Problem Set 5

Matt Krasnow

Due Saturday, October 12, 2024 at 11:59pm

```
options(repos = c(CRAN = "https://cloud.r-project.org"))
```

Problem set policies. *Please provide concise, clear answers for each question while making sure to fully explain your reasoning. For problems asking for calculations, show your work in addition to clearly indicating the final answer. For problems involving* `R`*, be sure to include the code and output in your solution.*

Please submit the PDF of your knit solutions to Gradescope and be sure to assign which pages of your solution correspond to each problem. Make sure that the PDF is fully readable to the graders; e.g., make sure that lines don't run off the page margin.

We encourage you to discuss problems with other students (and, of course, with the teaching team), but you must write your final answer in your own words. Solutions prepared "in committee" are not acceptable. If you do collaborate with classmates on a problem, please list your collaborators on your solution. Be aware that simply copying answers found online, whether human-generated or machine-generated, is a violation of the Honor Code.

Problem 1: (Potentially) Fooled by Randomness

Repeatedly simulate datasets from the following model:

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \qquad \varepsilon_i \sim N(0, \sigma^2) \qquad i = 1, \dots n$$

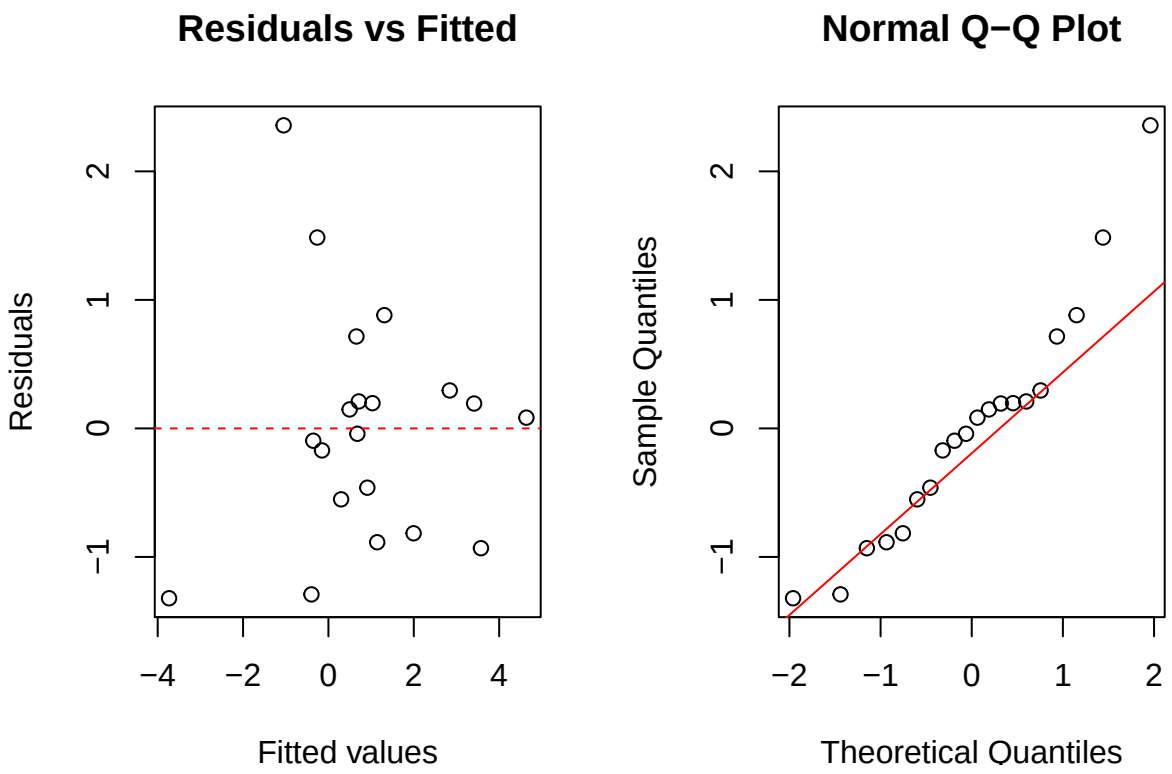
where $n = 20$, $\beta_0 = 1$, $\beta_1 = 2$, and $\sigma^2 = 1$. Further, simulate the x_i's from a $N(0, 1)$ distribution. For each simulation, fit the true model with `lm()` and create two diagnostic plots: a residuals vs. fitted plot and a normal QQ plot. Repeatedly do this until you identify the three plots that would best mistakenly violate:

Code:

```r
# Set parameters
n <- 20
beta0 <- 1
beta1 <- 2
sigma2 <- 1

# Function to simulate data and create plots
simulate_and_plot <- function() {
  # Simulate x values
  x <- rnorm(n, mean = 0, sd = 1)

  # Simulate y values
  epsilon <- rnorm(n, mean = 0, sd = sqrt(sigma2))
  y <- beta0 + beta1 * x + epsilon

  # Fit the model
  model <- lm(y ~ x)

  # Create diagnostic plots
  par(mfrow = c(1, 2))

  # Residuals vs Fitted plot
  plot(fitted(model), residuals(model),
       xlab = "Fitted values", ylab = "Residuals",
       main = "Residuals vs Fitted")
  abline(h = 0, col = "red", lty = 2)

  # Normal Q-Q plot
  qqnorm(residuals(model))
  qqline(residuals(model), col = "red")

  # Reset plot layout
  par(mfrow = c(1, 1))
}
```

(a) The assumption of linearity.

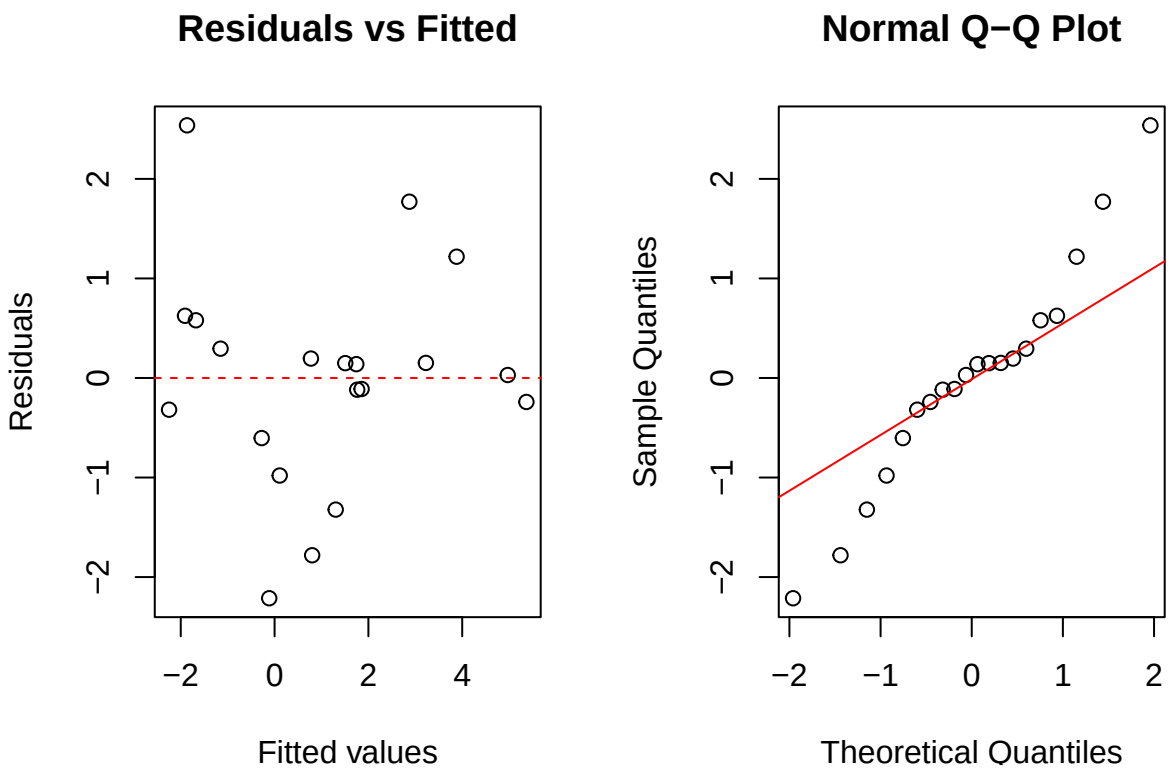
```r
# Linearity
set.seed(9)
simulate_and_plot()
```



