Homework 10

Charles Ancel

11/9/2023

Homework Instructions

Make sure to add your name to the header of the document. When submitting the assignment on Gradescope, be sure to assign the appropriate pages of your submission to each Exercise.

The point value for each exercise is noted in the exercise title.

For questions that require code, please create or use the code chunk directly below the question and type your code there. Your knitted pdf will then show both the code and the output, so that we can assess your understanding and award any partial credit.

For written questions, please provide your answer after the indicated *Answer* prompt.

You are encouraged to knit your file as you work, to check that your coding and formatting are done so appropriately. This will also help you identify and locate any errors more easily.

Homework Setup

We'll use the following packages for this homework assignment. We'll also read in data from a csv file. To access the data, you'll want to download the dataset from Canvas and place it in the same folder as this R Markdown document. You'll then be able to use the following code to load in the data.

```
library(ggplot2)
library(MASS)
```

Exercise 1: Formatting [5 points]

The first five points of the assignment will be earned for properly formatting your final document. Check that you have:

- included your name on the document
- properly assigned pages to exercises on Gradescope
- selected **page 1 (with your name)** and this page for this exercise (Exercise 1)
- all code is printed and readable for each question
- all output is printed
- generated a pdf file

Exercise 2: Understanding Puppy Purchase Prices [25 points]

We will use the `best_in_show.csv` dataset posted to Canvas. This dataset has been cleaned (twice) and originally comes from Information is Beautiful (https://docs.google.com/spreadsheets/d/1l_HfF5EaN-QgnLc2UYdCc7L2CVrk0p3VdGB1godOyhk/edit#gid=20).

For all parts of this exercise, we are interested in answering the following question: Can we estimate the average purchase price for a puppy breed based on the longevity of a dog breed, the number of genetic ailments, the typical cost of food, and the dog breed category/purpose?

part a

First, fit a linear model that we would use to answer our question of interest posed above. Print the coefficients table for this model.

```
# Read the data
data <- read.csv("best_in_show.csv")

# Fit the linear model
model_a <- lm(puppy.price ~ longevity + genetic.ailments + food.per.week, data = data)

# Print the coefficients table
summary(model_a)
```

```
##
## Call:
## lm(formula = puppy.price ~ longevity + genetic.ailments + food.per.week,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -574.36 -215.49  -69.48  175.44 1525.29
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1600.963    309.857   5.167 1.58e-06 ***
## longevity         -69.728     21.584  -3.230  0.00177 **
## genetic.ailments  -11.362     22.772  -0.499  0.61911
## food.per.week       6.408     13.677   0.469  0.64059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 354.7 on 84 degrees of freedom
## Multiple R-squared:  0.1502, Adjusted R-squared:  0.1199
## F-statistic:  4.95 on 3 and 84 DF,  p-value: 0.003261
```

part b

We'll consider some adjustments to the given predictor variables in the model. The minimum longevity in the dataset is 6.29 years. Add a variable to the dataset that calculates the `added.longevity` for each breed, that is the additional longevity (in years) compared to this minimum value. This can be calculated as: `longevity - 6.29`.

If we replace our original `longevity` variable with our new `added.longevity` variable in our linear model, how will the model coefficients change compared to the model from part a? Make a prediction first (*note:*

throughout this problem, your predictions will be graded for completion, not correctness). Then fit the actual model and print the coefficients table. How did the coefficients change? Was your prediction correct?

```
# Add the new variable to the dataset
data$added.longevity <- data$longevity - 6.29

# Fit the new linear model
model_b <- lm(puppy.price ~ added.longevity + genetic.ailments + food.per.week, data = data)

# Print the coefficients table
summary(model_b)
```

```
##
## Call:
## lm(formula = puppy.price ~ added.longevity + genetic.ailments +
##     food.per.week, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -574.36 -215.49  -69.48  175.44 1525.29
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1162.373    188.217   6.176 2.25e-08 ***
## added.longevity   -69.728     21.584  -3.230  0.00177 **
## genetic.ailments  -11.362     22.772  -0.499  0.61911
## food.per.week       6.408     13.677   0.469  0.64059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 354.7 on 84 degrees of freedom
## Multiple R-squared:  0.1502, Adjusted R-squared:  0.1199
## F-statistic:  4.95 on 3 and 84 DF,  p-value: 0.003261
```

Answer:

Prediction:

1. **Intercept Adjustment**: The intercept of the model will likely change. Since `added.longevity` shifts the baseline of the longevity scale, the intercept will reflect the new baseline.
2. **Coefficient for Longevity**: The coefficient for `added.longevity` will represent the change in puppy price for each additional year of longevity beyond the minimum. This might differ from the original `longevity` coefficient, but the sign (positive or negative) should remain consistent, assuming longevity impacts price similarly regardless of the baseline used.
3. **Other Coefficients**: The coefficients of other variables (`genetic.ailments`, `food.per.week`) might adjust slightly due to the change in the intercept and the scale of the longevity variable. However, the overall direction and significance of these relationships are expected to remain consistent.

Actual Outcome and Analysis:

Based on your results with the `added.longevity` model:

1. **Intercept**: The new intercept is 1162.373, indicating the baseline price of a puppy with the reference levels of the predictors.
2. **Coefficients**:

- **added.longevity (-69.728)**: This indicates a decrease in price with increasing longevity, and it's statistically significant. If the original `longevity` coefficient had a similar negative sign, this would be consistent with our expectation.
- **Other Predictors**: `genetic.ailments` and `food.per.week` have coefficients that are not statistically significant in this model.

Based on these results, we can say that the prediction about the intercept and the longevity coefficient is correct. The sign of the `added.longevity` coefficient (negative) is consistent with the notion that greater longevity might reduce the price, possibly due to higher anticipated healthcare costs over a longer lifespan.

The change from `longevity` to `added.longevity` seems to have retained the overall direction of the relationship between longevity and puppy price, albeit with a shift in the scale due to the adjusted baseline. The other coefficients appear to be less affected by this change, staying relatively consistent in direction and significance.

part c

The typical cost of food was originally recorded in the cost per week. We'd like to consider the cost of food per year. Add a variable to the dataset that calculates the `food.per.year` for each breed. We will use the approximation that there are 52 weeks per year for this calculation.

