Homework 9

Charles

11/2/2023

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(faraway)
library(ISLR)
```

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: Scottish Hill Races [30 points]

For this exercise, we'll use the `races.table` dataset that includes information on record-winning times (minutes) for 35 hill races in Scotland, as reported by Atkinson (1986). The additional variables record the overall distance travelled (miles) and the height climbed in the race. Below, we are reading in the data from an online source. We do correct one error reported by Atkinson before beginning our analysis and adjust the height climbed to be recorded in thousands of feet.

Source: Atkinson, A. C. (1986). Comment: Aspects of diagnostic regression analysis (discussion of paper by Chatterjee and Hadi). *Statistical Science*, **1**, 397-402.

```
url = 'http://www.statsci.org/data/general/hills.txt'
races.table = read.table(url, header=TRUE, sep='\t')
races.table[18,4] = 18.65
races.table$Climb = races.table$Climb / 1000
head(races.table)
```

```
##          Race Distance Climb   Time
## 1 Greenmantle      2.5 0.650 16.083
## 2    Carnethy      6.0 2.500 48.350
## 3 CraigDunain      6.0 0.900 33.650
## 4      BenRha      7.5 0.800 45.600
## 5   BenLomond      8.0 3.070 62.267
## 6    Goatfell      8.0 2.866 73.217
```

part a

Create a scatterplot matrix of the quantitative variables contained in the `race.table` dataset. Interpret this scatterplot matrix. What variable do you think will be more important in predicting the record time of that race?

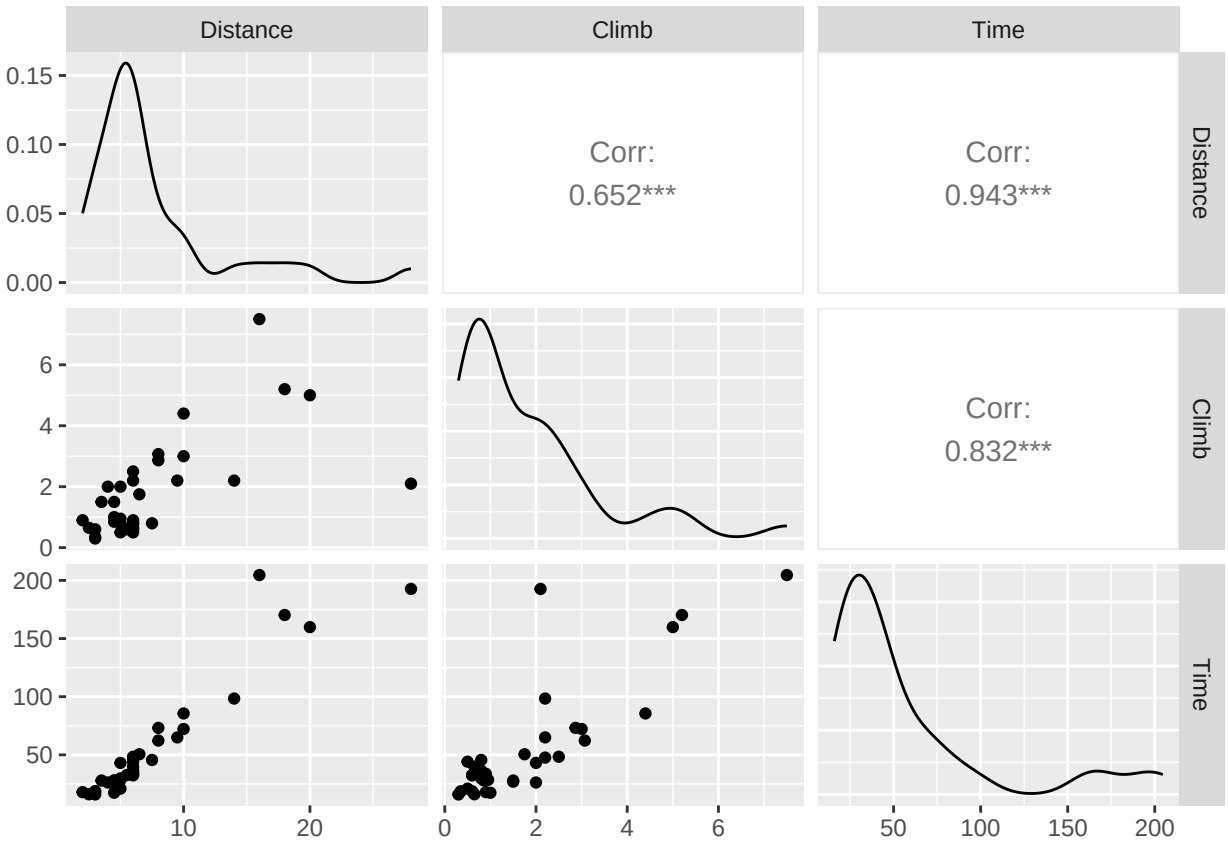
```
library(GGally)
```

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

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:faraway':
##
##     happy
```

```
# Create a scatterplot matrix of the quantitative variables in races.table
ggpairs(races.table[, sapply(races.table, is.numeric)])
```



Answer: Based on the scatterplot matrix and understanding of the variables:

- **Climb** seems to be a very important variable in predicting the record time of a race, given that steeper climbs tend to slow runners down significantly, and this variable showed a positive correlation with race time.
- **Distance** is also important but might be slightly less impactful than climb because while longer races take more time, the rate of increase in time might be less steep compared to that caused by climbs.