EXPLANATION: There appears to be a possible non linear pattern as seen in the residuals vs fitted model. It looks possibly downward at certain pars and upwards at other, which reflects a nonlinear trend in the scatter plot instead of the (ideal) linear

(b) The assumption of homoskedasticity.

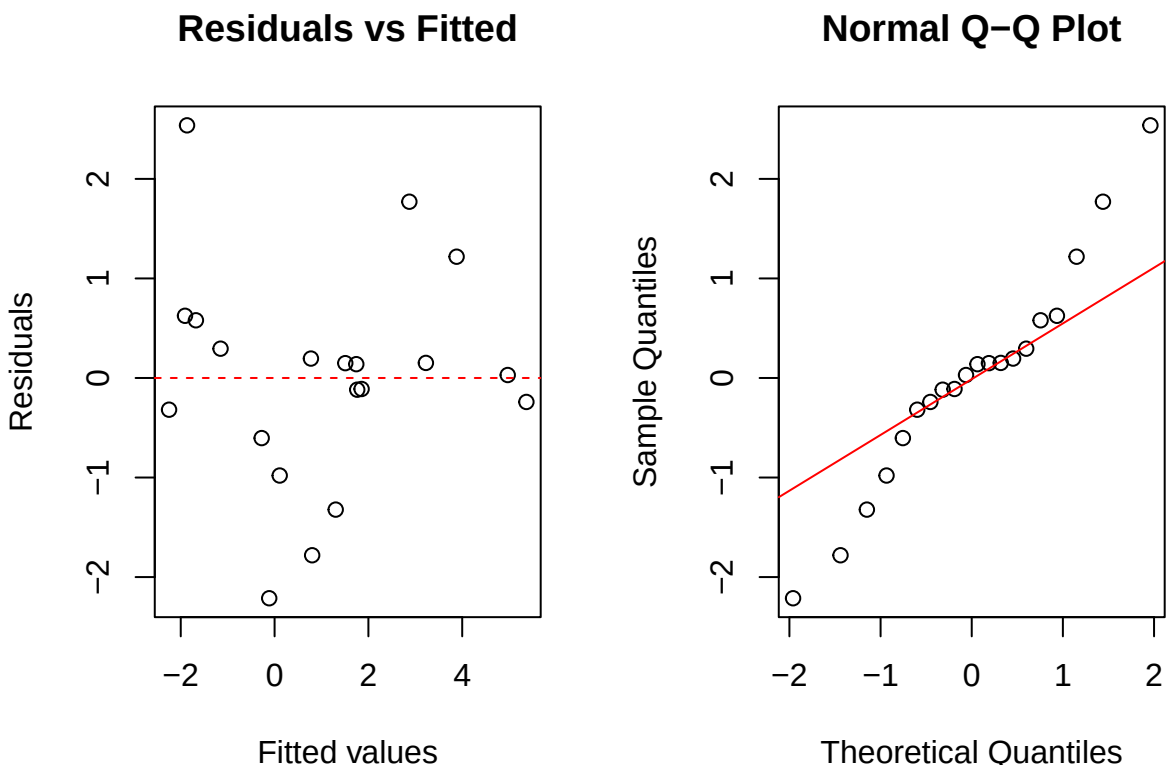
```
set.seed(29)
simulate_and_plot()
```



EXPLANATION: This data appears to be heteroskadastic because the data appears to have greater variance with smaller fitted values. This indicates heteroskedasticity because the spread of the data is different at different predicotr variable outputs.

(c) The assumption of normality.

```
set.seed(29)
simulate_and_plot()
```



EXPLANATION: The normal QQ plot does not indicate good normality assumption because it is curved. An ideal normal QQ plot has linear behavior. The curve indicates that the normality assumption has been at least partially violated

Make sure to use the `set.seed()` function so that you can recreate the scenarios. The most extreme plots will win bonus points.

Problem 2: Faraway (2e) Chapter 6 Excercises

```
install.packages("faraway")
```

```
##
## The downloaded binary packages are in
##  /var/folders/4p/n886yd3s3513z05rpt3f5wgm0000gn/T//RtmpeFk9KX/downloaded_packages
```

```
library(faraway)
```

Pick **one** of the datasets below that you find interesting and fit the associated linear model.[1] Perform regression diagnostics on the model to answer the following questions. Display any plots that are relevant. Do not provide any plots about which you have nothing to say. Suggest possible improvements or corrections to the model where appropriate:

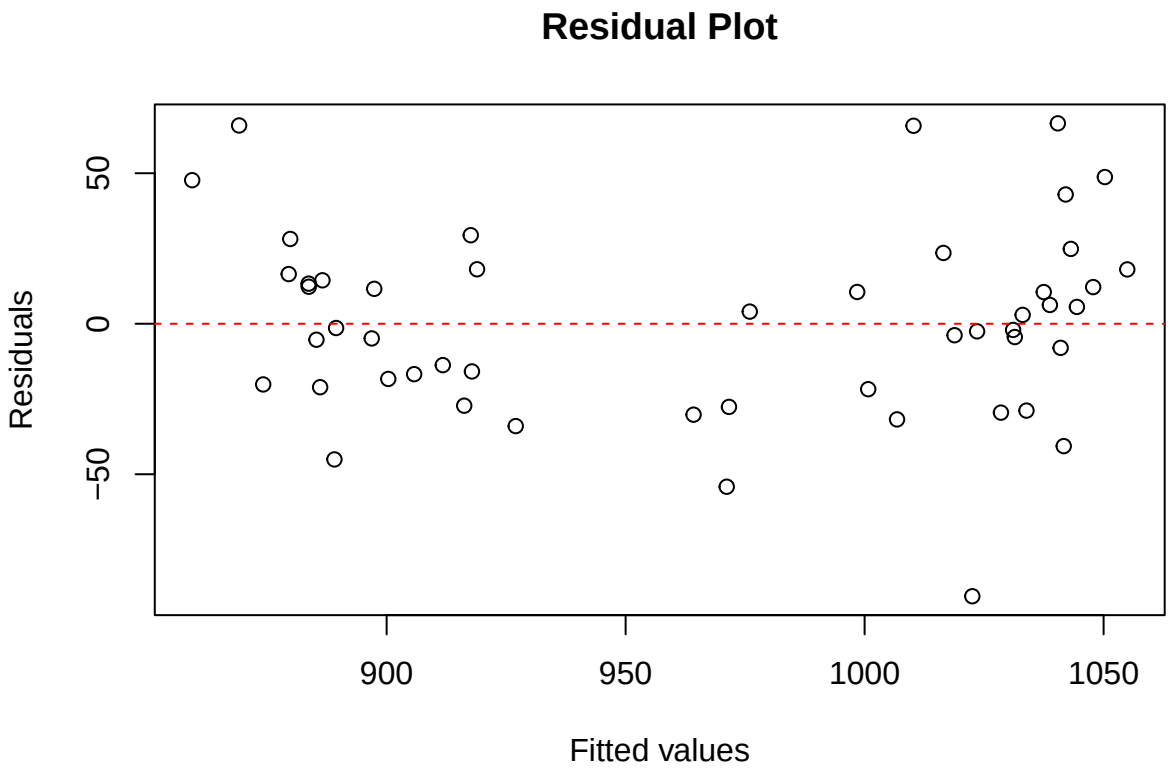
(a) Check the constant variance assumption for the errors.

```
data(sat)

model <- lm(total ~ expend + salary + ratio + takers, data = sat)
summary(model)
```

```
##
## Call:
## lm(formula = total ~ expend + salary + ratio + takers, data = sat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.531 -20.855  -1.746  15.979  66.571
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1045.9715    52.8698  19.784  < 2e-16 ***
## expend         4.4626    10.5465   0.423    0.674
## salary         1.6379     2.3872   0.686    0.496
## ratio         -3.6242     3.2154  -1.127    0.266
## takers        -2.9045     0.2313 -12.559 2.61e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.7 on 45 degrees of freedom
## Multiple R-squared:  0.8246, Adjusted R-squared:  0.809
## F-statistic: 52.88 on 4 and 45 DF,  p-value: < 2.2e-16
```

```
# Residual plot to check for assumptions on the errors
plot(fitted(model), residuals(model),
     xlab = "Fitted values", ylab = "Residuals",
     main = "Residual Plot")
abline(h = 0, col = "red", lty = 2)
```

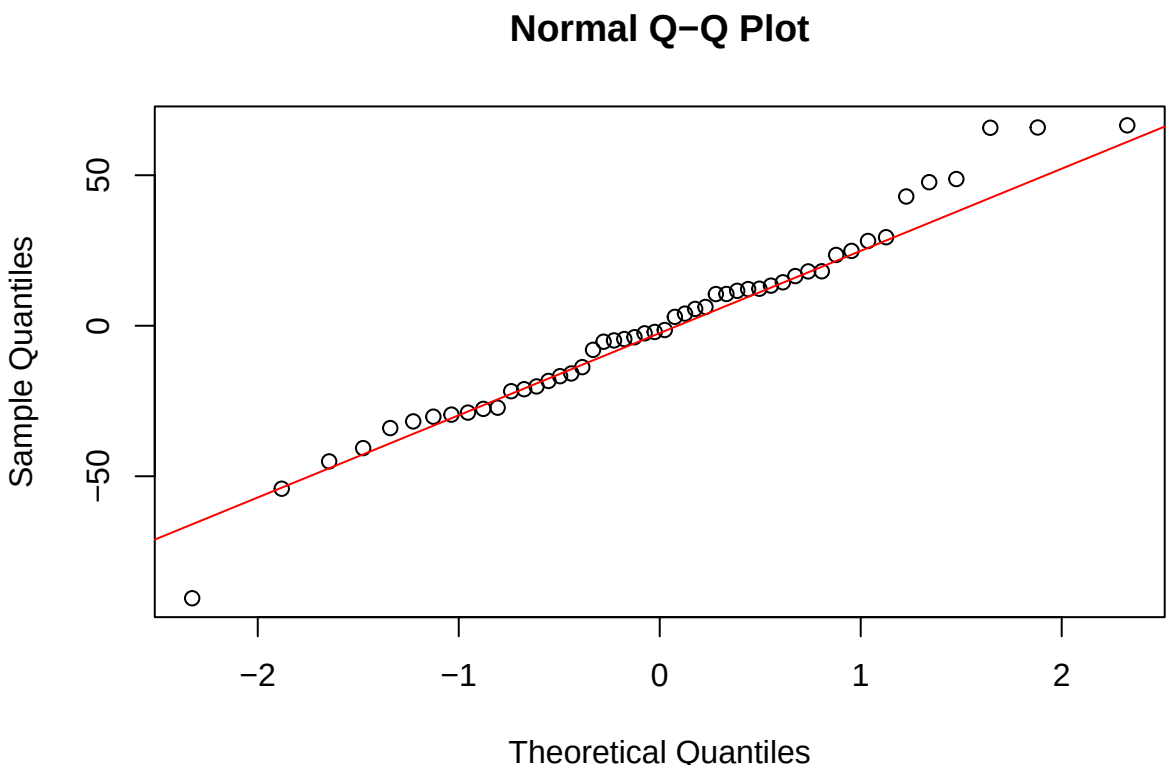
[1] If you don't find any of these datasets interesting, you still have to do the problem. Nice try.



The constant variance assumption is **reasonable** because the residual plot demonstrates that for all fitted values, there appears to be somewhat similar residual distribution around 0. If this assumption was not met, we would see more variation for different fitted values.

(b) Check the normality assumption.