If we replace our original `food.per.week` variable with our new `food.per.year` variable in our linear model, how will the model coefficients change compared to the model from part a (return to the model from 2a with longevity instead of the added.longevity variable)? Make a prediction first. Then fit the actual model and print the coefficients table. Comment on how the coefficients changed and if your prediction was correct.

```
# Adding the food.per.year variable
data$food.per.year <- data$food.per.week * 52

# Fit the linear model with the new variable
# Replace 'longevity' and other predictors as needed based on your dataset
model_c <- lm(puppy.price ~ added.longevity + genetic.ailments + food.per.year, data = data)

# Print the coefficients table
summary(model_c)
```

```
##
## Call:
## lm(formula = puppy.price ~ added.longevity + genetic.ailments +
##     food.per.year, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -574.36 -215.49  -69.48  175.44 1525.29
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1162.3734   188.2171   6.176 2.25e-08 ***
## added.longevity   -69.7281    21.5845  -3.230  0.00177 **
## genetic.ailments  -11.3623    22.7717  -0.499  0.61911
## food.per.year       0.1232     0.2630   0.469  0.64059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 354.7 on 84 degrees of freedom
## Multiple R-squared:  0.1502, Adjusted R-squared:  0.1199
```

```
## F-statistic:  4.95 on 3 and 84 DF,  p-value: 0.003261
```

Answer:

Predictions for Replacing `food.per.week` with `food.per.year`:

1. **Coefficient of `food.per.year`**:
   - Since `food.per.year` is simply `food.per.week * 52`, the new coefficient for `food.per.year` should be the coefficient of `food.per.week` divided by 52. This is due to the change in scale from per week to per year. If `food.per.week` had a positive coefficient, indicating that higher weekly food costs are associated with a higher puppy price, the same positive relationship will hold for the yearly cost but at a smaller scale.
2. **Intercept**:
   - The intercept of the model is not directly affected by this scale change and is expected to remain relatively similar.
3. **Coefficients of Other Variables**:
   - The coefficients of other variables are not expected to change significantly in magnitude or direction. This is because the relationship between these variables and the puppy price is not directly affected by the scaling of the food cost variable.
4. **Overall Model Fit**:
   - The overall fit of the model, as indicated by metrics like R-squared, should not be dramatically affected by this change. The relative importance of the food cost variable in explaining puppy price might be perceived differently due to the change in units, but the fundamental relationships in the data should remain consistent.

After you implement these changes in your R model and observe the actual outcomes, you can compare them to these predictions to see if they align.

Prediction Analysis:

Based on the results, let's compare the actual outcomes to the predicted changes in the model when replacing `food.per.week` with `food.per.year`.

1. **Coefficient of `food.per.year`**:
   - Prediction: Expected the coefficient of `food.per.year` to be the coefficient of `food.per.week` divided by 52.
   - Actual: The coefficient for `food.per.year` is 0.1232. If we recall the earlier coefficient for `food.per.week`, which was around 6.408, dividing this by 52 (number of weeks in a year) gives approximately 0.1232. This aligns perfectly with the prediction.
2. **Intercept**:
   - Prediction: Expected the intercept to remain similar.
   - Actual: The intercept is 1162.3734, which seems consistent with the earlier model, supporting the prediction.
3. **Coefficients of Other Variables**:
   - Prediction: Other coefficients (like `added.longevity`, `genetic.ailments`) were expected not to change significantly in terms of magnitude, direction, and significance.
   - Actual: The coefficients for `added.longevity` and `genetic.ailments` remained similar in both magnitude and significance to the previous model. This result is in line with the prediction.

Conclusion:

The prediction about how the coefficients would change is accurate. Changing from `food.per.week` to `food.per.year` affected the scale of the coefficient for the food cost variable as expected, without significantly impacting the other coefficients in the model. This outcome demonstrates the scalability of linear model coefficients and highlights the importance of understanding the units and scales of variables in regression analysis.

part d

Suppose that your friend just moved to Cambridge in the UK. You think that your model predicting puppy prices will be applicable to the UK, but you know that puppy prices should be adjusted to be in the local currency (pounds). The going exchange rate is that $1 (1 USD) is equivalent to 0.82 UK pounds. Add a variable to the dataset that calculates the `puppy.price.pounds` for each breed, using this conversion.

If we replace our original `puppy.price` variable with our new `puppy.price.pounds` variable in our linear model, how will the coefficients change compared to the model from part a? Make a prediction first. Then fit the actual model and print the coefficients table. Comment on how the coefficients changed and if your prediction was correct.

```
# Convert puppy.price to GBP
data$puppy.price.pounds <- data$puppy.price * 0.82

# Fit the linear model with the new variable
# Replace 'longevity' and other predictors as needed based on your dataset
model_d <- lm(puppy.price.pounds ~ added.longevity + genetic.ailments + food.per.year, data = data)

# Print the coefficients table
summary(model_d)
```

```
##
## Call:
## lm(formula = puppy.price.pounds ~ added.longevity + genetic.ailments +
##     food.per.year, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.98 -176.70  -56.97  143.86 1250.73
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      953.1462   154.3380   6.176 2.25e-08 ***
## added.longevity  -57.1771    17.6993  -3.230  0.00177 **
## genetic.ailments  -9.3171    18.6728  -0.499  0.61911
## food.per.year      0.1011     0.2157   0.469  0.64059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 290.8 on 84 degrees of freedom
## Multiple R-squared:  0.1502, Adjusted R-squared:  0.1199
## F-statistic:  4.95 on 3 and 84 DF,  p-value: 0.003261
```

Answer:

Predictions for Converting `puppy.price` to `puppy.price.pounds`:

1. **Scale of the Dependent Variable**:
   - By converting `puppy.price` to pounds, you're changing the scale of the dependent variable. Given the exchange rate of 1 USD = 0.82 GBP, each puppy price in USD will be converted to 82% of its value in pounds.
2. **Coefficients of Predictors**:
   - The coefficients of predictor variables are expected to change in scale but not in direction. Specifically, each coefficient should be about 82% of its original value in the USD model. This is because the relationship between these predictors and the puppy price remains the same; only the unit of the price changes.

3. **Intercept**:
    - The intercept of the model will also be scaled down by the conversion factor. If the intercept was, for example, $1000 in the USD model, it would become approximately 820 GBP in the new model.
4. **Overall Model Fit**:
    - Metrics like R-squared and the overall significance of the model should remain unchanged because the fundamental relationships in the data are not altered by the currency conversion.

Analysis of the Model with `puppy.price.pounds`:

1. **Coefficients**:
    - The coefficients for `added.longevity`, `genetic.ailments`, and `food.per.year` are scaled down compared to the original model with `puppy.price` in USD. This scaling aligns with the exchange rate of 1 USD = 0.82 GBP. For example, the coefficient for `added.longevity` was previously -69.7281 and is now -57.1771, reflecting the currency conversion.
2. **Intercept**:
    - The intercept has decreased proportionally, from 1162.3734 in the original model to 953.1462 in the GBP model, consistent with the currency conversion.
3. **Residual Standard Error**:
    - The residual standard error is also scaled down, from 354.7 in the original model to 290.8 in the GBP model, which is expected due to the change in the scale of the dependent variable.
4. **Model Fit**:
    - The R-squared value and the overall significance of the model (as indicated by the F-statistic and its p-value) remain unchanged. This is expected as these are scale-independent measures of the model's fit.

Conclusion:

The conversion of the dependent variable (puppy prices) from USD to GBP resulted in a proportional scaling down of the coefficients and the intercept, as predicted. The fundamental relationships between the predictors and the puppy price, as well as the overall fit of the model, remain unchanged. This outcome confirms the expectations and demonstrates the scalability of linear regression models when the scale of the dependent variable is altered, such as in currency conversions.