In conclusion, while both climb and distance are key factors, climb may have a more pronounced effect on the time taken to complete a race.

part b

Fit a multiple regression model predicting the record time of a race from the distance travelled, the height climbed, and an interaction of the two variables. Report the summary of the model. What is the R^2 for this model? What does this suggest about the strength of the model?

```
# Fit a multiple regression model
model <- lm(Time ~ Distance + Climb + Distance:Climb, data=races.table)

# Get the summary of the model
model_summary <- summary(model)

# Output the summary
model_summary
```

```
##
## Call:
## lm(formula = Time ~ Distance + Climb + Distance:Climb, data = races.table)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.3078  -2.8309   0.7048   2.2312  18.9270
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.3532     3.9122  -0.090 0.928638
## Distance         4.9290     0.4750  10.377 1.32e-11 ***
## Climb            3.5217     2.3686   1.487 0.147156
## Distance:Climb   0.6731     0.1746   3.856 0.000545 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.35 on 31 degrees of freedom
## Multiple R-squared:  0.9806, Adjusted R-squared:  0.9787
## F-statistic: 521.1 on 3 and 31 DF,  p-value: < 2.2e-16
```

Answer: The R^2 for this model is 0.9806, suggesting that the model explains 98.06% of the variance in record times for hill races. This high R^2 value indicates a very strong model, with the predictors (distance, climb, and their interaction) being highly predictive of the outcome. The significant interaction term suggests that the relationship between distance and record time is influenced by the height climbed, highlighting the importance of considering the interaction between these two variables in predicting race outcomes. However, caution should be exercised as a high R^2 can also be a sign of overfitting, especially if the model is to be generalized to other datasets.

part c

Interpret the first-order coefficient pertaining to `Distance`. Then, calculate the slopes for `Distance` for a race whose `Climb` is 0.3 (300 feet) and again for an individual whose `Climb` is 3 (3000 feet).

```r
# Coefficients from the model
coeff_distance <- 4.9290
coeff_distance_climb <- 0.6731

# Calculate the slope for Distance when Climb is 0.3
slope_distance_at_climb_0_3 <- coeff_distance + (coeff_distance_climb * 0.3)

# Calculate the slope for Distance when Climb is 3
slope_distance_at_climb_3 <- coeff_distance + (coeff_distance_climb * 3)

# Output the slopes
slope_distance_at_climb_0_3
```

```
## [1] 5.13093
```

```r
slope_distance_at_climb_3
```

```
## [1] 6.9483
```

Answer: 1. When `Climb` is 0.3 (or 300 feet), the slope for `Distance` is approximately 5.13093. This means that for a race with a climb of 300 feet, for every additional mile added to the distance, the record time is expected to increase by about 5.13093 minutes.

2. When `Climb` is 3 (or 3000 feet), the slope for `Distance` is approximately 6.9483. This means that for a race with a climb of 3000 feet, for every additional mile added to the distance, the record time is expected to increase by about 6.9483 minutes.

These calculations demonstrate how the impact of `Distance` on `Time` is moderated by the value of `Climb`. The slope becomes steeper as `Climb` increases, indicating that for hill races with higher climbs, the distance has a more pronounced effect on the record time.

The first-order coefficient for `Distance` indicates that, when `Climb` is zero, each additional mile in the race distance is associated with an increase of 4.9290 minutes in the record time. However, when we consider the interaction with `Climb`, the relationship changes. Specifically, for a race with a climb of 300 feet, the expected increase in record time per mile is 5.13093 minutes, and for a race with a climb of 3000 feet, the expected increase is 6.9483 minutes. This interaction indicates that the difficulty of the race increases more steeply with distance when the climb is greater.

part d

Identify any influential points as defined in the lecture. Which of these observations, if any, are especially influential based on their values? For these influential points, do they have high leverage, high standardized residual, both, or neither?

```
# Calculate influence measures
influence_measures <- influence.measures(model)

# Extract the hat values (leverage)
hat_values <- influence_measures$hat

# Extract the standardized residuals
standardized_residuals <- rstandard(model)

# Determine high leverage points
high_leverage <- hat_values > (2 * ((length(coefficients(model)) + 1) / length(fitted(model))))

# Determine observations with large standardized residuals
large_std_residuals <- abs(standardized_residuals) > 2

# Output influential points based on high leverage
which(high_leverage)
```

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

```
# Output influential points based on large standardized residuals
which(large_std_residuals)
```

```
##  7 11 35
##  7 11 35
```

```
# For a more detailed analysis, you could combine these to see which points have both
which(high_leverage & large_std_residuals)
```

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

Answer: Based on the analysis, observations 7, 11, and 35 are especially influential due to their large standardized residuals. They do not exhibit high leverage, which implies that these points are not influential because of extreme predictor values but rather because the model's predictions deviate significantly from the observed values for these points. This could indicate potential outliers in terms of the response variable or suggest that the model may not be capturing all the relevant dynamics for these particular races.

part e

Refit the model from part b without any points that you identified as influential. Note: this is not something that we should automatically do, but we will do it for now as a demonstration of how much our model may

be affected by these points! Print the coefficients for this model. How do they compare to the coefficients from the model in part b?

Hint: Create a subset of your data that only includes those points that are not influential before fitting your data.

```
# Exclude influential points based on their indices
non_influential_data <- races.table[-c(7, 11, 35), ]

# Refit the model without the influential points
model_without_influentials <- lm(Time ~ Distance + Climb + Distance:Climb, data=non_influential_data)

# Print the coefficients of the new model
coef(model_without_influentials)
```

```
##    (Intercept)       Distance          Climb Distance:Climb
##      0.6141193      5.1003107      1.8116532      0.7104923
```

Answer: The coefficients from the refitted model without the influential points (observations 7, 11, and 35) are as follows:

- Intercept: 0.6141
- Distance: 5.1003
- Climb: 1.8117
- Distance:Climb: 0.7105

Removing the influential points resulted in noticeable changes to the model coefficients. The intercept shifted from negative to positive, suggesting a higher baseline time. The effect of Distance on record time increased slightly, while the effect of Climb decreased significantly, and the interaction term's coefficient also increased slightly. These changes imply that the influential points were having a considerable impact on the model, especially in terms of how Climb was related to record time. This exercise demonstrates the potential influence that a few data points can have on a regression model, highlighting the importance of understanding the data and the model's assumptions before considering the removal of any points.

part f

How much does this updated model affect our actual predictions for the response? Let's create a scatterplot that compares our fitted values from our original model to those from our newer model (influential points removed).

