Name - NetID

Homework 10

Homework Instructions

For questions that require code, please create a code chunk directly below the question and type your code there. Your knitted pdf will show both your code and your output. You are encouraged to knit your file as you work to check that your coding and formatting is done so appropriately.

For written responses or multiple choice questions, please bold your (selected) answer.

Grading Details

All questions will be graded full credit (1 point), half credit (0.5 point) or no credit (0 points).

Full credit responses should have the correct response and appropriate code (if applicable). Half credit responses will have a reasonable attempt (typically no more than one small error or oversight), and no credit responses will be either non-attempts or attempts with significant errors.

Exercise 1

Consider the model

$$Y = 2 + 4x + \epsilon$$

where

$$\epsilon \sim N(0, \sigma^2 = x^2).$$

That is

$$\text{Var}[Y \mid X = x] = x^2.$$

Calculate

$$P[Y < -12 \mid X = -3].$$

```
# solution
x = -3
mu_x = 2 + 4 * x
sigma_x = sqrt(x ^ 2)
pnorm(-12, mean = mu_x, sd = sigma_x)
```

```
## [1] 0.2524925
```

Exercise 2

```r
# starter
airquality = na.omit(airquality)
```

For exercises 2 - 13, use `Ozone` as the response and `Temp` as a single predictor. The normality and constant variance assumptions are suspect for a simple model here. We will be exploring some other modeling options with predictor and response transformations!

Consider the quadratic model:

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$$

Fit this model and run the summary of this model.

```r
# solution
fit_quad = lm(Ozone ~ Temp + I(Temp ^ 2), data = airquality)
summary(fit_quad)
```

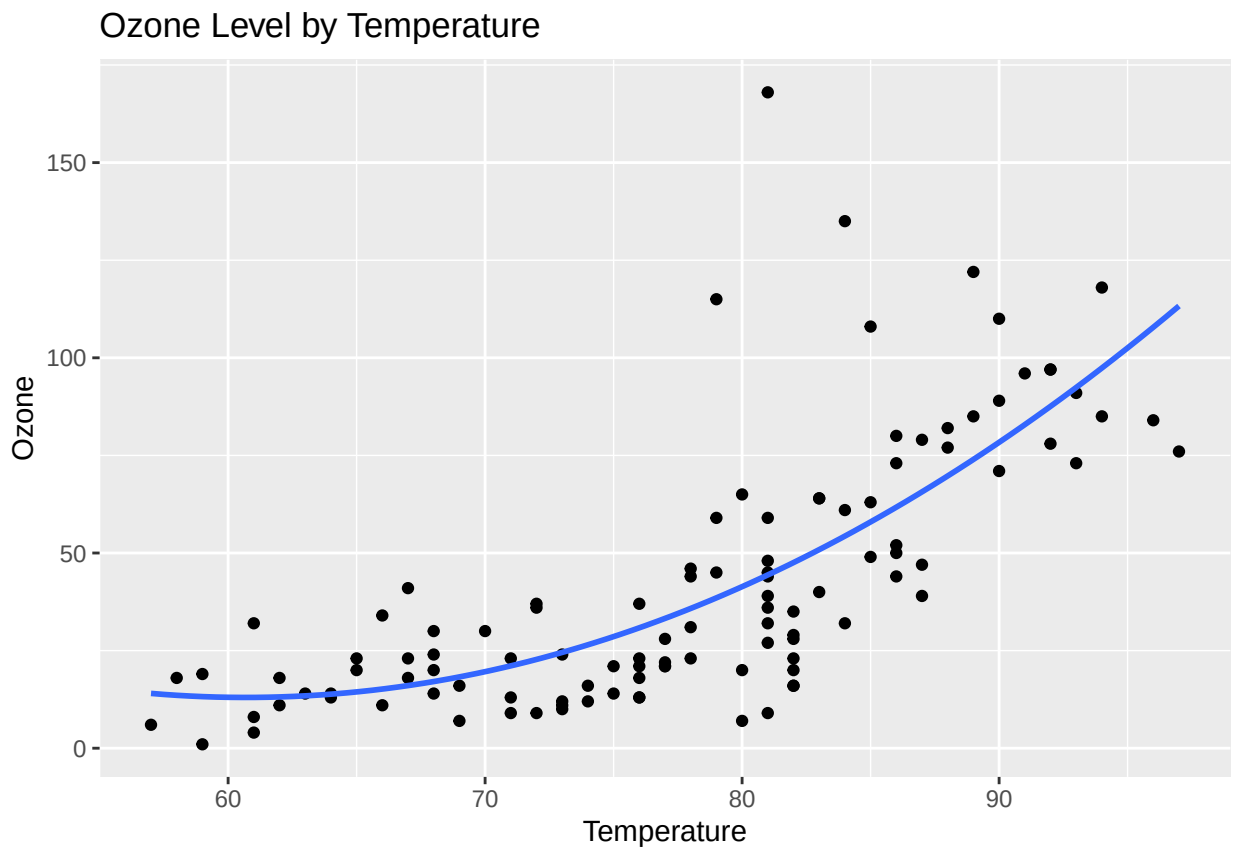
```
##
## Call:
## lm(formula = Ozone ~ Temp + I(Temp^2), data = airquality)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.270 -12.462  -3.072   9.439 123.618
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 292.95885  123.92019   2.364 0.019861 *
## Temp         -9.22680    3.25493  -2.835 0.005476 **
## I(Temp^2)     0.07602    0.02116   3.593 0.000494 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.71 on 108 degrees of freedom
## Multiple R-squared:  0.5426, Adjusted R-squared:  0.5342
## F-statistic: 64.07 on 2 and 108 DF,  p-value: < 2.2e-16
```

Exercise 3

Using the `ggplot2` package, create a scatterplot that uses `Temp` as a predictor of `Ozone`. Add a best fit equation to represent the quadratic model you just fit. Additionally:

- Add a title "Ozone Level by Temperature"
- Adjust the x axis label to "Temperature"
- Adjust the y axis scale to go by 20's
- All other formatting optional

```r
# solution
library(ggplot2)
ggplot(data = airquality, aes(x = Temp, y = Ozone)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), se = FALSE) +
  labs(title = "Ozone Level by Temperature", x = "Temperature")
```



Exercise 4

Run the shapiro wilk test on the residuals of this model.