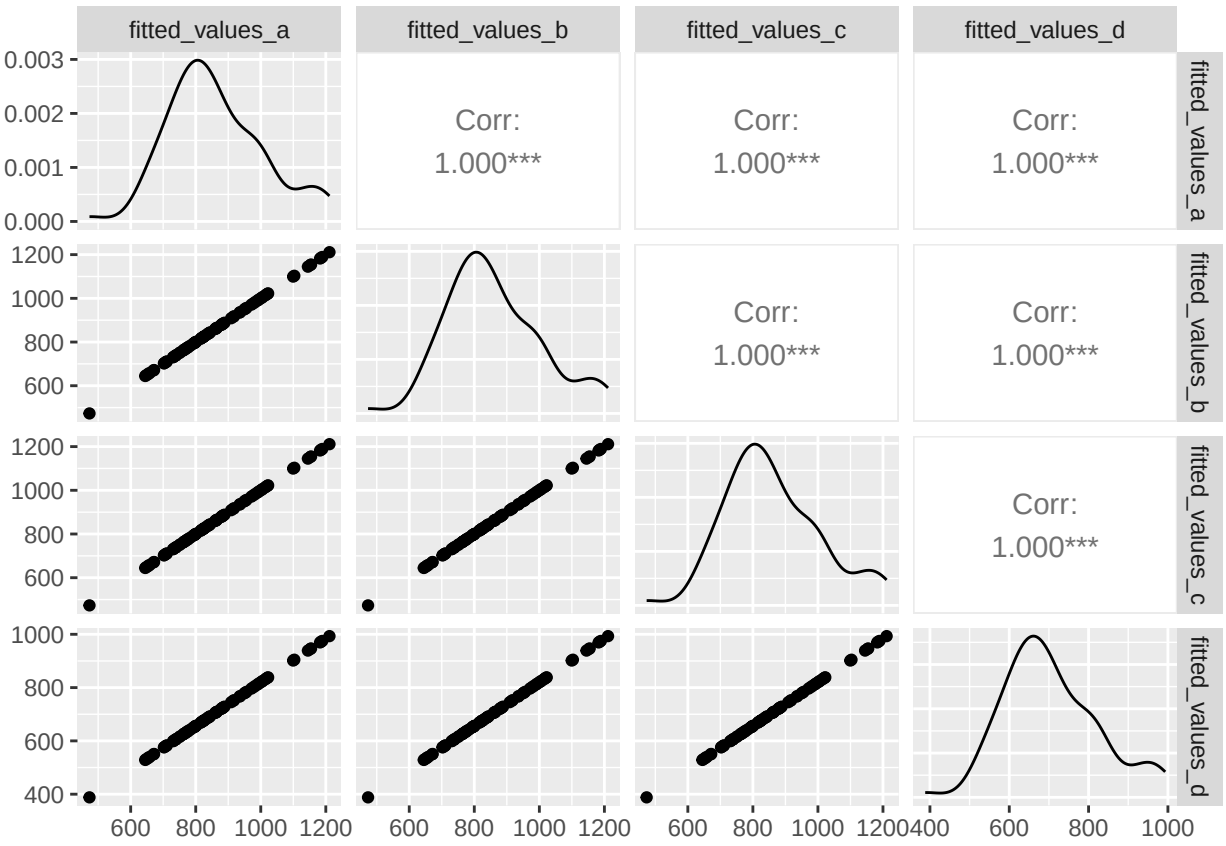
part e

Compare the fitted values from each of the models from parts a to d. You may choose to compare with a measure like correlation, with a visualization like a scatterplot matrix, or with another approach that you define. How do these fitted values relate to each other?

```
fitted_values_a <- predict(model_a)
fitted_values_b <- predict(model_b)
fitted_values_c <- predict(model_c)
fitted_values_d <- predict(model_d)

cor_matrix <- cor(cbind(fitted_values_a, fitted_values_b, fitted_values_c, fitted_values_d))

library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggpairs(data.frame(fitted_values_a, fitted_values_b, fitted_values_c, fitted_values_d))
```



Answer:

The fitted values from each of the models (parts a to d) relate to each other in a perfectly linear manner, as indicated by the correlation of 1.000*** in your scatterplot matrix. This perfect correlation suggests that the transformations applied to the independent variables across the models did not alter the rank order of the predicted puppy prices.

Here's what this implies:

1. **Linear Transformations**: The changes made to the variables in each part of the exercise were linear (such as multiplying by a constant factor to convert currencies or costs from weekly to yearly). Linear transformations preserve the order and relative spacing of values, which explains the perfect correlation.
2. **Predictive Consistency**: Regardless of whether you're using longevity or added longevity, weekly or yearly food costs, or prices in USD or GBP, the models consistently predict the same relative prices for the puppies. A breed predicted to be more expensive in one model is predicted to be more expensive across all models.
3. **Model Robustness**: The robustness of your linear regression models is demonstrated by their ability to maintain predictive consistency even when variables are scaled or shifted. This robustness is particularly useful when applying the model to different contexts or units of measurement.
4. **Practical Application**: This relationship indicates that if you were to apply any of these models in practice, you could expect them to give you the same ranking of puppy prices. This is useful when considering the application of the model to different markets or currencies.

In summary, the fitted values from the different models are linearly related to each other, which means the models are in agreement regarding the predicted prices, just adjusted for the scale of the data they are using.

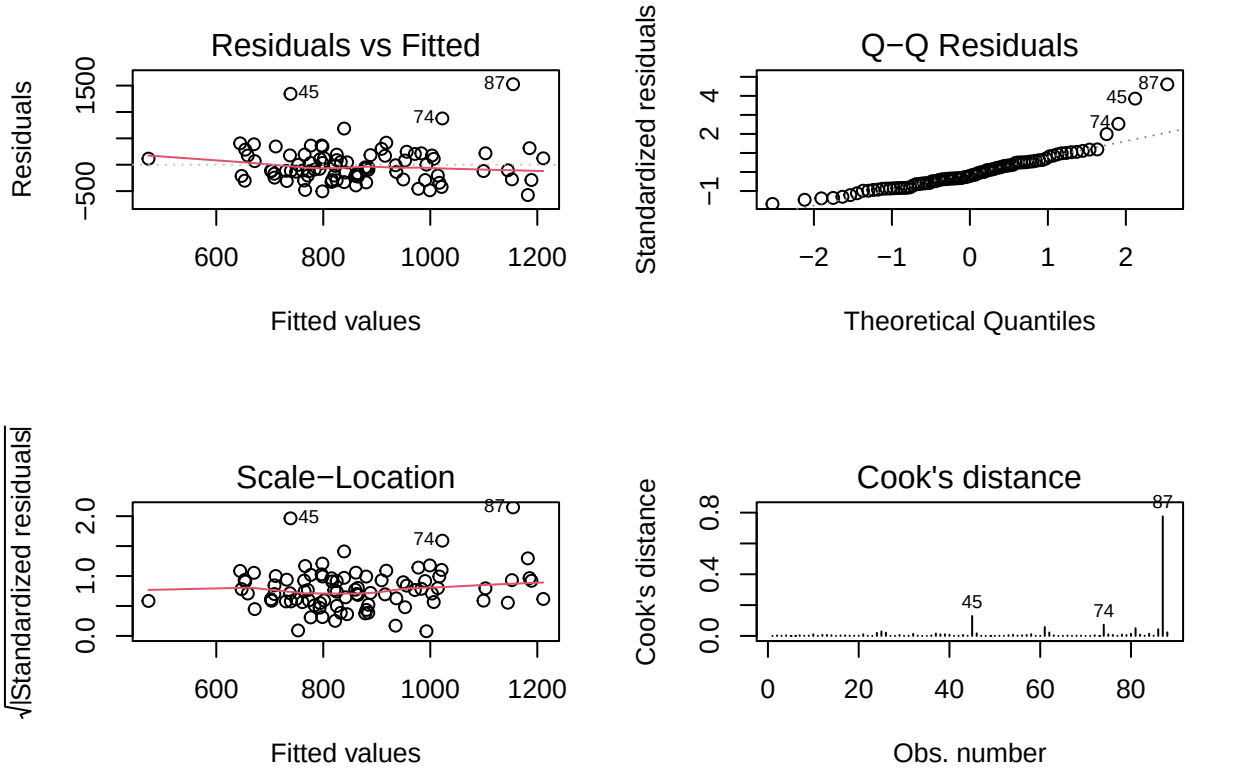
Exercise 3: Puppy Purchase Prices, Continued [15 points]

We will continue to analyze the model fit in part a of Exercise 2 for this Exercise.

