mpg$cty)
```

```
## [1] 17
```

Weeks 6 and 7

5. Take a look at the `mpg` data set. If we were to predict `hwy` using a linear regression model, what do you think would be good to use as predictors? Use any pre-analysis steps or general knowledge of the data set to support your ideas.

6. Using the `mpg` data set, build a model using `displ`, `drv`, and `class` to predict `hwy`. Explain your output from a **practical** perspective.

```
model_mpg <- lm(hwy ~ displ + drv + class, data = mpg)
summary(model_mpg)
```

```
##
## Call:
## lm(formula = hwy ~ displ + drv + class, data = mpg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5629 -1.3534 -0.1988  1.3004 13.9605
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      36.8795     1.7559  21.004  < 2e-16 ***
## displ            -2.1923     0.2375  -9.229  < 2e-16 ***
## drvf              3.1676     0.6261   5.059 8.77e-07 ***
## drvr              1.4251     0.7772   1.834 0.068030 .
## classcompact     -5.8422     1.4929  -3.913 0.000121 ***
## classmidsize     -6.1168     1.4787  -4.137 4.99e-05 ***
## classminivan    -10.2495     1.6156  -6.344 1.22e-09 ***
## classpickup     -10.3147     1.4329  -7.198 9.12e-12 ***
## classsubcompact  -5.2626     1.4492  -3.631 0.000350 ***
## classsuv         -9.2334     1.3456  -6.862 6.57e-11 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.609 on 224 degrees of freedom
## Multiple R-squared:  0.8155, Adjusted R-squared:  0.8081
## F-statistic:   110 on 9 and 224 DF,  p-value: < 2.2e-16
```

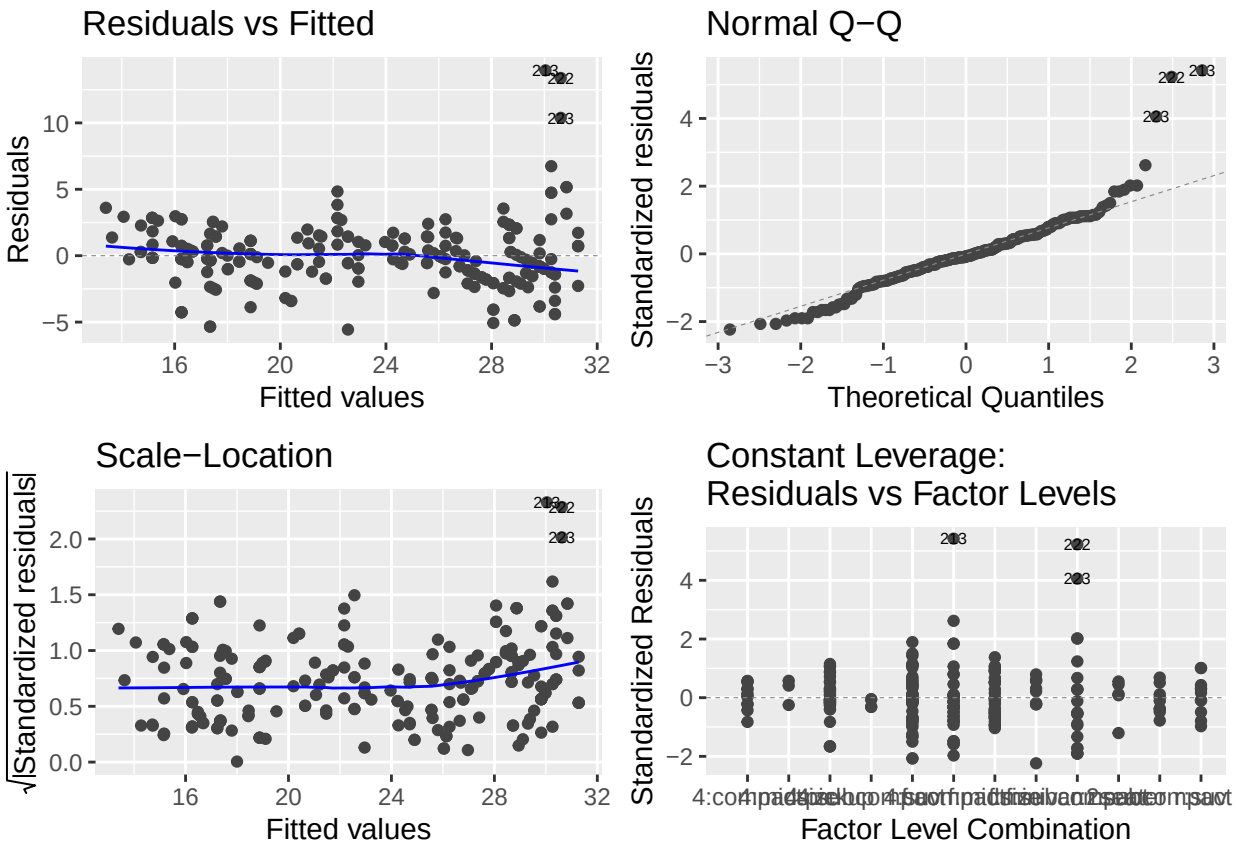
7. Check regression assumptions; explain why each assumption is met or not.