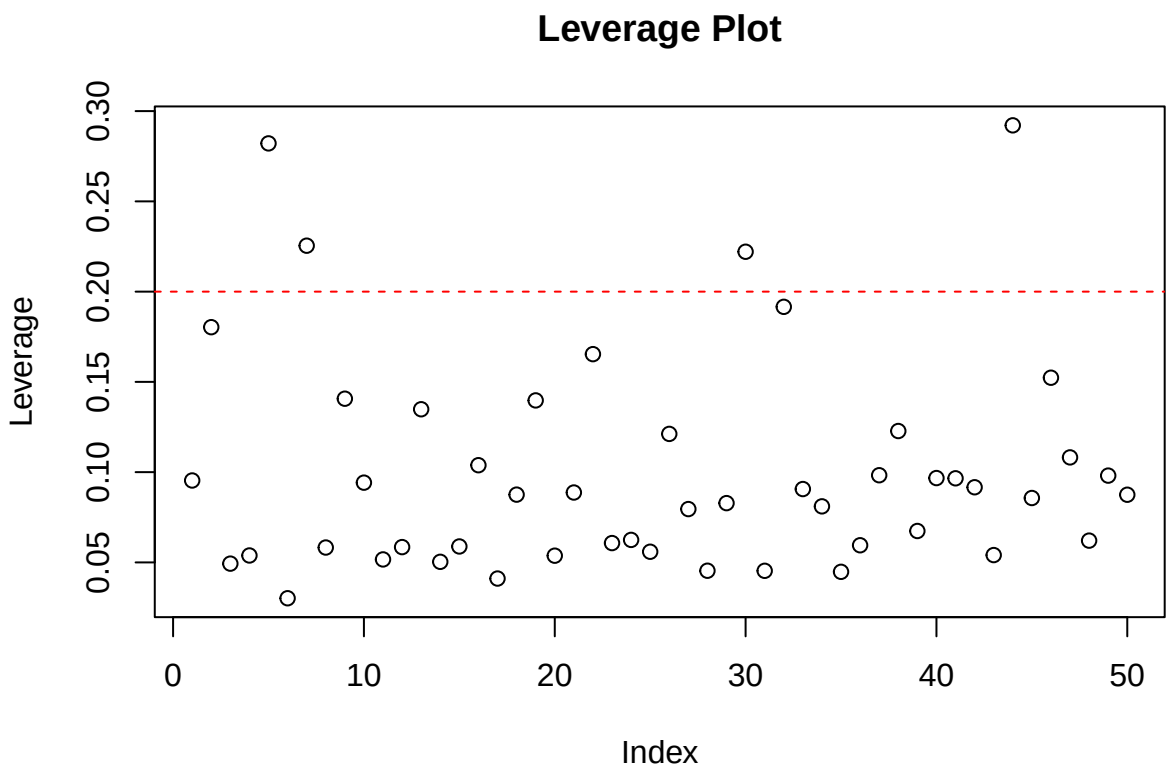
```
qqnorm(residuals(model))
qqline(residuals(model), col = "red")
```



The normality assumption is **reasonable** because the QQ plot demonstrates that the data is mostly normal. The QQ plot shows us this because the sample quantiles are shown to have a mostly linear relationship with the theoretical quantiles. More linear data demonstrates that the data is more normal. However, we do see some deviation at the tails, which is common becuase this is data is not perfectly normal – it is still an approximation.

(c) Check for large leverage points.

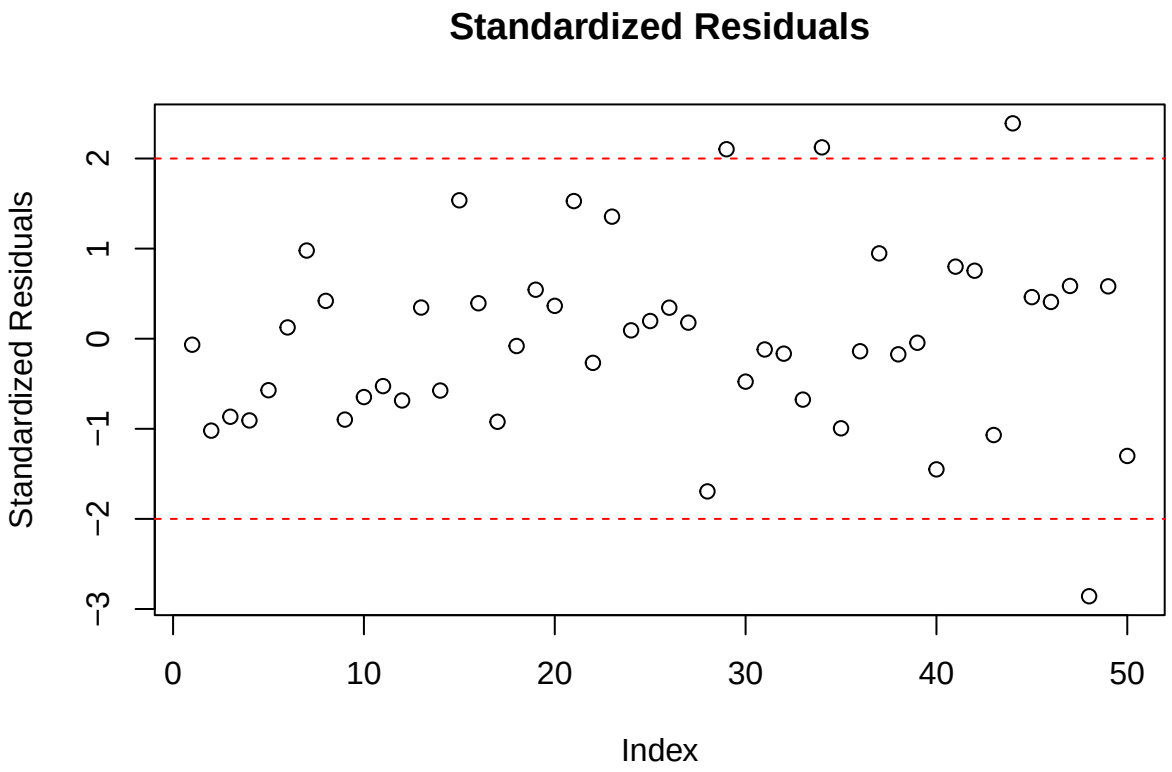
```
leverage <- hatvalues(model)
plot(leverage, main = "Leverage Plot", ylab = "Leverage")
abline(h = 2 * mean(leverage), col = "red", lty = 2)
```



Based on this graph, there appear to be 4 large leverage points. The red linei s set at 2x the average leverage so anything above it is considered high leverage.

(d) Check for outliers.

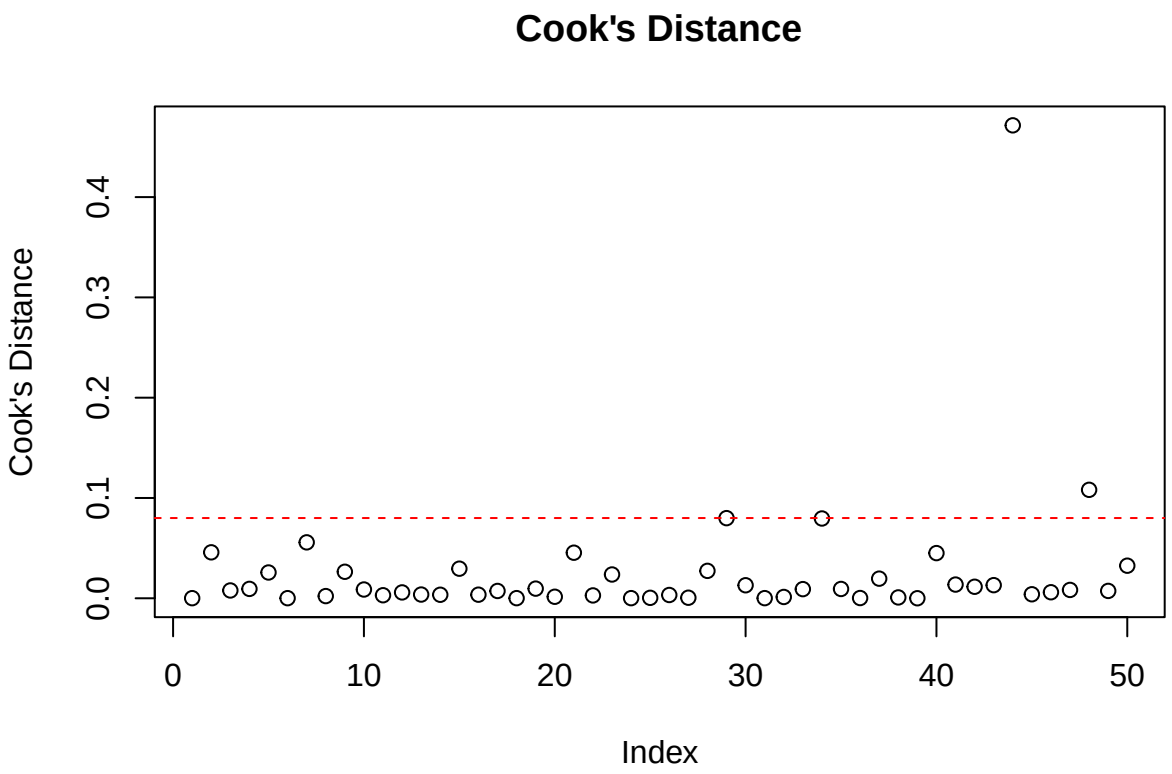
```
std_residuals <- rstandard(model)
plot(std_residuals, main = "Standardized Residuals", ylab = "Standardized Residuals")
abline(h = c(-2, 2), col = "red", lty = 2)
```