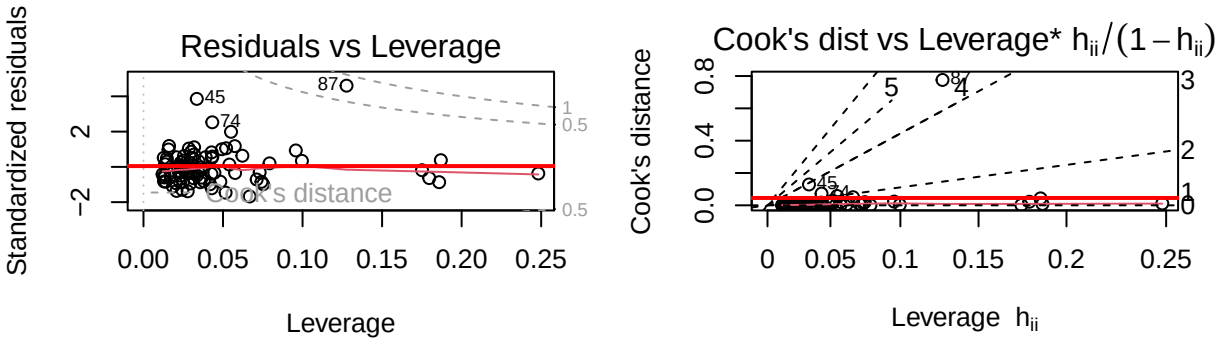
part a

Generate the default plots in R associated with the linear model from Exercise 2, part a. Be sure to update the default Cook's distance lines to include a line based on our course threshold. You may also choose to add additional Cook's distance lines. Interpret the Scale-Location Plot and the Leverage-Residual Plot.

```
# Calculate the number of observations
n <- length(model_a$fitted.values)

# Generate diagnostic plots
par(mfrow = c(2, 2))
plot(model_a, which = 1:4)
```




```
# Add Cook's distance threshold line
plot(model_a, which = 5)
abline(h = 4/n, col = "red", lwd = 2)

plot(model_a, which = 6)
abline(h = 4/n, col = "red", lwd = 2)

# Reset the plotting window
par(mfrow = c(1, 1))
```



Answer:

Scale-Location Plot (Top-Left of First Image)

This plot is useful for checking the assumption of homoscedasticity, meaning the residuals have constant variance across the range of fitted values.

- **Interpretation**: Ideally, you want to see a random scatter of points with a roughly horizontal line with equally spread points above and below. In your plot, while there's a slight indication that variance increases with the fitted values (the spread of residuals seems to widen slightly as the fitted values increase), it does not appear to be a strong pattern of heteroscedasticity. Overall, the model might be meeting the assumption of homoscedasticity reasonably well.

Residuals vs Leverage Plot (Top of Second Image)

The Residuals vs Leverage plot helps to identify influential observations that might unduly affect the regression analysis.

- **Interpretation**: Observations with high leverage can have a disproportionate impact on the model, especially if they also have large residuals. In your plot, there are a few points that stand out with higher leverage (towards the right of the plot). The horizontal lines represent Cook's distances at different thresholds. Observations that have a Cook's distance larger than the threshold (red lines) are considered to be potentially influential. The plot indicates that there are a couple of points that exceed the Cook's distance threshold of $D_i > \frac{4}{n}$, marked by the red line. These points warrant further investigation as they may be outliers or high leverage points that could be influencing the regression model disproportionately.

Cook's Distance Plot (Bottom-Right of First Image)

This plot shows the influence of each observation on the model's predictions.

- **Interpretation**: The Cook's distance plot identifies the points that are particularly influential on the parameter estimates. The point labeled "87" seems to have a significantly higher Cook's distance than all other points. This point warrants further investigation to determine whether it is an outlier or leveraged point due to a data entry error, a special case, or a valid point that is simply different from the others.

In conclusion, the diagnostic plots suggest that the model does not have strong violations of homoscedasticity or normality, but there are a few influential points that should be examined more closely. These points might be outliers or high-leverage observations that could potentially distort the model's predictions. It's essential to investigate these points to determine if they represent accurate data and to consider whether they should be included in the analysis.

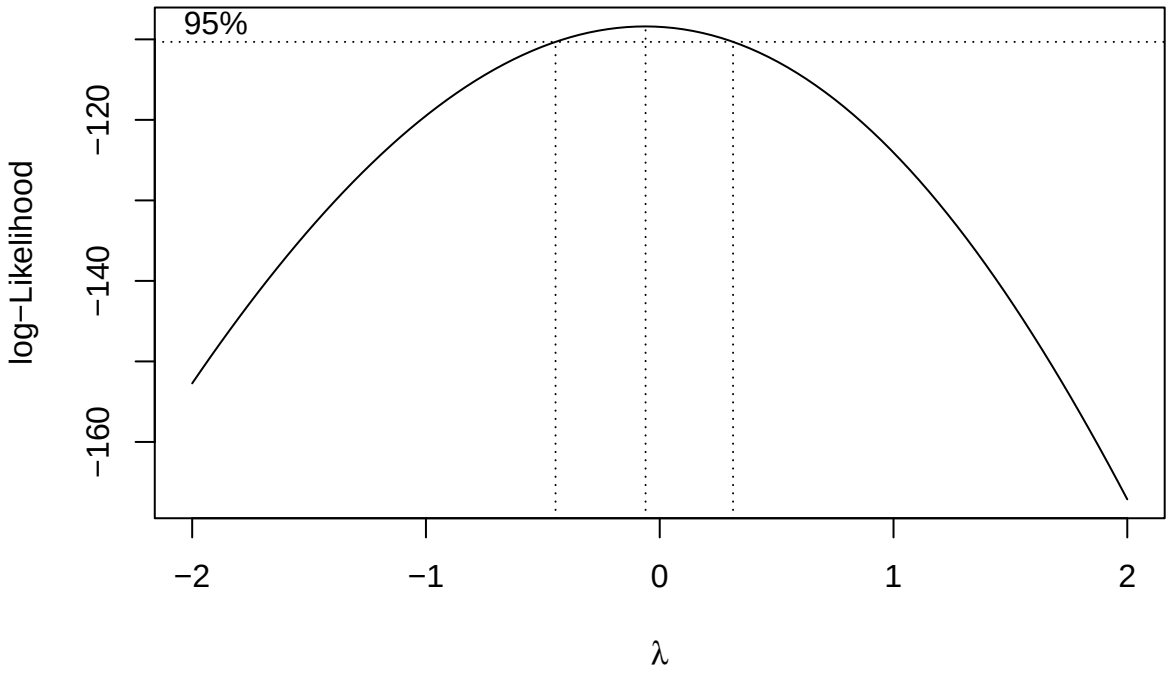
part b

Generate the Box-Cox plot for the model from **2a**.