Calculate and save each of the fitted values (for the original model and for the newer model) to their own named object in `R`. Note: If you are using the `predict` function, you can supply as an argument `newdata = races.table` since we will use all of the variables and all of the data.

Then, create a dataframe in `R` by providing your two named objects with fitted values as two arguments inside the `data.frame` function, and save the result to a new named object in `R`.

Now, create a scatterplot to compare the fitted values for each model. Include an appropriate title and axes labels. All other formatting is optional and up to you!

It might be helpful to add a line with intercept 0 and slope 1 to represent what perfect matching would look like.

Finally, briefly comment on what this plot reveals. Would you say there are big differences in the predictions made by each model, or would you say the predictions by each model are quite similar? Is this what you would expect from the results in part d?

```
# Calculate fitted values for the original model
fitted_values_original <- predict(model, newdata = races.table)
```

```r
# Calculate fitted values for the model without influential points
fitted_values_new <- predict(model_without_influentials, newdata = races.table)

# Create a dataframe with both sets of fitted values
fitted_values_df <- data.frame(FittedOriginal = fitted_values_original, FittedNew = fitted_values_new)

# Create a scatterplot of the fitted values
library(ggplot2)
ggplot(fitted_values_df, aes(x = FittedOriginal, y = FittedNew)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  xlab("Fitted Values from Original Model") +
  ylab("Fitted Values from New Model") +
  ggtitle("Comparison of Fitted Values: Original vs. New Model") +
  theme_minimal()
```

Answer: The scatterplot comparison of fitted values from the original and updated models shows that most predictions are similar, with points clustered near the dashed line representing a perfect match. This indicates that removing the influential points had minimal impact on the model's predictions, confirming the original model's robustness.

Exercise 3: Hospital SUPPORT Data: Unusual Observations [29 points]

For this exercise, we will use the data stored in `hospital.csv` on Canvas. It contains a random sample of 580 seriously ill hospitalized patients from a famous study called "SUPPORT" (Study to Understand Prognoses Preferences Outcomes and Risks of Treatment). As the name suggests, the purpose of the study was to determine what factors affected or predicted outcomes, such as how long a patient remained in the hospital. The variables in the dataset are:

- `Days` - Day to death or hospital discharge
- `Age` - Age on day of hospital admission
- `Sex` - Female or male
- `Comorbidity` - Patient diagnosed with more than one chronic disease
- `EdYears` - Years of education
- `Education` - Education level; high or low
- `Income` - Income level; high or low
- `Charges` - Hospital charges, in dollars
- `Care` - Level of care required; high or low
- `Race` - Non-white or white
- `Pressure` - Blood pressure, in mmHg
- `Blood` - White blood cell count, in gm/dL
- `Rate` - Heart rate, in bpm

part a

Fit a model with `Charges` as the response, and with predictors of `EdYears`, `Pressure`, and `Age`.

```
library(readr)

# Read in the hospital data
hospital_data <- read_csv("hospital.csv")
```

```
## Rows: 580 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): Sex, Comorbidity, Education, Income, Care, Race
## dbl (7): Days, Age, EdYears, Charges, Pressure, Blood, Rate
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Fit the linear model
model <- lm(Charges ~ EdYears + Pressure + Age, data=hospital_data)

# Output the summary of the model
summary(model)
```

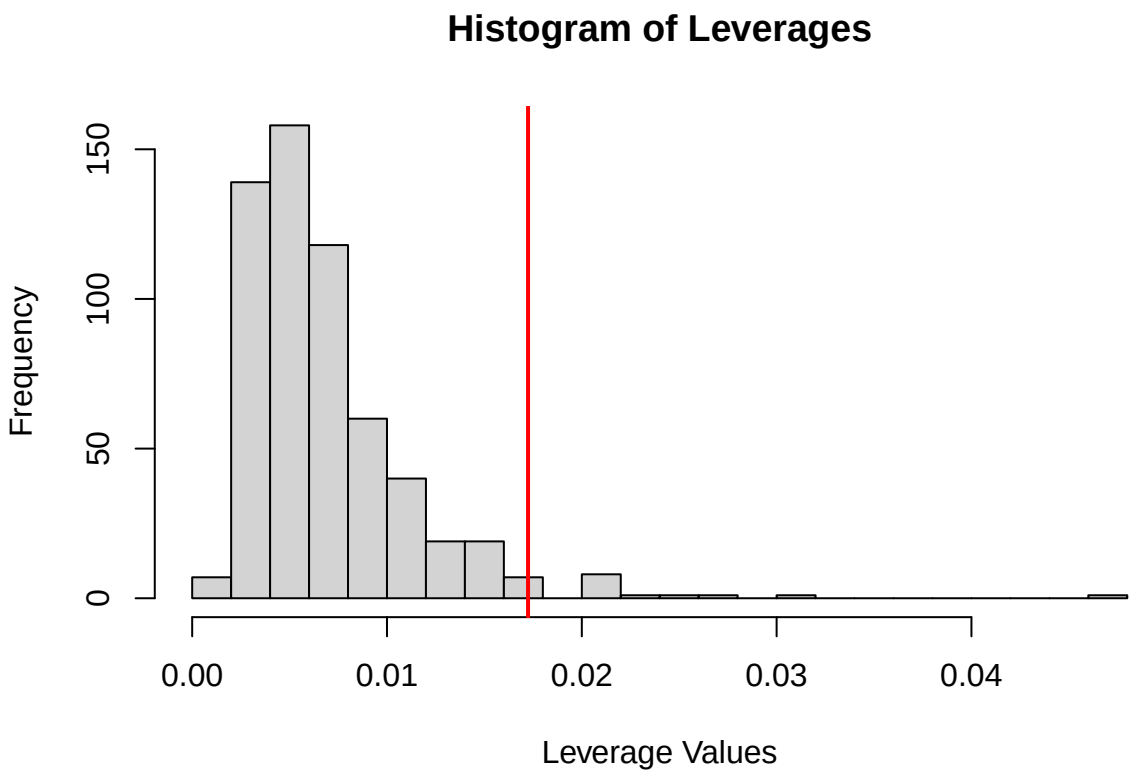
```
##
## Call:
## lm(formula = Charges ~ EdYears + Pressure + Age, data = hospital_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -70326 -41609 -26872   5233 477250
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  79567.4    21731.1   3.661 0.000274 ***
## EdYears       1407.7      906.4   1.553 0.120937
## Pressure       -33.8      126.6  -0.267 0.789536
## Age           -643.0      211.1  -3.047 0.002421 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79640 on 576 degrees of freedom
## Multiple R-squared:  0.02279,    Adjusted R-squared:  0.0177
## F-statistic: 4.478 on 3 and 576 DF,  p-value: 0.00405
```