```
library(car)
library(ggfortify)

vif(model_mpg)
```

```
##           GVIF Df GVIF^(1/(2*Df))
## displ 3.224461  1        1.795679
## drv   5.708799  2        1.545739
## class 6.863335  6        1.174117
```

```
autoplot(model_mpg, label.size = 2)
```




```
# Load up the geyser data set in the MASS package
library(MASS)
data("geyser")
```

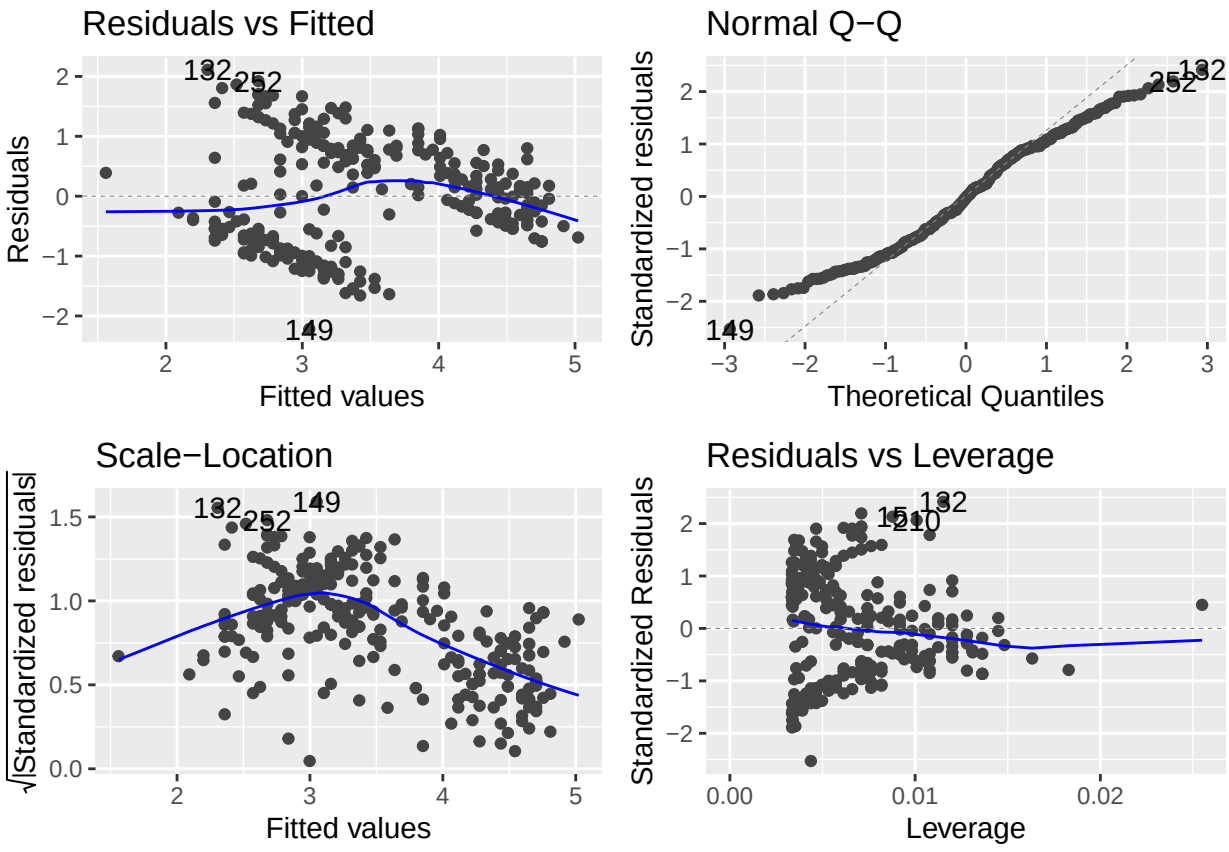
8. Perform some pre-analysis on the geyser data set; explain what you see.

9. Build a simple linear regression model predicting `duration` by `waiting`. Explain output and assumptions.