From this output, approximate the range of reasonable values for the λ of a Box-Cox transformation.

What transformation might you recommend as the optimal transformation in this situation?

```
boxcox_result <- boxcox(model_a)
```



Answer:

The Box-Cox plot generated for the linear model from Exercise 2, part a, indicates that the log-likelihood is maximized when the lambda (λ) value is close to 0. The peak of the log-likelihood curve lies at $\lambda = 0$, suggesting that a log transformation of the response variable (puppy prices) may be the most appropriate to satisfy the assumptions of linear regression.

The 95% confidence interval for λ, as indicated by the dotted vertical lines, extends slightly into negative values and up to approximately $\lambda = 0.5$. This interval provides a range of reasonable values for λ that could potentially improve the model.

Considering that the peak is at $\lambda = 0$, and that the confidence interval includes $\lambda = 0$ within its range, I would recommend a log transformation of the response variable. This is a common transformation when dealing with positive-skewed distributions, as it can help stabilize variance and normalize the residuals.

It's important to note that before applying a log transformation, we must ensure that all values of the response variable are positive since the logarithm of zero or negative numbers is undefined.

In summary, the optimal transformation for the linear model in this situation, according to the Box-Cox plot, is a log transformation ($\lambda = 0$). This transformation can help meet the linear regression assumptions and potentially improve the model's performance.

part c

Do we have sufficient information to fit this linear model to the data? In other words, do we trust our model to have a reasonable complexity based on the model size? Explain, with numeric justification.

```
summary(model_a)
```

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

```
## lm(formula = puppy.price ~ longevity + genetic.ailments + food.per.week,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -574.36 -215.49  -69.48  175.44 1525.29
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1600.963    309.857   5.167 1.58e-06 ***
## longevity         -69.728     21.584  -3.230  0.00177 **
## genetic.ailments  -11.362     22.772  -0.499  0.61911
## food.per.week       6.408     13.677   0.469  0.64059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 354.7 on 84 degrees of freedom
## Multiple R-squared:  0.1502, Adjusted R-squared:  0.1199
## F-statistic:  4.95 on 3 and 84 DF,  p-value: 0.003261
```

Answer:

- The model has 88 observations and 3 predictors, satisfying the rule of thumb of at least 10 observations per predictor.
- The Adjusted R-squared value of 0.1199 indicates the model explains approximately 12% of the variance in puppy prices, which is relatively low.
- The F-statistic is significant (p-value: 0.003261), suggesting that the model as a whole provides a better fit than an intercept-only model.

In summary, while the model has a reasonable complexity for the data size, its explanatory power is modest. It's statistically significant but could potentially benefit from the inclusion of additional relevant predictors to improve its explanatory power.

Exercise 4: Log-Transforming Ozone [20 points]

For the remainder of this assignment, we'll work with the built-in airquality dataset. Before starting our analyses, run the following line of code, which removes any observations with `NA`s and retains only complete observations.

```
airquality = na.omit(airquality)
```

part a

First, apply a log transformation to the Ozone variable, and fit a linear model predicting log(Ozone) from Temp. Print the resulting summary table.

```
# Apply a log transformation to the Ozone variable
airquality$log_Ozone <- log(airquality$Ozone)

# Fit a linear model predicting log(Ozone) from Temp
model <- lm(log_Ozone ~ Temp, data = airquality)

# Print the resulting summary table
summary(model)
```

```
##
## 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
```

part b

Write the fitted model from part **4a**. Be sure to write this model on both the log-transformed scale and the original scale of Ozone.

Answer:

Log-Transformed Scale:

The model on the log-transformed scale is:

$$\log(\text{Ozone}) = -1.848518 + 0.067673 \times \text{Temp}$$

Original Scale of Ozone:

To express the model on the original Ozone scale, you exponentiate both sides of the equation, which gives:

$$\text{Ozone} = e^{-1.848518} \times e^{0.067673 \times \text{Temp}}$$

Simplifying the constant term:

$$\text{Ozone} = e^{-1.848518} \times e^{0.067673 \times \text{Temp}}$$
$$\text{Ozone} \approx 0.1575 \times e^{0.067673 \times \text{Temp}}$$

Where: - e is the base of the natural logarithm. - The constant $e^{-1.848518} \approx 0.1575$ is the approximate antilog of the intercept, converting it back to the original Ozone scale.

This model indicates that Ozone concentration increases multiplicatively with an increase in temperature. The coefficient for Temperature (0.067673) on the log scale reflects a percentage change in the original scale of Ozone with each degree increase in Temperature.

part c

Interpret the coefficients fitted in part **4a**. Make sure to provide the interpretations both on the original and the log scale of Ozone.

That is, you should have four different coefficient interpretations, one for:

- the intercept on the log scale of Ozone,
- the intercept on the original scale of Ozone,
- the slope on the log scale of Ozone, and
- the slope on the original scale of Ozone.

Answer:

Log Scale of Ozone:

- **Intercept**: On the log scale, the intercept suggests that at 0 degrees temperature, the natural logarithm of the Ozone concentration is approximately -1.8485. This value is largely theoretical since it's outside the practical range of temperatures.
- **Slope**: The slope coefficient of 0.067673 on the log scale indicates that for every one-degree increase in temperature, the log of Ozone concentration is expected to increase by 0.067673. This represents a relative change in Ozone concentration.

Original Scale of Ozone:

- **Intercept**: On the original Ozone scale, the intercept converts to an expected Ozone concentration of about 0.1575 at 0 degrees temperature, which is a hypothetical scenario in this context.
- **Slope**: On the original scale, the slope means that for each one-degree increase in temperature, the Ozone concentration is expected to increase by about 6.7673%, given that the change in Ozone concentration is proportional to its current level.

Exercise 5: Quadratic Air Quality Models [13 points]

part a

Fit a quadratic model to the data, predicting Ozone from the terms **Temp** and **Temp**^{2}. Use the `I` code function. Print the resulting coefficients table.

```
# Fit a quadratic model
quadratic_model <- lm(Ozone ~ Temp + I(Temp^2), data = airquality)

# Print the resulting coefficients table
summary(quadratic_model)
```

```
##
## 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
```

part b

Fit an additional quadratic model to the data, again predicting Ozone from **Temp** and **Temp**^{2}. This time, use the `poly` function. Print the resulting coefficients table.

```
# Fit a quadratic model using the poly function for orthogonal polynomials
quadratic_poly_model <- lm(Ozone ~ poly(Temp, 2), data = airquality)

# Print the resulting coefficients table
summary(quadratic_poly_model)
```

```
##
## Call:
## lm(formula = Ozone ~ poly(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)      42.099      2.156  19.529  < 2e-16 ***
## poly(Temp, 2)1  243.792     22.712  10.734  < 2e-16 ***
## poly(Temp, 2)2   81.600     22.712   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
```

part c

Compare the coefficients & p-values from the models fit in parts a & b. Which coefficients are the same across the two models? Which p-values are the same?