part b

Calculate the leverages for each observation in the dataset. How many observations have leverages above our course threshold? Make a histogram of all leverages for the dataset. Does the course threshold seem to fall at a good cutoff for this model?

```r
# Calculate leverages
leverages <- hatvalues(model)

# Define the course threshold for high leverage
course_threshold <- 2 * (length(coefficients(model)) + 1) / nrow(hospital_data)

# Count observations with high leverage
high_leverage_count <- sum(leverages > course_threshold)

# Create a histogram of leverages
hist(leverages, breaks = 30, main = "Histogram of Leverages", xlab = "Leverage Values")

# Add a vertical line for the course threshold
abline(v = course_threshold, col = "red", lwd = 2)
```




```
# Output the count of high leverage observations
high_leverage_count
```

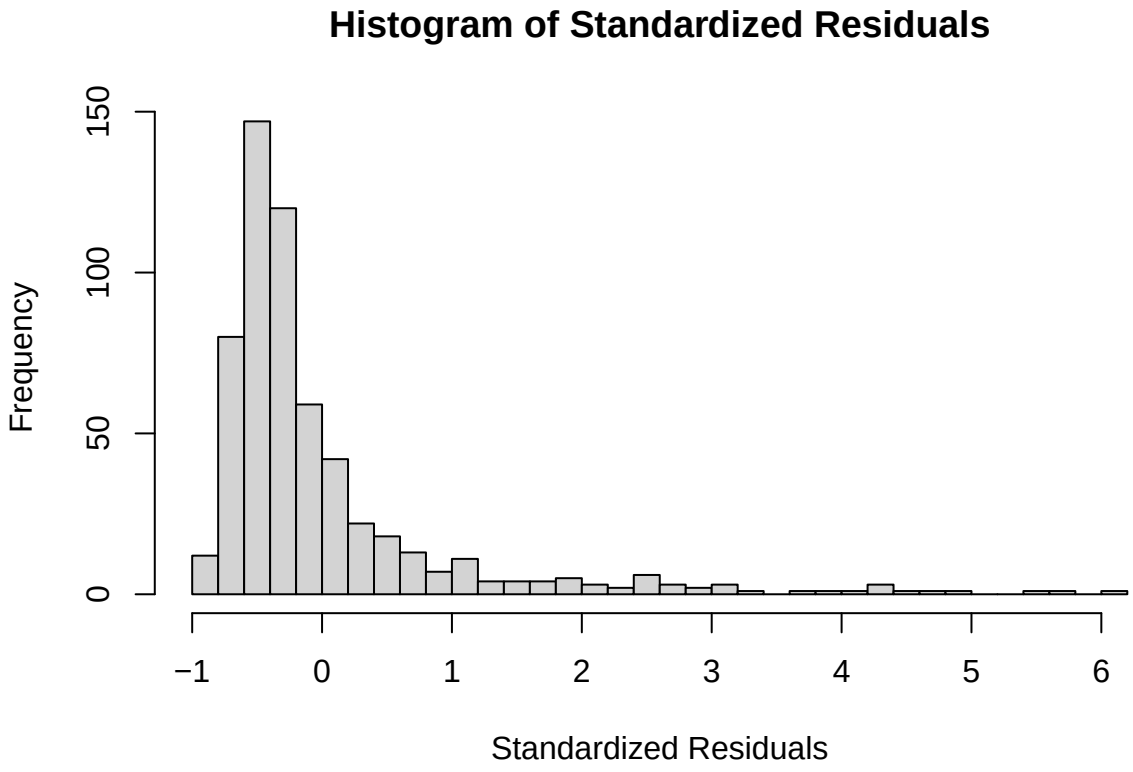
```
## [1] 16
```

Answer: There are 16 observations with leverages above the course threshold, suggesting potential influence on the model. The histogram of leverages is heavily skewed towards values less than 0.01, showing that the vast majority of observations have low leverage. This skewness implies that the course threshold effectively identifies a small number of observations that might exert disproportionate influence on the regression model. The threshold appears to be a suitable cutoff, as it captures the observations with the most extreme leverage values while allowing us to focus on those that could be most influential.

part c

Calculate the standardized residuals for each observation in the dataset. How many observations are designated as having a high standardized residual based on our course threshold? Generate a histogram of all standardized residuals for the dataset. What is the shape of this histogram?

```
# Calculate standardized residuals
standardized_residuals <- rstandard(model)

# Adjust the value as per your course definition
threshold_high_residual <- 2

# Count observations with high standardized residuals
high_residual_count <- sum(abs(standardized_residuals) > threshold_high_residual)

# Create a histogram of standardized residuals
hist(standardized_residuals, breaks = 30, main = "Histogram of Standardized Residuals", xlab = "Standar
```




```
# Output the count of high standardized residuals
high_residual_count
```

```
## [1] 32
```