Also, create a residual plot of the model we created. You may use the base plot features *or* ggplot2 to do this. - Please make sure you label your axes as "fitted" and "residuals" - Give your plot the title "Residuals for 2nd Order Model" - All other formatting optional

Note: To use ggplot2 here, you would need to make a data frame or tibble with two variables–one for the residuals and one for the fitted values. ggplots require a `data = ...` *argument!*

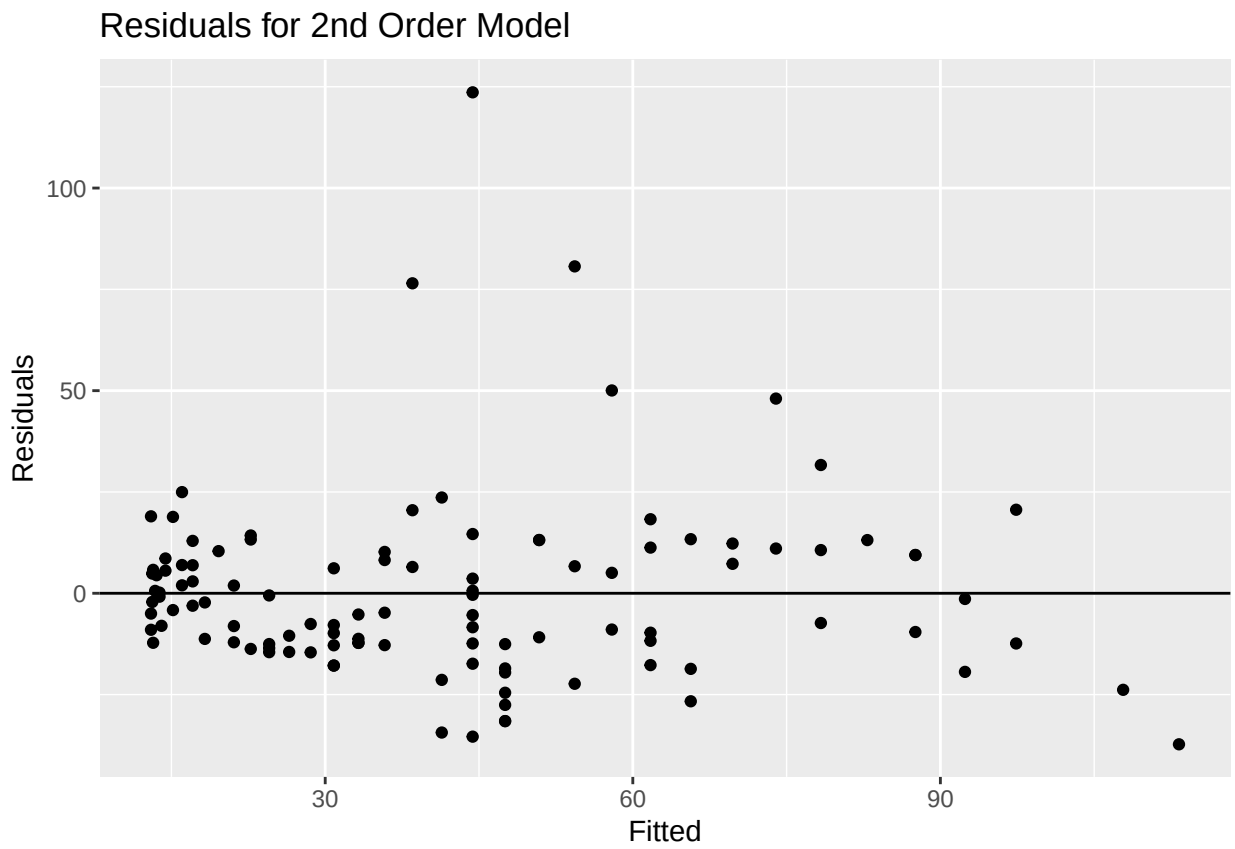
```
#solution
shapiro.test(resid(fit_quad))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(fit_quad)
## W = 0.82306, p-value = 3.249e-10
```

```
plot(resid(fit_quad) ~ fitted(fit_quad),
     main = "Residuals for 2nd Order Model",
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(h = 0)
```


```
#or

fit_data = data.frame(resid = resid(fit_quad), fitted = fitted(fit_quad))
ggplot(data = fit_data, aes(x = fitted, y = resid)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 0) +
  labs(title = "Residuals for 2nd Order Model", y = "Residuals", x = "Fitted")
```



Exercise 5

Based on your original scatterplot, your regression output, the shapiro-wilk test, and your residual plot, which statement makes the most sense?

- The quadratic model does not seem to improve upon the simple model.
- **The quadratic model fits better than the simple model, but there is still some fit issues related to non-normality of the residuals**
- The quadratic model fits better than the simple model, and the residuals appear to be close to normal

Exercise 6

Now, consider the quartic model:

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \epsilon$$

Fit and run a summary of the quartic model.