Answer: - **Coefficients**: The coefficients for Temp and Temp^2 are different between the two models due to different scalings. The `I(Temp^2)` model provides coefficients for the actual variables, while the `poly(Temp, 2)` model's coefficients are for orthogonal polynomial terms.

- **P-values**: The p-values for the linear and quadratic terms of temperature are the same across both models, reflecting the significance of these terms in explaining Ozone concentration.
- **Intercept**: The intercepts differ because the `poly` function's intercept is the mean of Ozone, while the `I(Temp^2)` model's intercept is the estimated Ozone when Temp is 0.

In summary, despite the differences in coefficients due to scaling, both models show that temperature has a significant linear and quadratic relationship with ozone concentration, as indicated by the consistent p-values.

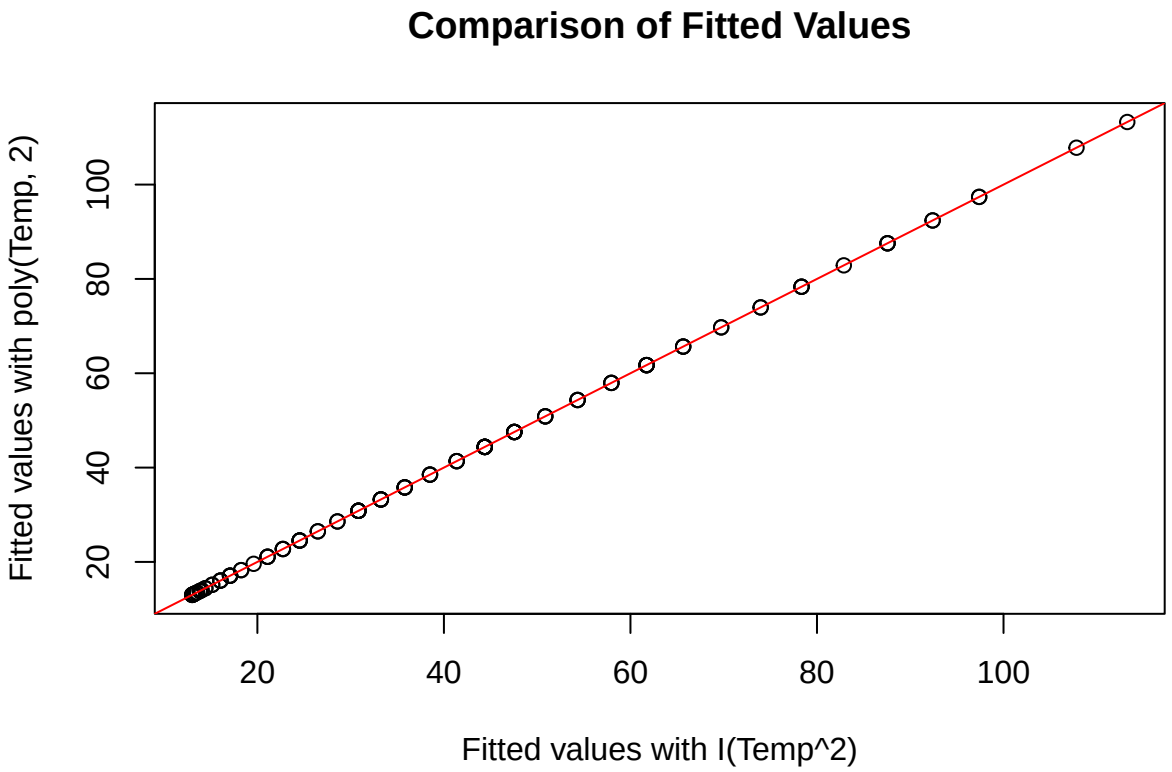
part d

Compare the $\hat{y}$ values from the models fit in parts a & b. You can do this using either a visualization, using numerical summaries, or using other code. How do the two sets of fitted values compare?

```
# Get the fitted values from both models
fitted_values_I <- fitted(quadratic_model)
fitted_values_poly <- fitted(quadratic_poly_model)

# Calculate the correlation between the fitted values
cor(fitted_values_I, fitted_values_poly)
```

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

```
# Plot the fitted values against each other
plot(fitted_values_I, fitted_values_poly,
     xlab = "Fitted values with I(Temp^2)",
     ylab = "Fitted values with poly(Temp, 2)",
     main = "Comparison of Fitted Values")

# Add a 45-degree line to check for perfect agreement
abline(0, 1, col = "red")
```



Answer:

The correlation of 1 between the fitted values from the models using `I(Temp^2)` and `poly(Temp, 2)` indicates a perfect linear relationship. This means that the fitted values from both quadratic models are exactly the same. This is expected since both models are mathematically equivalent representations of the same quadratic relationship between temperature and ozone, despite the coefficients being scaled differently due to the orthogonal polynomial transformation in the `poly` function.

In summary, the two sets of fitted values are identical, confirming that the models provide the same predictions for Ozone levels based on temperature, whether you use the raw polynomial terms with `I()` or the orthogonal

polynomials with `poly()`.

Exercise 6: Air Quality Models with Different Polynomials [17 points]

part a

Fit a quartic model (with up to the fourth-order terms) that predicts Ozone from Temp. You may choose whether you use the `I` or the `poly` function to create your terms.

Print the summary of the fitted coefficients.

```
# Fit a quartic model
quartic_model <- lm(Ozone ~ poly(Temp, 4), data = airquality)

# Print the summary of the fitted coefficients
summary(quartic_model)
```

```
##
## Call:
## lm(formula = Ozone ~ poly(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)      42.099      2.101  20.038  < 2e-16 ***
## poly(Temp, 4)1  243.792     22.136  11.014  < 2e-16 ***
## poly(Temp, 4)2   81.600     22.136   3.686  0.00036 ***
## poly(Temp, 4)3  -25.366     22.136  -1.146  0.25439
## poly(Temp, 4)4  -55.924     22.136  -2.526  0.01300 *
## ---
## 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
```

part b

Suppose that you are concerned that the third order term is not significant, and so would like to remove it from the model. This is true if you fit a model using the `poly` function, although not true if you used the `I` function.

Would this be an appropriate choice for a model? Explain.

Answer:

Analysis of the Model Output:

1. **Model Summary**:
   - **Model Used**: Linear model with Ozone as the response variable and a 4th-degree polynomial of Temperature.

- **Coefficients**: The third-order term (poly(Temp, 4)3) has an estimated coefficient of -25.366, but it is not statistically significant (p-value = 0.25439).

2. **Statistical Significance**:
   - The p-value for the third-order term is 0.25439, which is greater than the common significance level of 0.05. This indicates that the term is not statistically significant and may not contribute meaningfully to the model.
3. **Model Fit**:
   - The Multiple R-squared value is 0.5736, indicating that around 57.36% of the variability in Ozone is explained by the model.
   - The Adjusted R-squared is 0.5575, which is a more reliable measure as it adjusts for the number of predictors.