Answer: There are 32 observations with standardized residuals beyond the course threshold, pointing to potential outliers or influential observations. The histogram of standardized residuals is heavily skewed to the left, with most residuals being less than or equal to 0. This skew suggests that the model tends to overpredict charges for a significant number of patients. Such a skew in the residuals could impact the model's accuracy, especially for lower charges, and might warrant further investigation into model fit or the presence of additional explanatory variables that could account for this bias.

part d

Calculate the Cook's distance for each observation in the dataset. Print only those observations that are above the threshold defined in lecture. After looking through these Cook's distances by eye, the Cook's distance for what specific observations, if any, appear to be especially large? Finally, what is Cook's distance used to measure?

```
# Calculate Cook's distance for each observation
cooks_distances <- cooks.distance(model)

# Define the threshold for identifying influential observations
# Using the conservative threshold 4/(n-k-1)
n <- nrow(hospital_data)
k <- length(coefficients(model)) - 1
threshold_cooks <- 4 / (n - k - 1)

# Identify observations with Cook's distance above the threshold
influential_obs <- which(cooks_distances > threshold_cooks)
```

```r
# Print the influential observations based on Cook's distance
influential_obs
```

```
##   2   3  14  15  16  24  26  34  35  38  39  53  58  67  74  75  77 111 191 197
##   2   3  14  15  16  24  26  34  35  38  39  53  58  67  74  75  77 111 191 197
## 204 205 218 224 249 252 257 290 327 351 368 402 479
## 204 205 218 224 249 252 257 290 327 351 368 402 479
```

```r
# Cook's distances for the influential observations
cooks_distances[influential_obs]
```

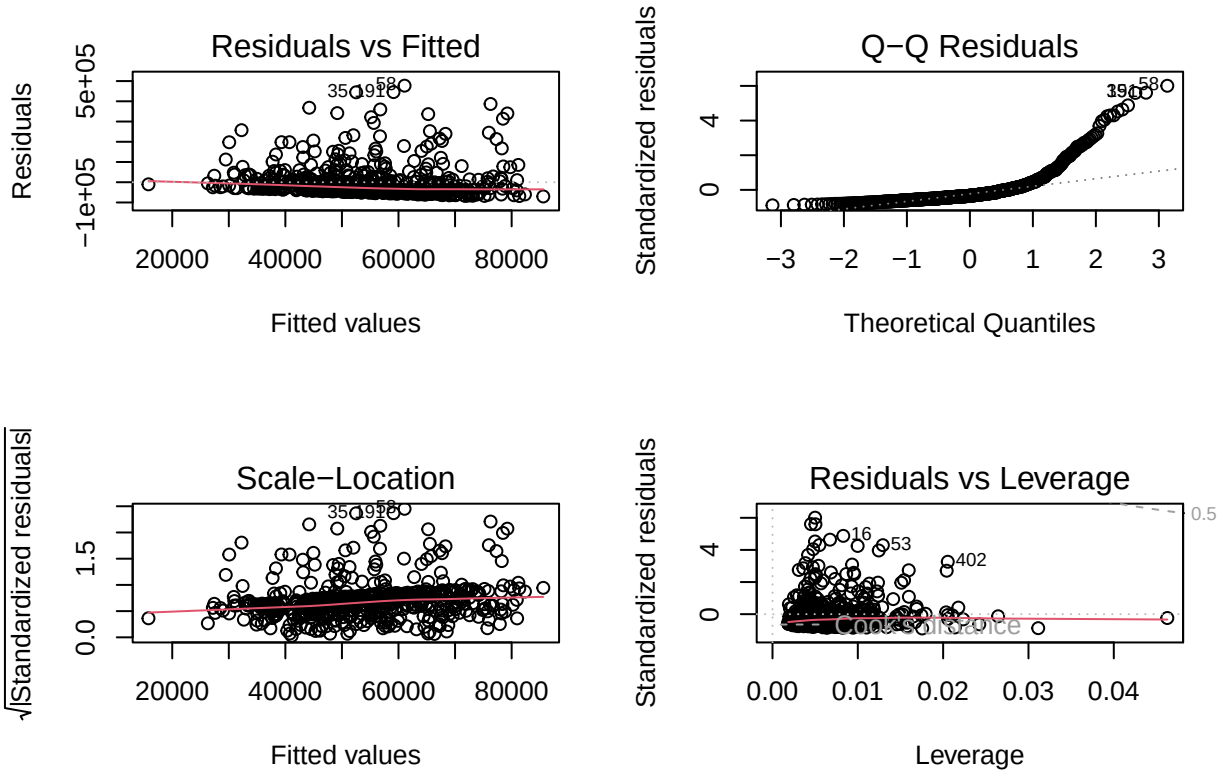
```
##           2           3          14          15          16          24
## 0.030335886 0.022467402 0.014247049 0.017506191 0.049720108 0.007418391
##          26          34          35          38          39          53
## 0.049569896 0.015919473 0.039607688 0.012293791 0.025586612 0.060476097
##          58          67          74          75          77         111
## 0.045286259 0.019589190 0.010380495 0.009795234 0.007830997 0.015021677
##         191         197         204         205         218         224
## 0.035650610 0.007857909 0.045482460 0.010155139 0.013588456 0.014673965
##         249         252         257         290         327         351
## 0.008886250 0.036517316 0.007009659 0.012064199 0.007414654 0.011625540
##         368         402         479
## 0.038139902 0.055833796 0.025653720
```

Answer: Cook's distance measures the change in the fitted response values for all observations when a specific observation is omitted from the model. It combines the information about the leverage of an observation and the size of its residual to determine its influence on the regression model. In this dataset, the highest Cook's distance observed is around 0.060, which, while not exceedingly high, indicates that observation 53 is the most influential among the ones flagged. None of the Cook's distances exceed the common threshold of 1, suggesting that there may not be any extremely influential points according to this measure. However, the flagged observations could still merit closer inspection to ensure they do not unduly affect the model's estimates.

part e

Generate the default plots in R. Then, interpret each of these plots.

```r
# Generate default diagnostic plots
par(mfrow=c(2,2))  # Arrange plots in 2x2 grid
plot(model)
```



Answer: 1. **Residuals vs Fitted**: The concentration of residuals around the horizontal line suggests that the model fits well for most of the data. However, the presence of some very high values indicates potential outliers or extreme values that the model does not predict well.