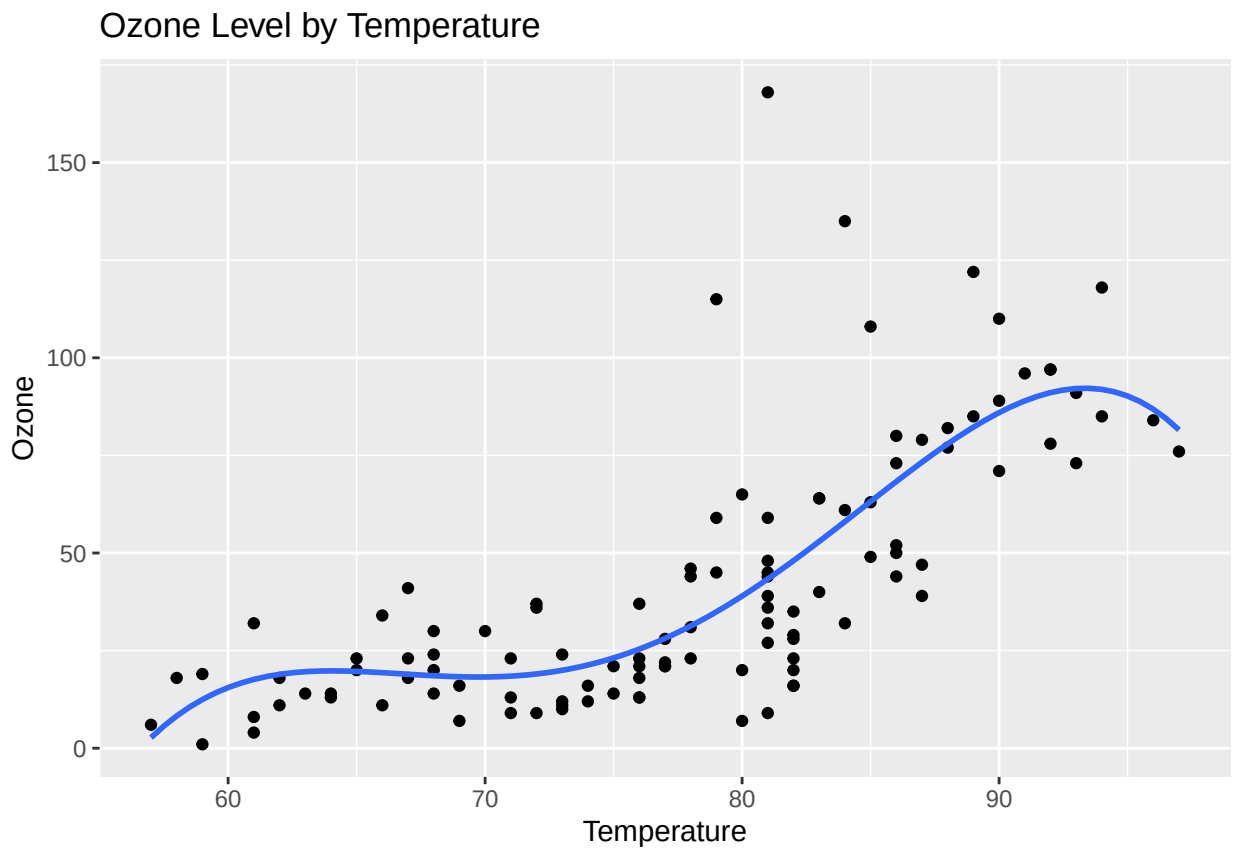
Then, using `ggplot2`, create a scatterplot and add a best fit equation that fits the quartic model (4th order polynomial). Additionally:

- Add a title "Ozone Level by Temperature"
- Adjust the x axis label to "Temperature"
- All other formatting optional

```r
# solution
fit_quar = lm(Ozone ~ Temp + I(Temp ^ 2) + I(Temp ^ 3) + I(Temp ^ 4), data = airquality)
summary(fit_quar)
```

```
##
## Call:
## lm(formula = Ozone ~ Temp + I(Temp^2) + I(Temp^3) + I(Temp^4),
##     data = airquality)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.344 -10.655  -2.386   5.914 124.656
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.469e+04  6.389e+03  -2.298   0.0235 *
## Temp         8.053e+02  3.401e+02   2.368   0.0197 *
## I(Temp^2)   -1.637e+01  6.735e+00  -2.431   0.0167 *
## I(Temp^3)    1.462e-01  5.881e-02   2.486   0.0145 *
## I(Temp^4)   -4.828e-04  1.911e-04  -2.526   0.0130 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.14 on 106 degrees of freedom
## Multiple R-squared:  0.5736, Adjusted R-squared:  0.5575
## F-statistic: 35.65 on 4 and 106 DF,  p-value: < 2.2e-16
```

```r
library(ggplot2)
ggplot(data = airquality, aes(x = Temp, y = Ozone)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2) + I(x^3) + I(x^4), se = FALSE) +
  labs(title = "Ozone Level by Temperature", x = "Temperature")
```



Exercise 7

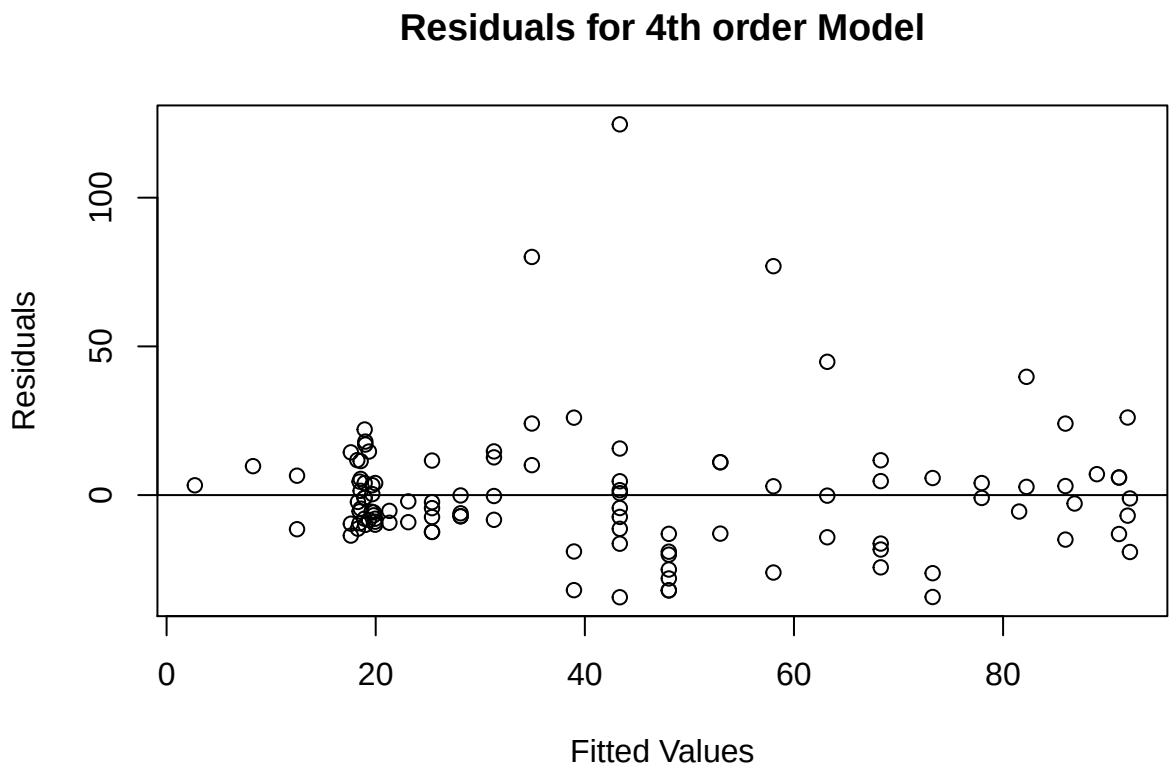
Run the shapiro wilk test on the residuals of this model.