There appear to be 4 potential outliers becuase they appear outside the red lines.

(e) Check for influential points.

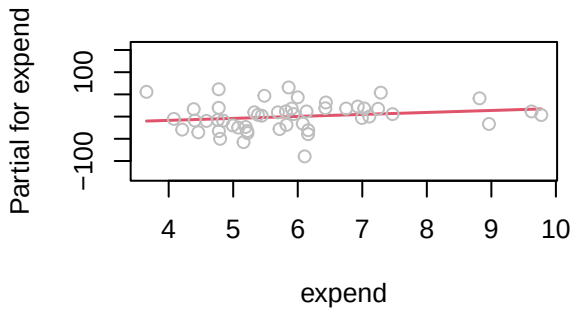
```r
# cooks distance to calculate influential points
cooks_d <- cooks.distance(model)
plot(cooks_d, main = "Cook's Distance", ylab = "Cook's Distance")
abline(h = 4 / length(cooks_d), col = "red", lty = 2)
```

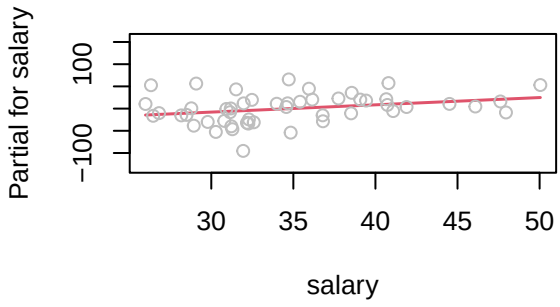


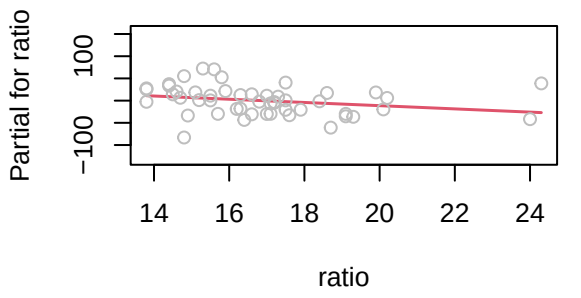
Some poitns fall directly on the line; including these points, there are 4 potential influential points. The line is set at 4/n.

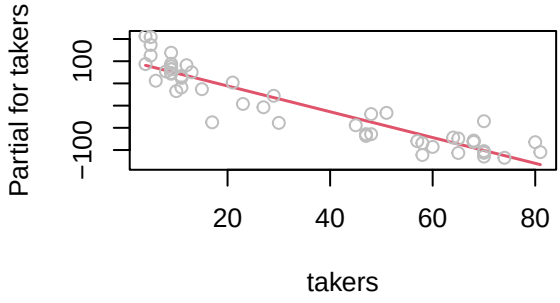
(f) Check the structure of the relationship between the predictor and the response.

```
par(mfrow = c(2, 2))
termplot(model, partial.resid = TRUE, terms = c("expend", "salary", "ratio", "takers"))
```









The

structure of the model looks to be **pretty good** becuase each of the predictors is well fitted to the linear best fit line. These graphs show when we control for each of the other variables and only plot one at a time. This is important to see so that we can see the actual relationship for each variable without having to create a crazy 5D plot :) I wonder if there is colinearity between expend and slaray, as they seem to follow very similar trends. Since the slope of the lines is not strong for expend, salary, and ratio, we wonder how strong of predictors these variables are. Takers seems to be a much stronger predictor, becuase it has a greater (magnitude) of slope. more analysis is necessary for this to be conclusive.

Datasets:

`sat`: contains data on school expenditure and test scores in the US in 1994-95 (in the `faraway` package). For this dataset, fit a model with total SAT score (`total`) as the dependent variable, and `expend`, `salary`, `ratio` and `takers` as the independent variables.

`teengamb`: contains data on a survey conducted on teenage gambling in Britain (in the `faraway` package). For this dataset, fit a model with `gamble` as the dependent variable and all the other variables as independent variables.

`prostate`: contains data on a study of men with prostate cancer due to receive radical prostatectomy (in the `faraway` package). For this dataset, fit a model with `lpsa` as the dependent variable, and all the other variables as independent variables.

`swiss`: contains standardized fertility measure and socioeconomic indicators for each of 47 French-speaking provinces of Switzerland around 1888 (in `datasets` package). For this dataset, fit a model with `Fertility` as the dependent variable and all the other variables as the independent variables.

`cheddar`: contains data on a study of cheddar cheese from the LaTrobe Valley of Victoria, Australia, in which samples of cheese were analyzed for their chemical composition and were subjected to taste tests (in the `faraway` package). Overall taste scores were obtained by combining the scores from several tasters. For this dataset, fit a model with `taste` as the dependent variable and the other three variables as the independent variables.

`happy`: contains data on happiness from a sample of students collected in a University of Chicago MBA class (in the `faraway` package). For this dataset, fit a model with `happy` as the dependent variable, and the other four variables as independent variables.

`tvdoctor`: contains data on life expectancy, doctors and televisions collected on 38 countries in 1993 (in the `faraway` package). For this dataset, fit a model with `life` as the dependent variable and the other two variables as independent variables.

Question 3: ANOVA and Regression

The included dataset `mouse.csv` contains body weight data on a sample of mice from eight inbred strains that are the founder strains used to create a resource known as the *Collaborative Cross*. In this problem, we will hold off on doing diagnostics until part **(d)**, even though in a real-world analysis, any interpretation should come after you perform your diagnostics, and satisfactorily address any potential issues in your dataset.