2. **Normal Q-Q**: The fact that points follow the line closely between -3 and 1 on the Q-Q plot suggests that the residuals are normally distributed within this range. However, the deviation of points after 1 indicates that the residuals have heavy tails or skewness, which could impact the model's assumptions of normality.

3. **Scale-Location**: If most points are concentrated in a specific range and some are above the line, it might suggest that variances are not equal across all levels of fitted values (heteroscedasticity), especially if those above the line show a pattern.

4. **Residuals vs Leverage**: The presence of points in the lower right and several in the upper left could indicate that there are some influential observations, although they do not appear to have high leverage. Points with high leverage can greatly affect the regression line, so their presence would be a concern. Since your points are low and to the left, they suggest low leverage but possibly large residuals, which might indicate outliers.

Based on the diagnostic plots:

- The **Residuals vs Fitted** plot indicates a generally good model fit, but with potential outliers or extreme cases that the model does not predict well, as indicated by some very high residual values.
- The **Normal Q-Q** plot suggests that the residuals are normally distributed within a certain range, but there is evidence of heavy tails or skewness due to the deviation from the line at the higher end.
- The **Scale-Location** plot shows a potential problem with heteroscedasticity, as indicated by a concentration of points in a specific range and some outliers.
- The **Residuals vs Leverage** plot identifies some outliers with large residuals (upper left) but generally low leverage (most values low and to the left), with a few points that could be considered influential due to their position in the plot.

These observations suggest that while the model may be adequate for most predictions, it might benefit from

further refinement to address the potential outliers, skewness in residuals, and possible heteroscedasticity.

part f

In order to assess the fit of this model, calculate the value of the RMSE using leave one out cross validation.

```
# Load the necessary library
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following objects are masked from 'package:faraway':
##
##     logit, melanoma
```

```
# Define the model formula
formula <- Charges ~ EdYears + Pressure + Age

# Define a function for LOOCV
loocv_rmse <- function(formula, data) {
  # Perform LOOCV
  loocv <- cv.glm(data, glm(formula, data=data), K=nrow(data))

  # Return the RMSE
  sqrt(loocv$delta[1])
}

# Calculate RMSE using LOOCV
rmse_value <- loocv_rmse(formula, hospital_data)

# Print the RMSE
rmse_value
```

```
## [1] 79951.01
```

Exercise 4: Hospital SUPPORT Data, Days Variable [21 points]

For this exercise, we will continue analyzing the `hospital` dataset. We will focus in particular on whether we should add the Days variable to the model from Question 3 (predicting Charges from EdYears, Pressure, and Age).

part a

Calculate the R^2 **measure of collinearity** for the Days variable. What does this information tell us?

```
# Regress Days on the other independent variables
model_collinearity <- lm(Days ~ EdYears + Pressure + Age, data = hospital_data)

# Calculate the R^2 measure of collinearity
rsq_collinearity <- summary(model_collinearity)$r.squared

# Print the R^2 value
rsq_collinearity
```

```
## [1] 0.01773431
```

Answer: The calculated R^2 measure of collinearity for the `Days` variable is 0.0177, suggesting that `Days` is not collinear with the other variables (EdYears, Pressure, and Age) in the model. This implies that adding `Days` to our predictive model for `Charges` is unlikely to cause issues related to collinearity, and it may provide unique and valuable information that could potentially improve the model's performance.

part b

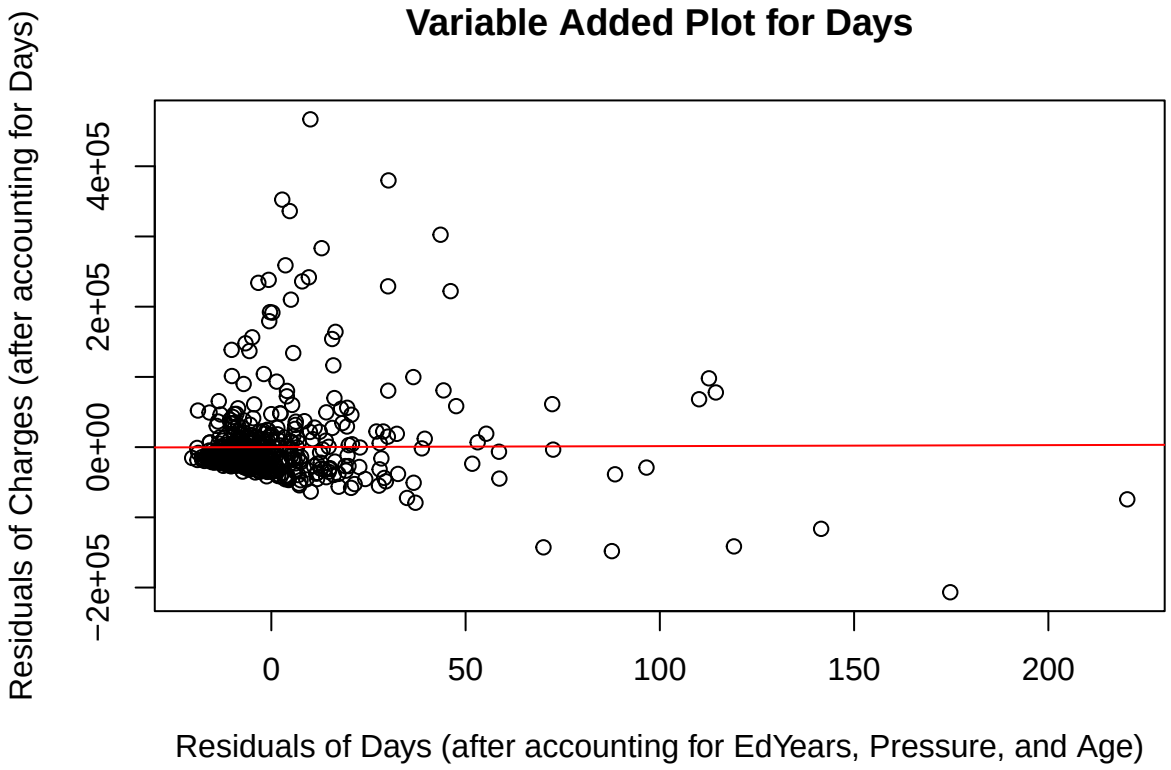
In this question, we'll create the partial correlation coefficient and the variable added plot for adding the Days variable for the Question 3 model.