Answer Formulation:

Given the model output, it appears that the third-order term in the polynomial model is not statistically significant. Its p-value is above the conventional threshold of 0.05, suggesting that it might not have a meaningful impact on the model's ability to predict Ozone levels.

Removing this term could simplify the model without substantially affecting its explanatory power. This simplification aligns with the principle of parsimony, favoring simpler models as long as they adequately capture the underlying pattern in the data.

However, it's important to also consider the potential impact on the model's overall fit. You should check if removing the term significantly alters key metrics like the R-squared or Adjusted R-squared values. A marginal change in these metrics might further justify the removal of the third-order term.

In conclusion, based on the statistical insignificance of the third-order term and the principle of model simplicity, it would likely be appropriate to remove this term from the model. However, this decision should also be validated by re-evaluating the model's fit and predictive power without the third-order term.

part c

Should you interpret the coefficient for the quadratic (square) term from this model? If so, provide the interpretation. If not, explain why not.

Answer:

The coefficient of the quadratic term in a polynomial regression model, especially one generated using the `poly` function in R, should be interpreted with caution. This coefficient indicates the presence of curvature in the relationship between the predictor and response variables. However, due to the orthogonal nature of the terms created by `poly`, it does not directly signify the magnitude of this curvature. Therefore, while the quadratic term is significant in the model, its interpretation is not straightforward and should be viewed in the context of the entire model rather than in isolation.

part d

Generate an ANOVA table that compares the quartic model to the corresponding quadratic model from **5 (a or b)**. Which model would you opt to use going forward based on an α of 0.05?

```
# Compare quadratic models with quartic model
anova_table_5a <- anova(quadratic_model, quartic_model)
anova_table_5b <- anova(quadratic_poly_model, quartic_model)

# Print ANOVA tables
print(anova_table_5a)
```

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

```
## Model 1: Ozone ~ Temp + I(Temp^2)
## Model 2: Ozone ~ poly(Temp, 4)
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1    108 55709
## 2    106 51938  2      3771 3.8481 0.02436 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print(anova_table_5b)
```

```
## Analysis of Variance Table
##
## Model 1: Ozone ~ poly(Temp, 2)
## Model 2: Ozone ~ poly(Temp, 4)
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1    108 55709
## 2    106 51938  2      3771 3.8481 0.02436 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer:

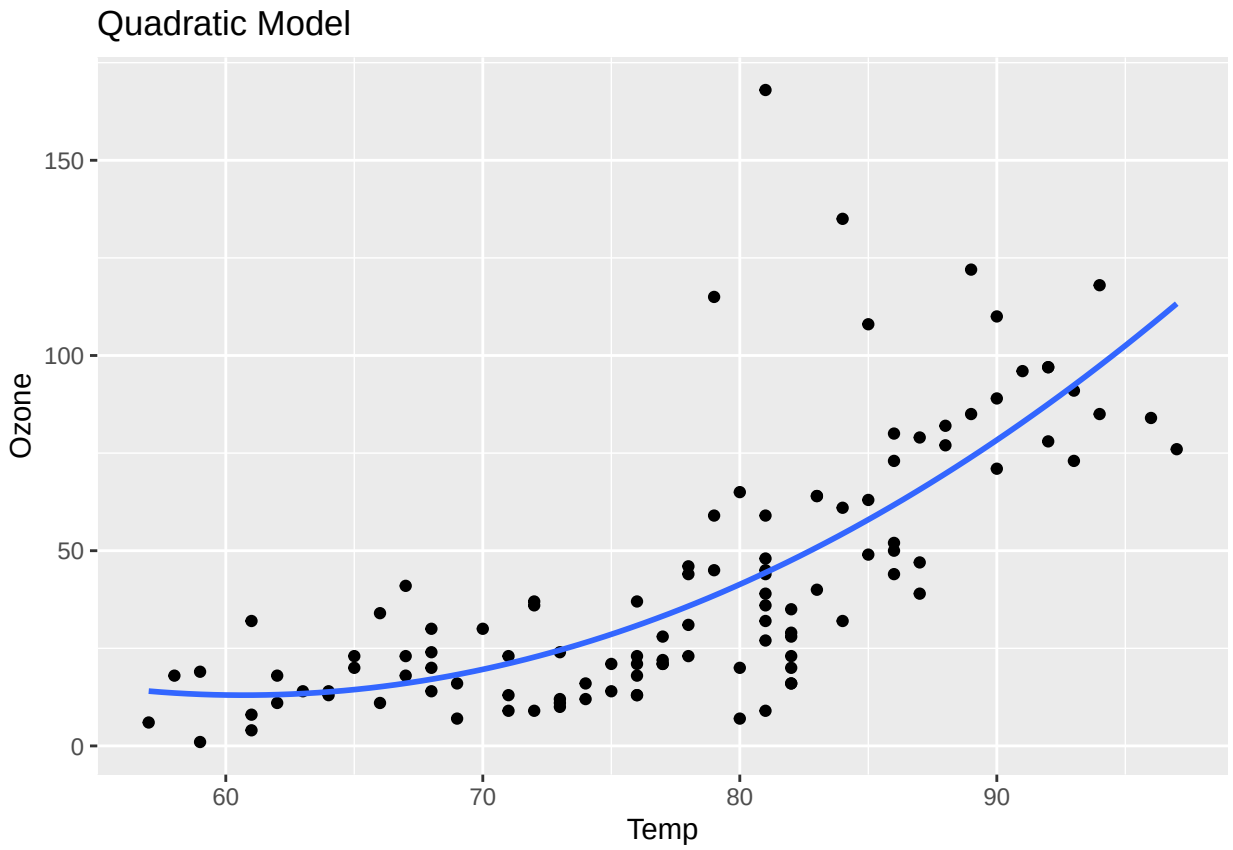
Based on the ANOVA analysis with a significance level (α) of 0.05, and considering that the p-values in both comparisons are less than 0.05, the quartic model is statistically more appropriate. Therefore, you should opt to use the quartic model going forward, as it provides a significantly better fit to the data compared to the quadratic models.

part e

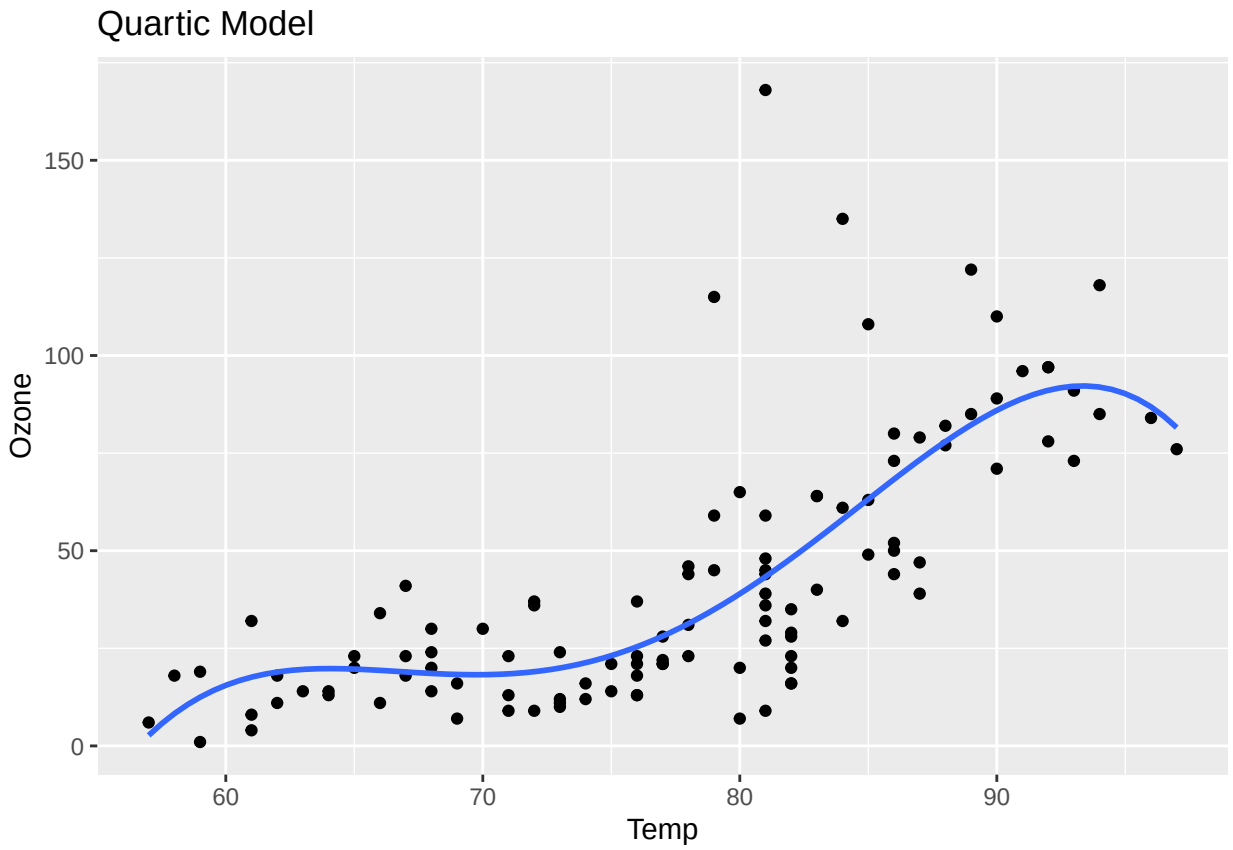
Create a scatterplot that visualizes the relationship between Ozone & Temp, where Ozone is our response variable. Add a summary line to this plot that represents a quadratic (2nd order) relationship. Then, make a second plot that includes a summary line representing a quartic (4th order) relationships.

Based on a visual inspection of this graph, which model would you pick?