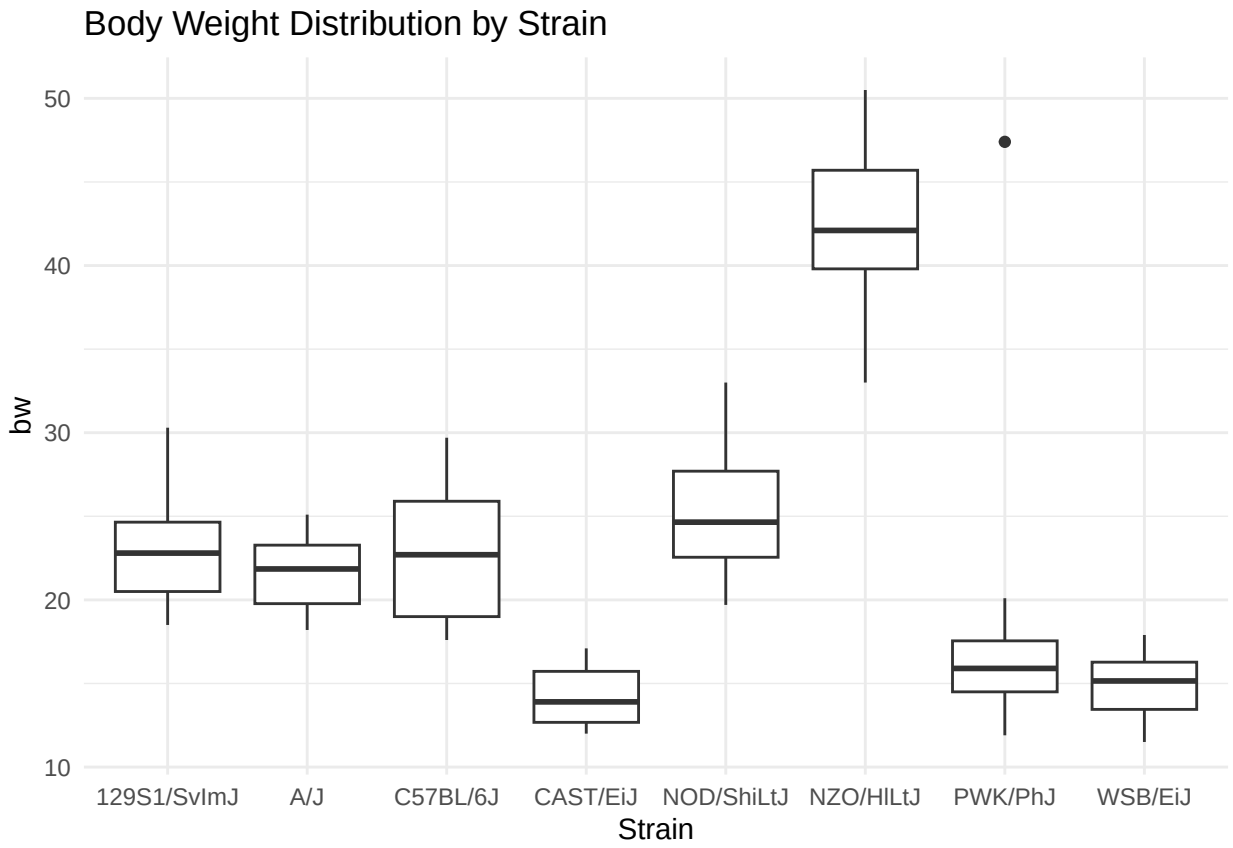
```
# eda
data <- read.csv('data/mouse.csv')
summary(data)
```

```
##     strain                bw
##  Length:276         Min.   :11.50
##  Class :character   1st Qu.:15.78
##  Mode  :character   Median :20.10
##                     Mean   :22.29
##                     3rd Qu.:25.12
##                     Max.   :50.50
```

```
str(data)
```

```
## 'data.frame':    276 obs. of  2 variables:
##  $ strain: chr  "129S1/SvImJ" "129S1/SvImJ" "129S1/SvImJ" "129S1/SvImJ" ...
##  $ bw    : num  24.2 20.7 20 21.8 24.1 21.2 21.8 20.5 20.2 20.5 ...
```

```
library(ggplot2)

ggplot(data, aes(x = strain, y = bw)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Body Weight Distribution by Strain",
       x = "Strain", y = "bw")
```




```r
library(dplyr)

strain_summary <- data %>%
  group_by(strain) %>%
  summarise(
    mean_weight = mean(bw),
    median_weight = median(bw),
    sd_weight = sd(bw),
    min_weight = min(bw),
    max_weight = max(bw)
  )

print(strain_summary)
```

```
## # A tibble: 8 x 6
##   strain      mean_weight median_weight sd_weight min_weight max_weight
##   <chr>             <dbl>         <dbl>     <dbl>      <dbl>      <dbl>
## 1 129S1/SvImJ        23.3          22.8      3.16       18.5       30.3
## 2 A/J                21.6          21.8      2.09       18.2       25.1
## 3 C57BL/6J           22.9          22.7      3.89       17.6       29.7
## 4 CAST/EiJ           14.2          13.9      1.65       12         17.1
## 5 NOD/ShiLtJ         25.1          24.6      3.37       19.7       33
## 6 NZO/HlLtJ          42.4          42.1      4.28       33         50.5
## 7 PWK/PhJ            16.5          15.9      5.27       11.9       47.4
## 8 WSB/EiJ            15.0          15.2      1.77       11.5       17.9
```

(a) Using ANOVA (that is, use the `aov()` function in R) formally test the hypothesis that the population body weights for all eight strains are equal. Make sure to formally state your null and alternative hypotheses,

your test statistic, the level of your test, and the associated p-value. Describe your conclusions in language suitable for a non-statistician collaborator.

```
anova_result <- aov(bw ~ strain, data = data)
summary(anova_result)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## strain        7  19236  2748.0   227.5 <2e-16 ***
## Residuals   268   3237    12.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on our ANOVA, we can (with reasonable confidence) say that the strain is a predictor for body weight. In more layman's terms, this means that the different strains of inbred mice likely have different body weights. So, after we do some more analysis, we will be able to identify which mice have different body weights and perhaps use that to predict the body weights of new mice that we find (not sure where you find these inbred mice, but ok!).

(b) Now fit the model with `lm()`, including an intercept. Interpret that intercept. Note the correspondence between the F-statistic from the regression model, and that in the ANOVA table.