Also, create a residual plot of the model we created. You may use the base plot features *or* ggplot2 to do this. - Please make sure you label your axes as "fitted" and "residuals" - Give your plot the title "Residuals for 4th Order Model" - All other formatting optional

```
# solution
shapiro.test(resid(fit_quar))
```

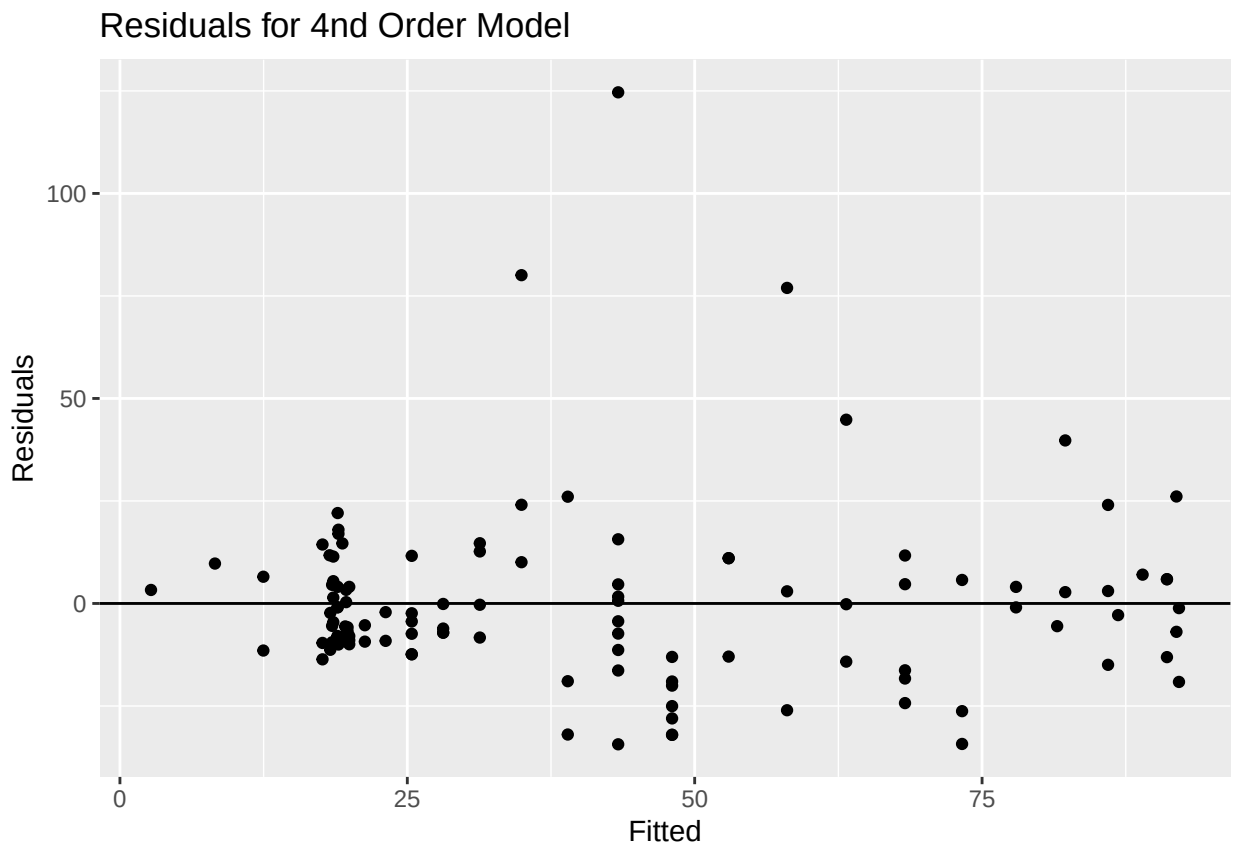
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(fit_quar)
## W = 0.7986, p-value = 5.009e-11
```

```
plot(resid(fit_quar) ~ fitted(fit_quar),
     main = "Residuals for 4th order Model",
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(h = 0)
```




```
#or

fit_data2 = data.frame(resid = resid(fit_quar), fitted = fitted(fit_quar))
ggplot(data = fit_data2, aes(x = fitted, y = resid)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 0) +
  labs(title = "Residuals for 4nd Order Model", y = "Residuals", x = "Fitted")
```



Exercise 8

Now, fit the model using a log transformation

$$\log(y) = \beta_0 + \beta_1 x + \epsilon.$$

Fit and run a summary of the log model.

Then, using `ggplot2`, create a scatterplot and add a best fit equation that fits the log model. To do this, define y as `log(Ozone)` in the aes argument, and then just fit a simple model with `y ~ x` in the formula argument.

Additionally:

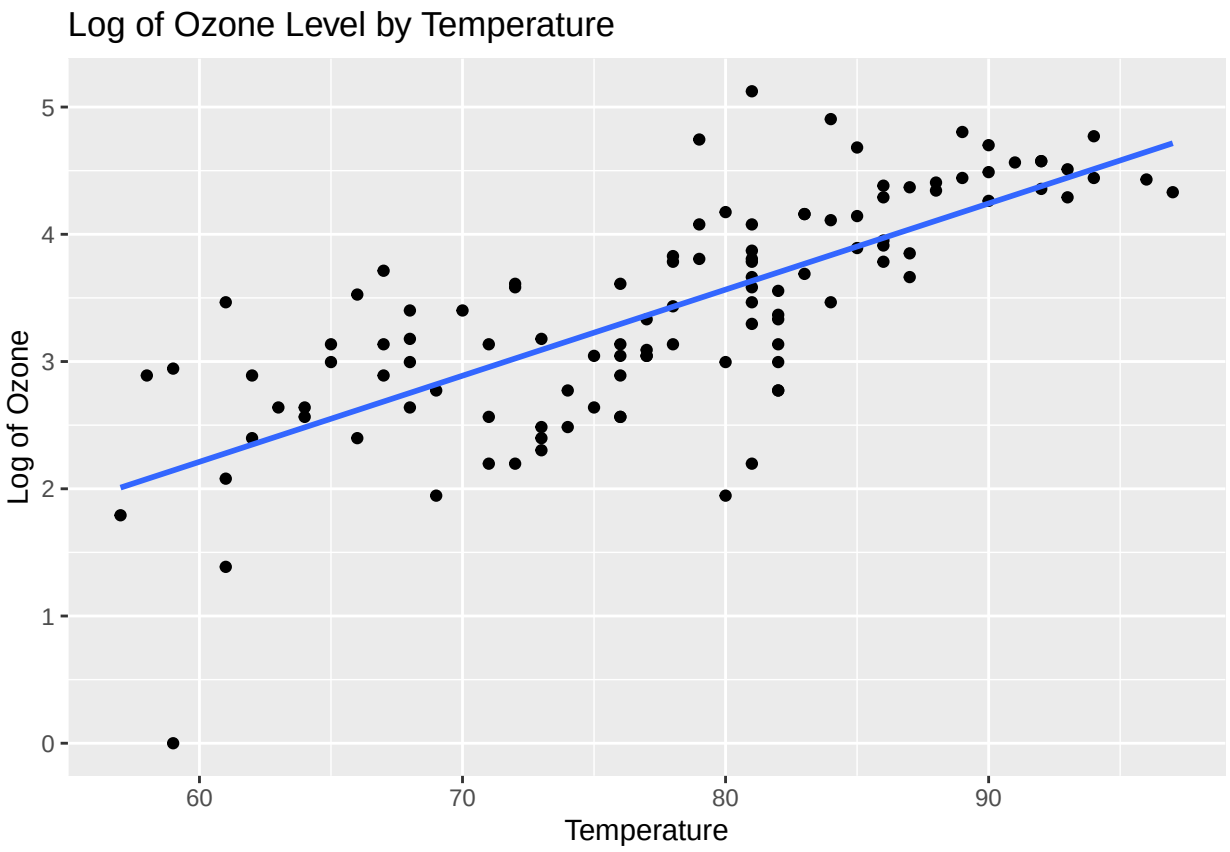
- Add a title "Log of Ozone Level by Temperature" <–Notice this is different!
- Adjust the x axis label to "Temperature"
- Adjust the y axis label to "Log of Ozone" <– Notice this is different!
- All other formatting optional