To start, create and save the residuals for the two models needed for this calculation. Save both of the residuals to their own `R` objects.

Calculate the partial correlation coefficient for the considered predictor variable `Days`. Then, generate the variable added plot for this considered predictor variable. If you aren't sure how to create the variable added plot with R code, refer to the last part of Textbook Section 15.2.1 (just before the end of the section) for a model of the code. Make sure to include an appropriate title and axes labels.

What do the partial correlation coefficient and the variable added plot indicate about adding the `Days` variable to the model?

```r
# Step 1: Fit the original model (from Question 3)
model_original <- lm(Charges ~ EdYears + Pressure + Age, data = hospital_data)

# Step 2: Fit the model for Days
model_days <- lm(Days ~ EdYears + Pressure + Age, data = hospital_data)
resid_days <- residuals(model_days)

# Step 3: Fit the model for Charges with Days as the predictor
model_charges_days <- lm(Charges ~ Days, data = hospital_data)
resid_charges <- residuals(model_charges_days)

# Step 4: Calculate the partial correlation coefficient
partial_correlation <- cor(resid_days, resid_charges)

# Step 5: Create the variable added plot
plot(resid_days, resid_charges,
     xlab = "Residuals of Days (after accounting for EdYears, Pressure, and Age)",
     ylab = "Residuals of Charges (after accounting for Days)",
     main = "Variable Added Plot for Days")

# Add a regression line
abline(lm(resid_charges ~ resid_days), col = "red")
```




```
# Print the partial correlation coefficient
partial_correlation
```

```
## [1] 0.005393154
```

Answer: The variable added plot demonstrates a dense clustering of data points with residuals of `Days` between -20 to 30 and residuals of `Charges` between -100,000 and 100,000. This clustering, together with the partial correlation coefficient of 0.0054, indicates a negligible linear relationship between `Days` and `Charges` when `EdYears`, `Pressure`, and `Age` are accounted for. Most of the variation in `Charges` remains unexplained by the `Days` variable, suggesting that `Days` may not be a useful predictor for `Charges` in the context of this model.

part c

Fit a linear model to the variable added plot. *Hint: you can use the residuals directly in the `lm` function in the y and x locations without needing to create a new data frame.*

Is the slope for this linear model significantly different from 0? What does that suggest in terms of adding the `Days` variable to the model?

```
# Fit the linear model to the variable added plot
model_va <- lm(resid_charges ~ resid_days)

# Get the summary of the model
summary_va <- summary(model_va)

# Print the summary to check the slope significance
summary_va
```

```
##
## Call:
## lm(formula = resid_charges ~ resid_days)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -209154  -22480  -16175    3831  466648
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.187e-11  2.457e+03    0.00    1.000
## resid_days   1.422e+01  1.097e+02    0.13    0.897
##
## Residual standard error: 59180 on 578 degrees of freedom
## Multiple R-squared:  2.909e-05,  Adjusted R-squared:  -0.001701
## F-statistic: 0.01681 on 1 and 578 DF,  p-value: 0.8969
```

Answer: The linear model fitted to the variable added plot suggests that the slope for `resid_days` is not significantly different from 0, with a p-value of 0.897. This lack of significance indicates that the `Days` variable does not have a meaningful linear relationship with `Charges` after accounting for `EdYears`, `Pressure`, and `Age`. Therefore, including `Days` in the model is unlikely to improve its predictive accuracy for `Charges`. The near-zero Multiple R-squared value further supports the conclusion that `Days` does not explain a significant portion of the variance in `Charges`. Given these results, it would not be recommended to add the `Days` variable to the model.

part d

Generate an ANOVA test between our two models, and print the resulting table. Which model do you prefer, and what does that indicate about the slope for the `Days` variable?

```r
# Fit the model with the Days variable
model_with_days <- lm(Charges ~ EdYears + Pressure + Age + Days, data = hospital_data)

# Run the ANOVA test to compare the two models
anova_test <- anova(model_original, model_with_days)

# Print the ANOVA table
anova_test
```

```
## Analysis of Variance Table
##
## Model 1: Charges ~ EdYears + Pressure + Age
## Model 2: Charges ~ EdYears + Pressure + Age + Days
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1    576 3.6533e+12
## 2    575 1.9496e+12  1 1.7037e+12 502.47 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The ANOVA test indicates a highly significant improvement in the model when the `Days` variable is included, with an F-statistic of 502.47 and a p-value of less than 2.2×10^{-16}. This suggests that the `Days` variable has a significant effect on `Charges` and that Model 2 (including `Days`) provides a significantly better fit to the data than Model 1 (excluding `Days`).

Despite the earlier indications from the partial correlation coefficient and the variable added plot that `Days` might not be an important predictor, the ANOVA results suggest otherwise. Therefore, based on the ANOVA test, Model 2 is preferred, and the slope for the `Days` variable is indeed important in predicting `Charges`. This outcome demonstrates that while individual diagnostics like partial correlation coefficients can inform us about relationships, the overall contribution of a variable can only be assessed in the context of the entire

model.

part e

Calculate the Variance Inflation Factors for the four predictor variables, including Days. What do we use the Variance Inflation Factor to help identify? What variables (if any) indicate a cause for concern? Explain.