```
lm_model <- lm(bw ~ strain, data = data)
summary(lm_model)
```

```
##
## Call:
## lm(formula = bw ~ strain, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3697 -2.1734 -0.2973  1.8268 30.8605
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       23.2781     0.6144  37.889  < 2e-16 ***
## strainA/J         -1.6531     0.8688  -1.903   0.0582 .
## strainC57BL/6J    -0.3937     0.8688  -0.453   0.6508
## strainCAST/EiJ    -9.0670     0.8444 -10.738  < 2e-16 ***
## strainNOD/ShiLtJ   1.8572     0.8560   2.170   0.0309 *
## strainNZO/HlLtJ   19.0916     0.8622  22.142  < 2e-16 ***
## strainPWK/PhJ     -6.7386     0.8114  -8.305 4.99e-15 ***
## strainWSB/EiJ     -8.3193     0.8560  -9.719  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.475 on 268 degrees of freedom
## Multiple R-squared:  0.856,  Adjusted R-squared:  0.8522
## F-statistic: 227.5 on 7 and 268 DF,  p-value: < 2.2e-16
```

Interpretations: the intercept represents the baseline mouse. This mouse strain is considered to be the standard and all other mice are compared to it. In this case, the strain is the 129S1/SvImJ strain. We know this becuase it is the only strain missing from the table. The coefficients represent how much on average the different strains of mice weigh in comparison to the average of the baseline strain. (e.g.a/j is on average 1.6 units lighter than the baseline). Interestingly, the F statistic from the ANOVA and the regresssion are the same! This makes sense because we can interpret a linear regression for categorical predictors as an ANOVA – they are the same. intuition: We are testing the same hypothesis: H_0: all the mice weights are the same for the strains / the strains are not predictors of bw. H_A: The mice weights are different for different strains /

the strains are predictors of bw.

(c) We mentioned in class that the one-way ANOVA "Sum of Squares Between" corresponds to the additional sum of squares accounted for by including the factor of interest, compared to fitting an intercept-only model. Show that this is the case in this scenario by computing the sum of squares accounted for by an intercept-only model, and the sum of squares from the `lm()` fit that controls for strain. Show that the difference between these two sums of squares corresponds to the "Sum of Squares Within" in your ANOVA table in part **(a)**. No need to do any math here, just compute the relevant quantities in R.

```
# Fit intercept-only model
intercept_only_model <- lm(bw ~ 1, data = data)

# Summary of intercept-only model
summary(intercept_only_model)
```

```
##
## Call:
## lm(formula = bw ~ 1, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.795  -6.520  -2.195   2.830  28.205
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.2949     0.5441   40.97   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.04 on 275 degrees of freedom
```

```
# Extract Total Sum of Squares (SST)
anova_intercept <- anova(intercept_only_model)
SST <- anova_intercept$`Sum Sq`[1]
cat("Total Sum of Squares (SST) from intercept-only model:", SST, "\n")
```

```
## Total Sum of Squares (SST) from intercept-only model: 22473.31
```

```
# Fit model with strain as predictor
strain_model <- lm(bw ~ strain, data = data)

# Summary of strain model
summary(strain_model)
```

```
##
## Call:
## lm(formula = bw ~ strain, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3697 -2.1734 -0.2973  1.8268 30.8605
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       23.2781     0.6144  37.889  < 2e-16 ***
## strainA/J         -1.6531     0.8688  -1.903   0.0582 .
## strainC57BL/6J    -0.3937     0.8688  -0.453   0.6508
```

```
## strainCAST/EiJ    -9.0670     0.8444 -10.738  < 2e-16 ***
## strainNOD/ShiLtJ   1.8572     0.8560   2.170   0.0309 *
## strainNZO/HlLtJ   19.0916     0.8622  22.142  < 2e-16 ***
## strainPWK/PhJ     -6.7386     0.8114  -8.305 4.99e-15 ***
## strainWSB/EiJ     -8.3193     0.8560  -9.719  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.475 on 268 degrees of freedom
## Multiple R-squared:  0.856,  Adjusted R-squared:  0.8522
## F-statistic: 227.5 on 7 and 268 DF,  p-value: < 2.2e-16
```

```
# Extract Residual Sum of Squares (SSW)
anova_strain <- anova(strain_model)
SSW <- anova_strain$`Sum Sq`[2]
cat("Residual Sum of Squares (SSW) from strain model:", SSW, "\n")
```

```
## Residual Sum of Squares (SSW) from strain model: 3237.005
```

```
# Compute Sum of Squares Between (SSB)
SSB <- SST - SSW
cat("Sum of Squares Between (SSB):", SSB, "\n")
```

```
## Sum of Squares Between (SSB): 19236.31
```

```
# ANOVA result from part (a)
anova_a <- summary(anova_result)
SSB_a <- anova_a[[1]]$`Sum Sq`[1]
cat("Sum of Squares Between from ANOVA (part a):", SSB_a, "\n")
```

```
## Sum of Squares Between from ANOVA (part a): 19236.31
```

```
# Compare with computed SSB
cat("Computed Sum of Squares Between (SSB):", SSB, "\n")
```