```
#solution
fit_log = lm(log(Ozone) ~ Temp, data = airquality)
summary(fit_log)
```

```
##
## Call:
```

```
## lm(formula = log(Ozone) ~ Temp, data = airquality)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.14417 -0.32555  0.02066  0.34234  1.49100
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.848518   0.455080  -4.062  9.2e-05 ***
## Temp         0.067673   0.005807  11.654  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5804 on 109 degrees of freedom
## Multiple R-squared:  0.5548, Adjusted R-squared:  0.5507
## F-statistic: 135.8 on 1 and 109 DF,  p-value: < 2.2e-16
```

```r
library(ggplot2)
ggplot(data = airquality, aes(x = Temp, y = log(Ozone))) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x , se = FALSE) +
  labs(title = "Log of Ozone Level by Temperature", x = "Temperature", y = "Log of Ozone")
```



Exercise 9

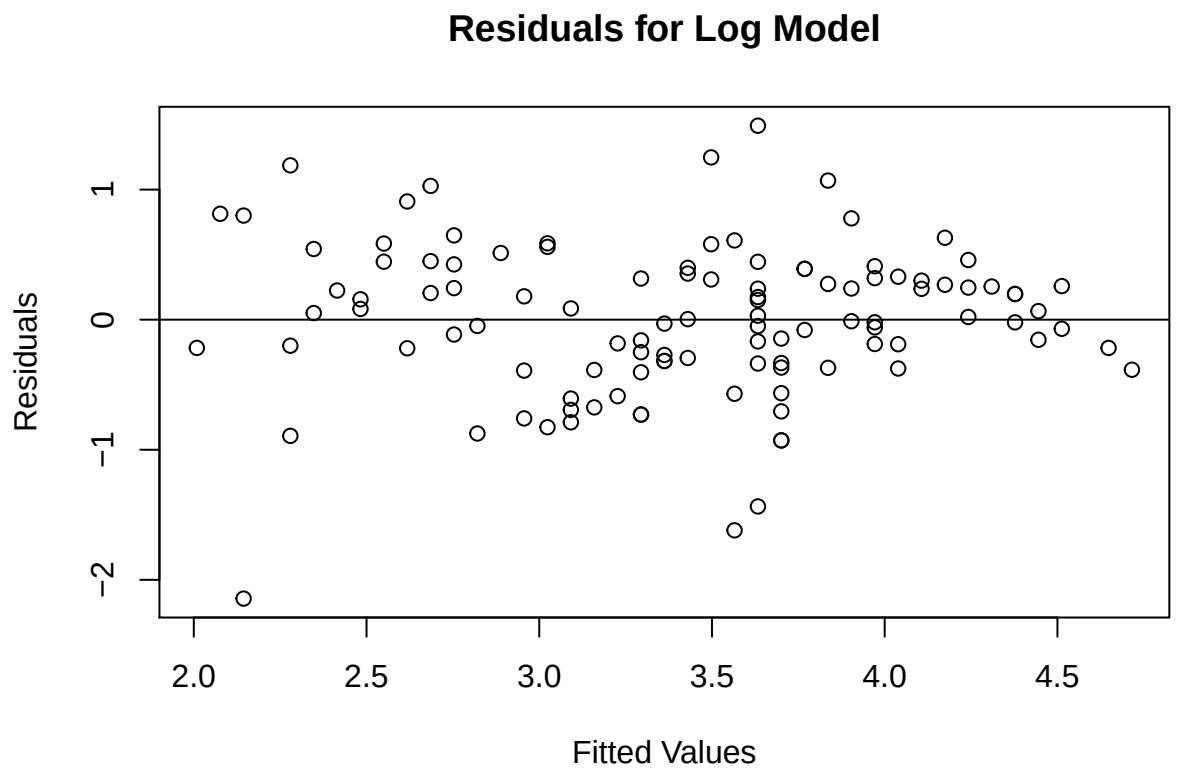
Run the shapiro wilk test on the residuals of this model.

Also, create a residual plot of the model we created. You may use the base plot features *or* ggplot2 to do this. - Please make sure you label your axes as "fitted" and "residuals" - Give your plot the title "Residuals for Log Model" - All other formatting optional