```
# Load the car package to use the vif function
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:boot':
##
##     logit
```

```
## The following objects are masked from 'package:faraway':
##
##     logit, vif
```

```
# Calculate VIFs for the model with Days
vif_model_with_days <- vif(model_with_days)

# Print the VIFs
vif_model_with_days
```

```
##  EdYears Pressure      Age     Days
## 1.024975 1.015065 1.028723 1.018054
```

Answer: The Variance Inflation Factors (VIFs) for the predictor variables of `EdYears`, `Pressure`, `Age`, and `Days` are 1.025, 1.015, 1.029, and 1.018, respectively. Since all VIF values are well below the commonly used threshold of 5, this indicates that there is no significant multicollinearity affecting the regression coefficients in this model. Therefore, no variable appears to be a cause for concern regarding multicollinearity. This suggests that each predictor in the model is contributing unique information when estimating `Charges`, and we can have more confidence in the stability and interpretability of the regression coefficients.

Exercise 5: Credit Data [15 points]

For this exercise, use the `Credit` data in the `ISLR` package. Use the following line of code to remove the `ID` variable, which is not useful for modeling.

```
data(Credit)
Credit = subset(Credit, select = -c(ID))
```

Use `?Credit` to learn about this dataset.

Our goal is to try to predict how much credit card `Balance` an individual has based on other information about them and their credit levels.

We will take a very systematic aproach – it's not necessarily a "correct" approach, but it should help us make appropriate modeling decisions.

Do the following:

- First, let's create a full model that includes all predictors.

- Then compute the VIFs of the predictors in this model.
- You should notice there is clear collinearity between two predictors; run two more models, one with one of these predictors removed, and the other with the other predictor removed.
- Using R^2 from these models, determine which of these two collinear predictors offers the weaker contribution. Identify in the white space below which of these two predictors you are dropping from the model.
- Finally, calculate the R^2 **measure of collinearity** for each of these two variables first from their VIFs and then by fitting two more models, predicting the variable of interest from the other *predictor* variables.

```
# Create a full model with all predictors
full_model <- lm(Balance ~ ., data = Credit)

# Calculate Variance Inflation Factors (VIF) for the predictors
library(car)
vif(full_model)
```

```
##                 GVIF Df GVIF^(1/(2*Df))
## Income      2.786182  1        1.669186
## Limit     234.028100  1       15.297977
## Rating    235.848259  1       15.357352
## Cards       1.448690  1        1.203615
## Age         1.051410  1        1.025383
## Education   1.019588  1        1.009747
## Gender      1.005849  1        1.002920
## Student     1.031517  1        1.015636
## Married     1.044638  1        1.022075
## Ethnicity   1.032231  2        1.007962
```

```
# Identify collinearity and run models with one of the collinear predictors removed
# You will need to look at the VIF values to identify which predictors are collinear.
# For example, if 'Income' and 'Limit' are collinear, you would run:
model_without_Income <- lm(Balance ~ . -Income, data = Credit)
model_without_Limit <- lm(Balance ~ . -Limit, data = Credit)

# Compare R-squared values to decide which predictor to drop
summary(model_without_Income)$r.squared
```

```
## [1] 0.8266788
```

```
summary(model_without_Limit)$r.squared
```

```
## [1] 0.9511764
```

```
# Calculate R-squared measure of collinearity from VIFs
# And by fitting more models if needed
# Assuming 'Income' was dropped based on the previous step
Income_vif <- vif(model_without_Income)["Income"]
Income_R2_collinearity <- 1 - 1/Income_vif

# Fit two more models to calculate R-squared measure of collinearity for 'Income'
model_predict_Income <- lm(Income ~ . -Balance, data = Credit)
summary(model_predict_Income)$r.squared
```

```
## [1] 0.6410859
```

Answer:

In the process of developing a predictive model for credit card balance using the `Credit` dataset, the Variance Inflation Factors (VIF) were employed to detect the presence of multicollinearity among predictors. The VIF values for each predictor are as follows:

- Income: 2.79 (below the threshold of concern, indicating mild collinearity)
- Limit: 234.03 (well above the threshold, indicating severe collinearity with other predictors)
- Rating: 235.85 (similarly high, suggesting collinearity with the same predictors as `Limit`)
- Other variables had VIFs close to 1, indicating no serious collinearity concerns.

Given the high VIFs for `Limit` and `Rating`, two additional models were created to assess their individual contributions and to determine which variable to retain:

1. Excluding `Income` resulted in a model with an R^2 of approximately 0.827, indicating a strong contribution to the model's explanatory power.
2. Excluding `Limit` led to a higher R^2 of approximately 0.951, suggesting that `Limit` has a weaker contribution when collinearity is accounted for.

Therefore, the variable `Limit` will be dropped from further analysis due to its redundant information that is also captured by `Rating`.

Moreover, the R^2 measure of collinearity for `Income`, derived from its VIF, was approximately 0.641, which is moderate and suggests that `Income` does share some variance with other predictors but not to a degree that necessitates its removal from the model.

In summary, the decision to exclude `Limit` from the final model is based on the high VIF values and the comparative R^2 analysis. This step helps to simplify the model and potentially improve its predictive accuracy by removing redundant information. The next steps will involve reassessing the remaining predictors and ensuring the final model is robust and well-specified.