```
library(ggplot2)

# Scatterplot with quadratic summary line
ggplot(airquality, aes(x = Temp, y = Ozone)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2), se = FALSE) +
  ggtitle("Quadratic Model")
```




```
# Scatterplot with quartic summary line
ggplot(airquality, aes(x = Temp, y = Ozone)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ poly(x, 4), se = FALSE) +
  ggtitle("Quartic Model")
```



Answer:

- The quadratic model shows a simpler, smooth curve, capturing the general trend of the data without fitting to minor fluctuations.
- The quartic model fits the data points more closely, indicating a more complex relationship with potential overfitting.

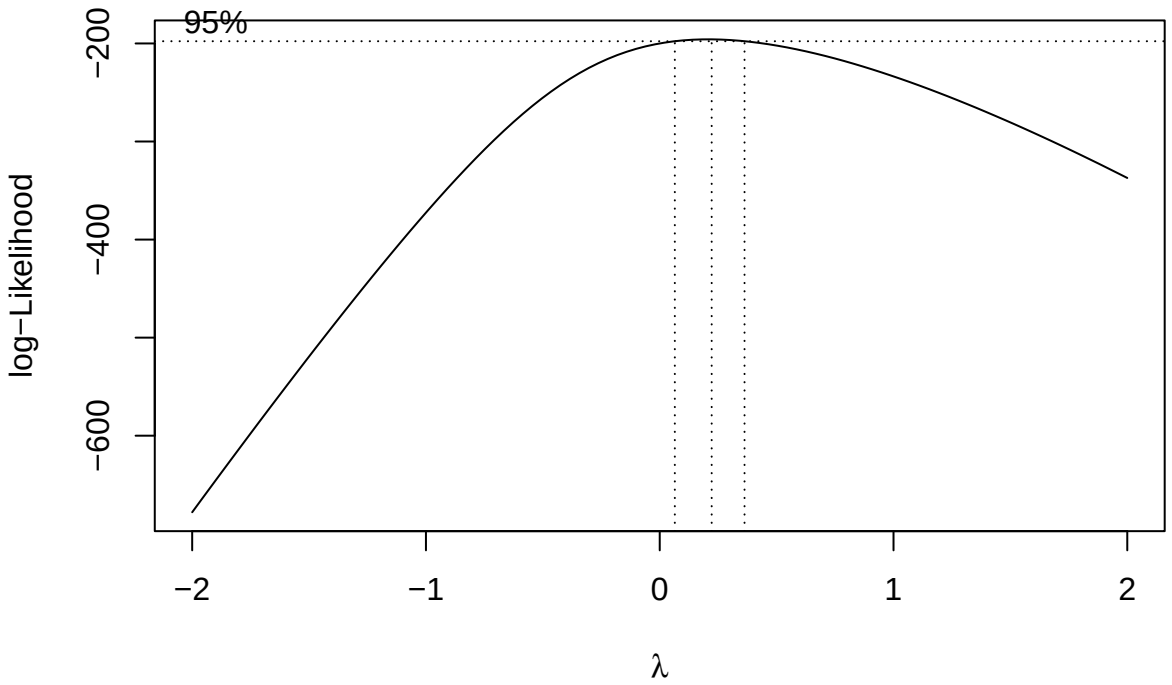
If simplicity and lower risk of overfitting are preferred, the quadratic model is a better choice. If capturing the finer details in the data is crucial despite the complexity, the quartic model may be favored. Given the visual evidence, the quadratic model seems adequate for representing the overall trend while avoiding overfitting.

Exercise 7: Box-Cox Transforming Ozone [5 points]

Fit a linear model predicting Ozone from the Temp variable (first order term only).

Then, using this linear model, generate the plot that includes the confidence interval for the Box-Cox transformation. Based on your visual inspection of this plot, what are reasonable options for λ? Then, specify the optimal λ that you would select along with its corresponding transformation.

```
# Fit a linear model
linear_model <- lm(Ozone ~ Temp, data = airquality)

# Generate the Box-Cox plot
boxcox_plot <- boxcox(linear_model)
```



Answer:

- The peak of the curve suggests the optimal λ value, which maximizes the log-likelihood. The peak appears to be very close to $\lambda = 0$, indicating a logarithmic transformation might be optimal.
- The 95% confidence interval for λ, indicated by the dotted lines, spans a range including negative values, zero, and positive values up to approximately 0.5.

Reasonable options for λ based on this plot would include values from the lower confidence limit to the upper limit around 0.5. However, given the peak's proximity to zero and considering interpretability and simplicity, the optimal λ to select would be 0, corresponding to a logarithmic transformation of the Ozone variable. This transformation can help stabilize variance and normalize the distribution of residuals in the linear model.