```
# solution
shapiro.test(resid(fit_log))
```

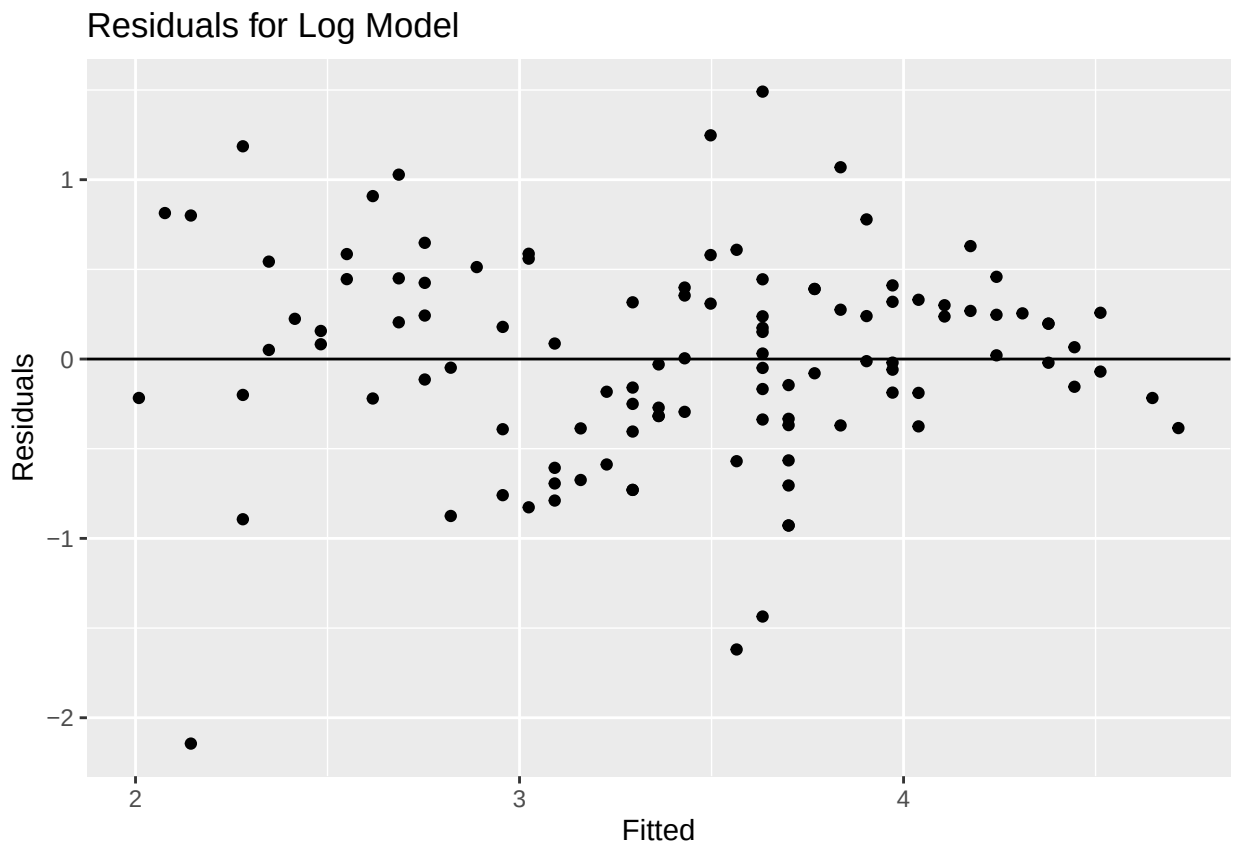
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(fit_log)
## W = 0.97667, p-value = 0.04867
```

```
plot(resid(fit_log) ~ fitted(fit_log),
     main = "Residuals for Log Model",
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(h = 0)
```




```
#or
```

```r
fit_data2 = data.frame(resid = resid(fit_log), fitted = fitted(fit_log))
ggplot(data = fit_data2, aes(x = fitted, y = resid)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 0) +
  labs(title = "Residuals for Log Model", y = "Residuals", x = "Fitted")
```



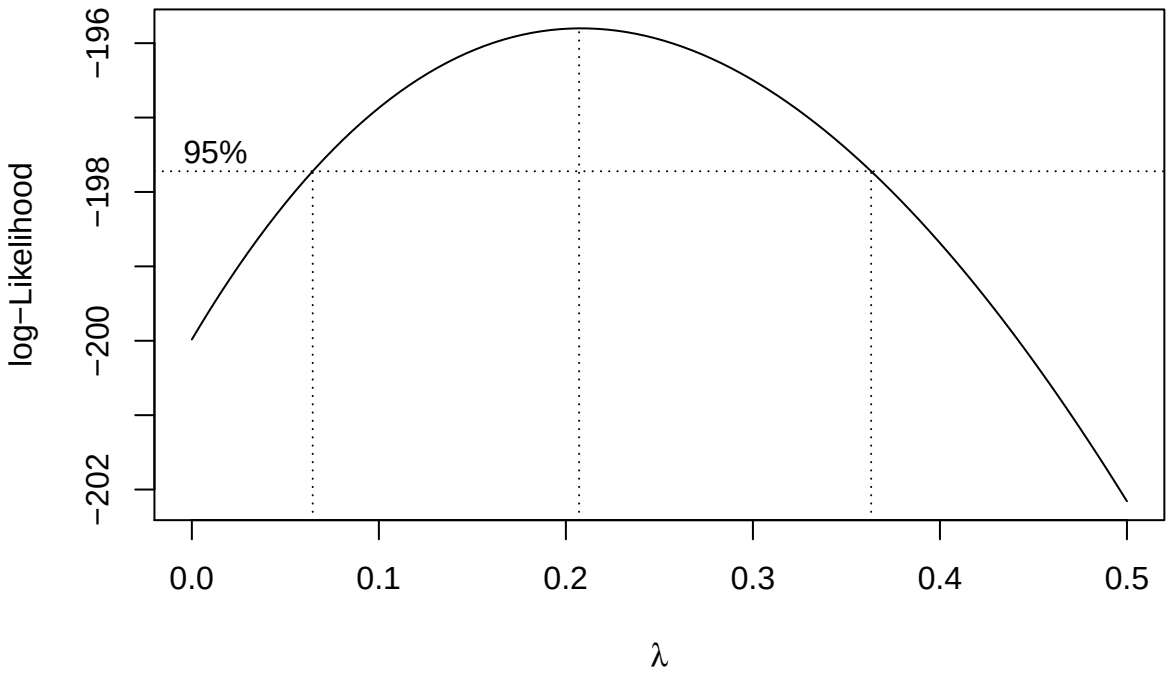
Exercise 10

Lastly, let's explore a Box-Cox Method Transformation for the Response variable.

Fit a simple model with `Ozone` predicted from `Temp`. Then using the `MASS` package, run the `boxcox` function on this model. Be sure to plot the result.