```r
model_geyser1 <- lm(duration ~ waiting, data = geyser)
summary(model_geyser1)
```

```
##
## Call:
## lm(formula = duration ~ waiting, data = geyser)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.21805 -0.72357 -0.01979  0.75071  2.11109
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.313144   0.269935   27.09   <2e-16 ***
## waiting     -0.053272   0.003666  -14.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.879 on 297 degrees of freedom
## Multiple R-squared:  0.4155, Adjusted R-squared:  0.4136
## F-statistic: 211.2 on 1 and 297 DF,  p-value: < 2.2e-16
```

```r
autoplot(model_geyser1)
```

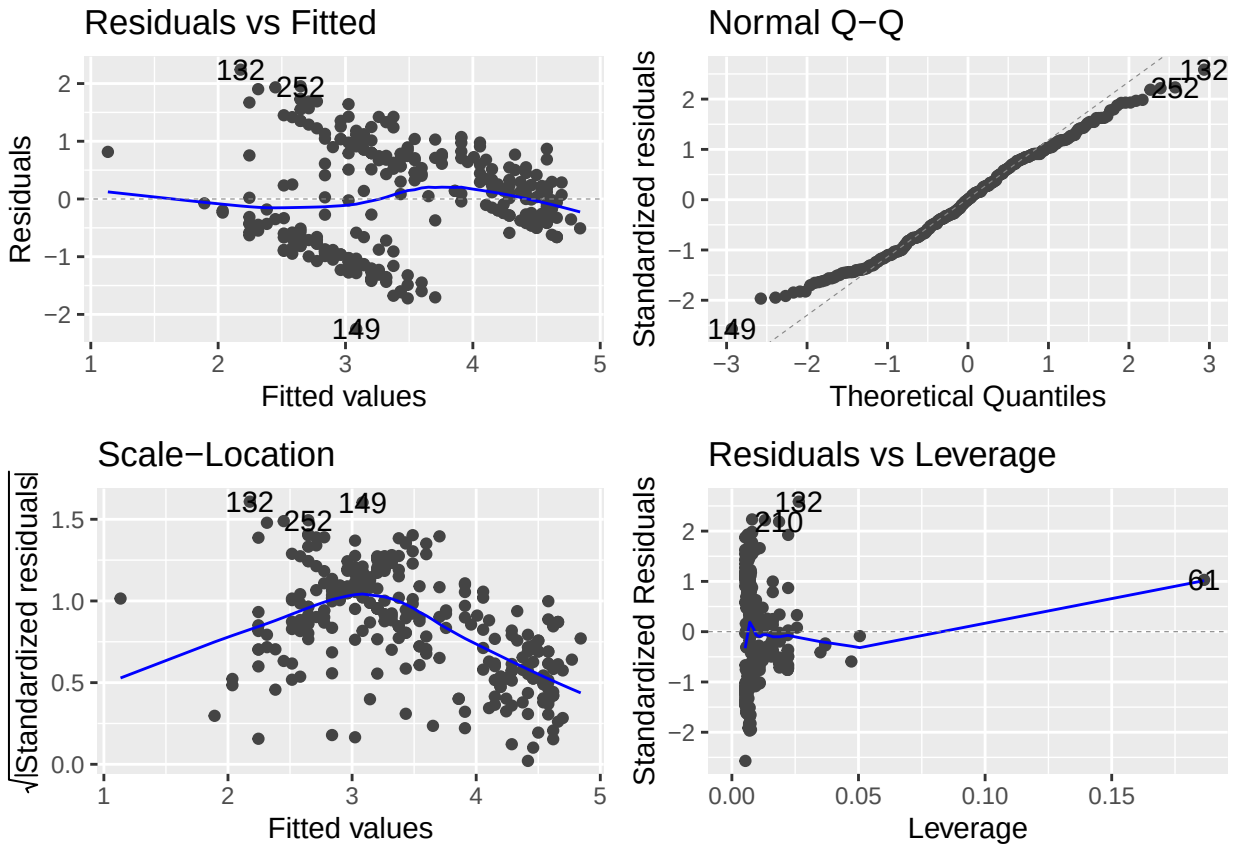


10. Build a polynomial model predicting the same thing as 9. You choose the degree. Explain your choices and output. Compare this model with the one you built in 9. Which do you think is better?

```r
model_geyser2 <- lm(duration ~ poly(waiting, 2), data = geyser)
summary(model_geyser2)
```

```
##
## Call:
## lm(formula = duration ~ poly(waiting, 2), data = geyser)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.25113 -0.66002 -0.00038  0.70681  2.24057
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.4608     0.0508  68.130   <2e-16 ***
## poly(waiting, 2)1 -12.7738     0.8784 -14.543   <2e-16 ***
## poly(waiting, 2)2  -1.0607     0.8784  -1.208    0.228
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8784 on 296 degrees of freedom
## Multiple R-squared:  0.4184, Adjusted R-squared:  0.4145
## F-statistic: 106.5 on 2 and 296 DF,  p-value: < 2.2e-16
```

```r
autoplot(model_geyser2)
```




```r
anova(model_geyser1, model_geyser2)
```

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

```
##
## Model 1: duration ~ waiting
## Model 2: duration ~ poly(waiting, 2)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    297 229.50
## 2    296 228.38  1    1.1252 1.4583 0.2282
```