Hint: Consider narrowing down the lambda window to a more targeted range.

```r
# solution
library(MASS)
model = lm(Ozone ~ Temp, data = airquality)
boxcox(model, plotit = TRUE, lambda = seq(0, 0.5, 0.05))
```



Exercise 11

Using your results from the previous exercise, Fit a model that transforms the response variable by the appropriate lambda value. Go ahead and round your lambda value to the nearest tenth (For example, 0.1 or 0.2 or 0.3,. . .).

Run a summary of that model.

Then, using `ggplot2`, create a scatterplot and add a best fit equation that fits this model. Just like with the log model, you'll need to adjust y in the aes argument, and then just fit a simple model with `y ~ x` in the formula argument.

Additionally:

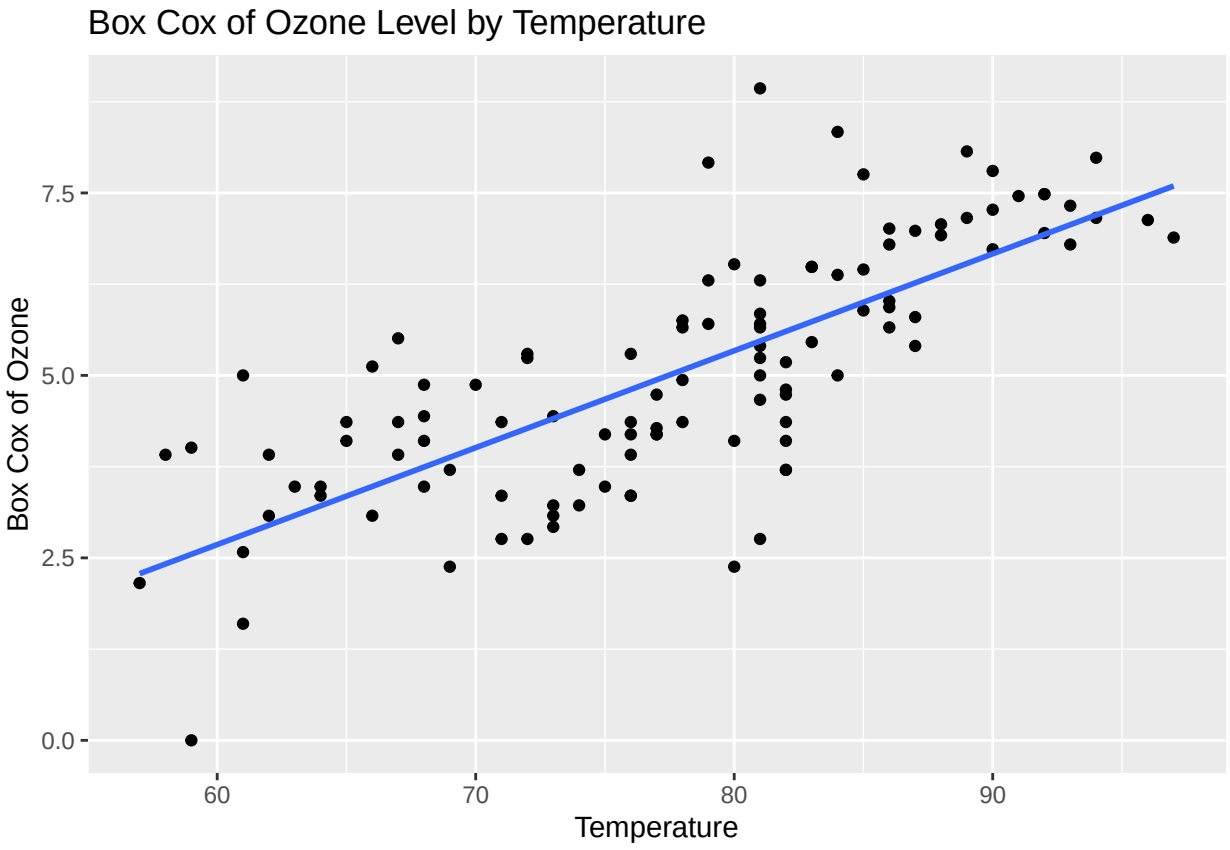
- Add a title "Box-Cox of Ozone Level by Temperature" <–Notice this is different!
- Adjust the x axis label to "Temperature"
- Adjust the y axis label to "Box Cox of Ozone" <– Notice this is different!
- All other formatting optional

```
#solution
fit_box = lm((((Ozone ^ 0.2) - 1) / 0.2) ~ Temp, data = airquality)
summary(fit_box)
```

```
##
## Call:
## lm(formula = (((Ozone^0.2) - 1)/0.2) ~ Temp, data = airquality)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9590 -0.7474  0.0192  0.6886  3.4620
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.28554    0.86787   -6.09 1.72e-08 ***
## Temp         0.13279    0.01107   11.99  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.107 on 109 degrees of freedom
## Multiple R-squared:  0.5688, Adjusted R-squared:  0.5649
## F-statistic: 143.8 on 1 and 109 DF,  p-value: < 2.2e-16
```

```
library(ggplot2)
ggplot(data = airquality, aes(x = Temp, y = ((Ozone ^ 0.2) - 1) / 0.2)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x , se = FALSE) +
  labs(title = "Box Cox of Ozone Level by Temperature", x = "Temperature", y = "Box Cox of Ozone")
```



Exercise 12

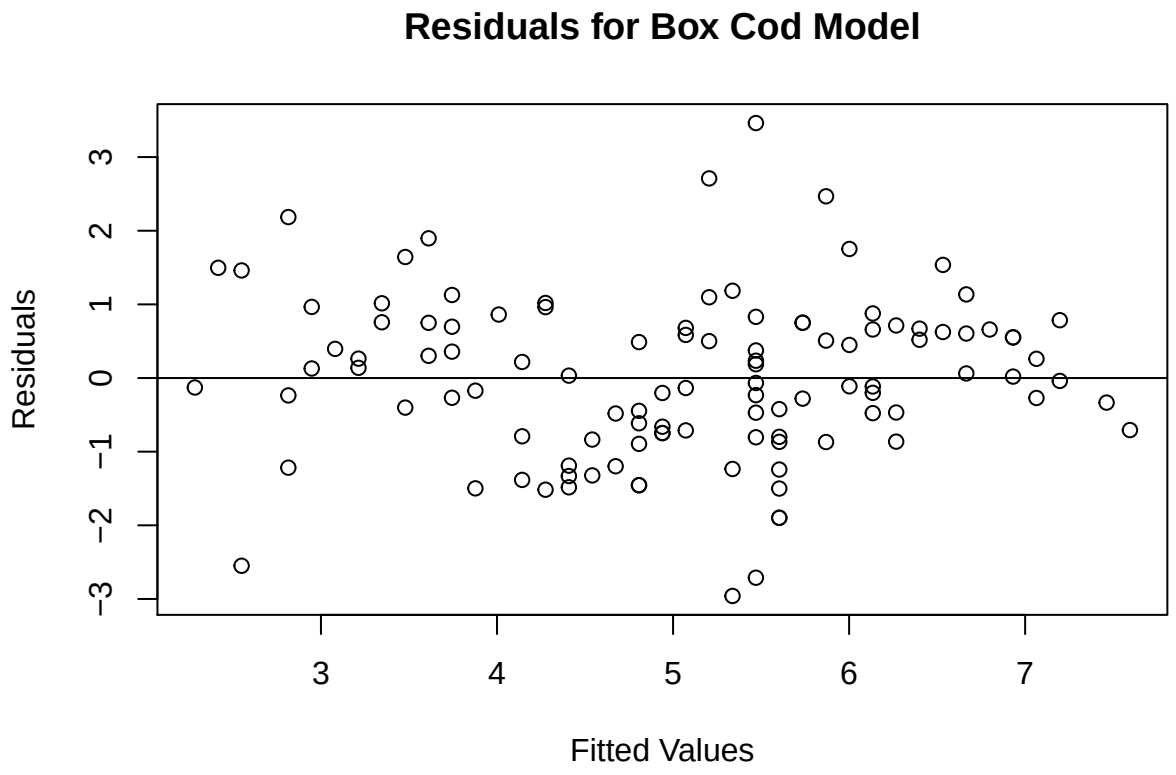
Run the shapiro wilk test on the residuals of this model.

Also, create a residual plot of the model we created. You may use the base plot features *or* ggplot2 to do this. - Please make sure you label your axes as "fitted" and "residuals" - Give your plot the title "Residuals for Box Cox Model" - All other formatting optional

```
# solution
shapiro.test(resid(fit_box))
```

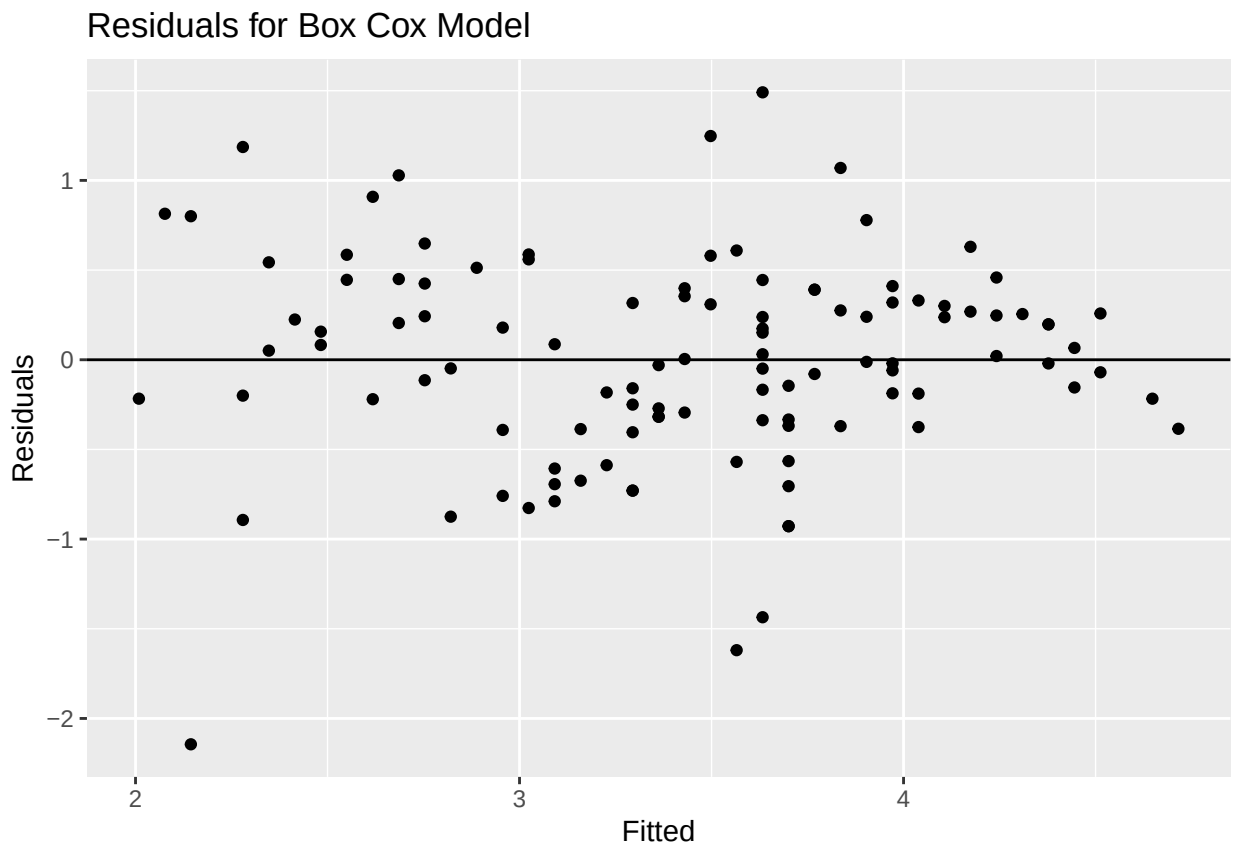
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(fit_box)
## W = 0.98893, p-value = 0.5021
```

```
plot(resid(fit_box) ~ fitted(fit_box),
     main = "Residuals for Box Cod Model",
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(h = 0)
```




```
#or

fit_data2 = data.frame(resid = resid(fit_log), fitted = fitted(fit_log))
ggplot(data = fit_data2, aes(x = fitted, y = resid)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 0) +
  labs(title = "Residuals for Box Cox Model", y = "Residuals", x = "Fitted")
```



Exercise 13

We have investigated 4 alternative model options to the simple linear fit. Which of these four alternative models do you believe fits normality and constant variance assumption the best?

Note: We've seen several other considerations for choosing a model as well: Lowest RMSE on new data, Highest R^2, F-test comparison of models... in the next two chapters, we'll spend more time on variable selection and choosing a sensible model!

- Quadratic model
- Quartic model
- Log Model
- **Box Cox Model**