```
## Computed Sum of Squares Between (SSB): 19236.31
```

```
# Check if they are equal (within a small tolerance)
if (abs(SSB - SSB_a) < 1e-6) {
  cat("The computed SSB matches the Sum of Squares Between from ANOVA (part a).\n")
} else {
  cat("There is a discrepancy between the computed SSB and the ANOVA Sum of Squares Between.\n")
}
```

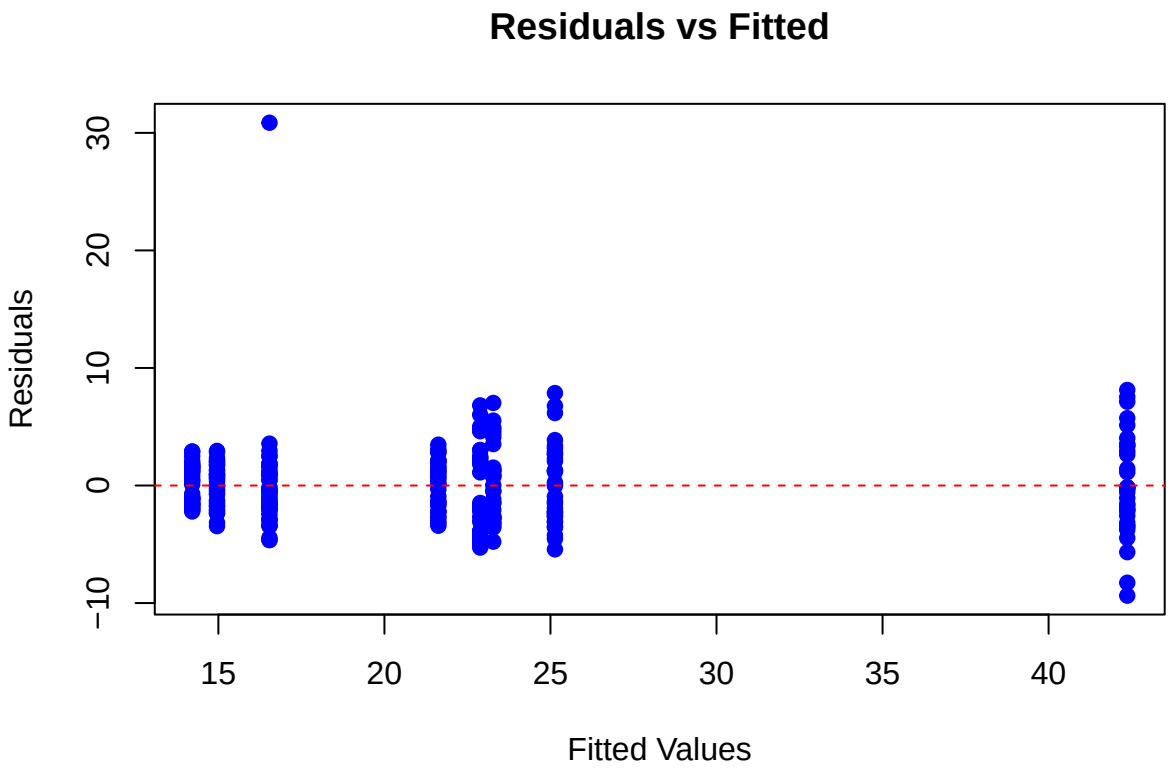
```
## The computed SSB matches the Sum of Squares Between from ANOVA (part a).
```

Thus it is proven that thesum of squares between from the linear regression and the anova are the same – showing that they are analagous for categorical predictor variables

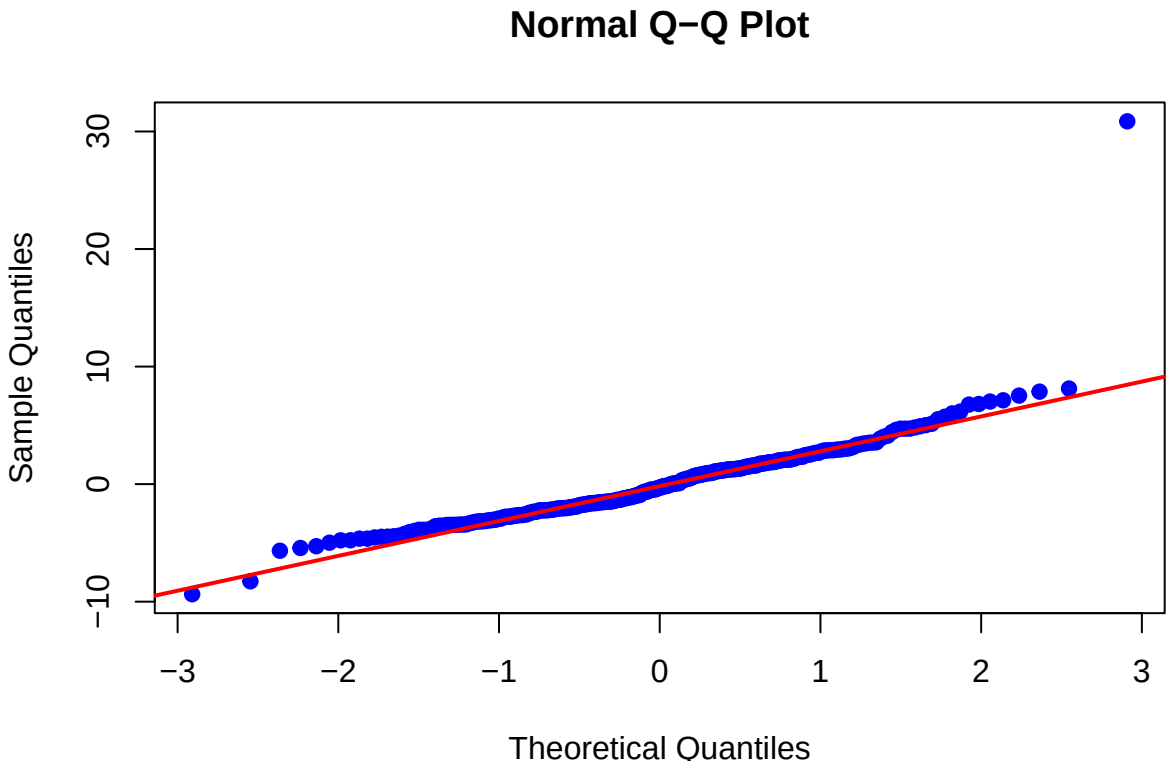
(d) Now, compute residual diagnostics from your model in part **(b)**. Suppose you were actually analyzing this dataset for a collaborator. Based on your diagnostics, what should you have discussed with your collaborator before you proceeeded to analyze these data? Do you have any idea of what might have happened in this case?

```
# Residuals vs Fitted Plot
plot(lm_model$fitted.values, lm_model$residuals,
     main = "Residuals vs Fitted",
     xlab = "Fitted Values",
     ylab = "Residuals",
```

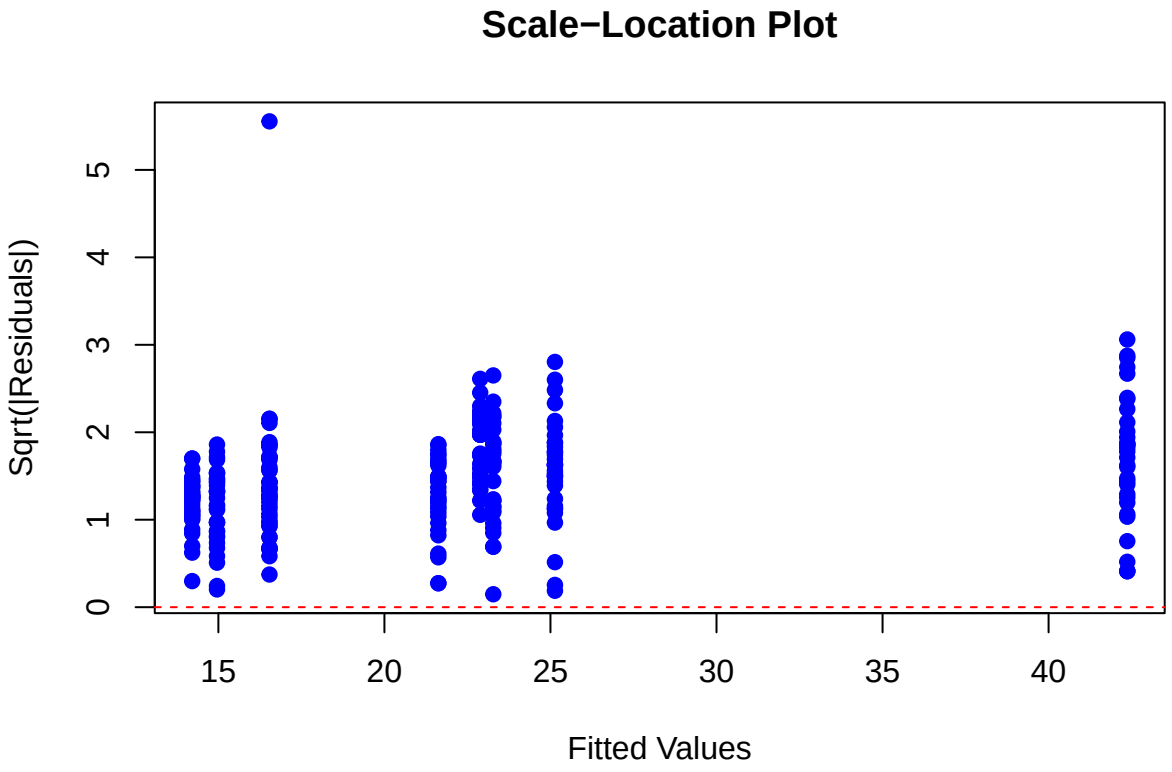
```
     pch = 19, col = "blue")
abline(h = 0, lty = 2, col = "red")
```



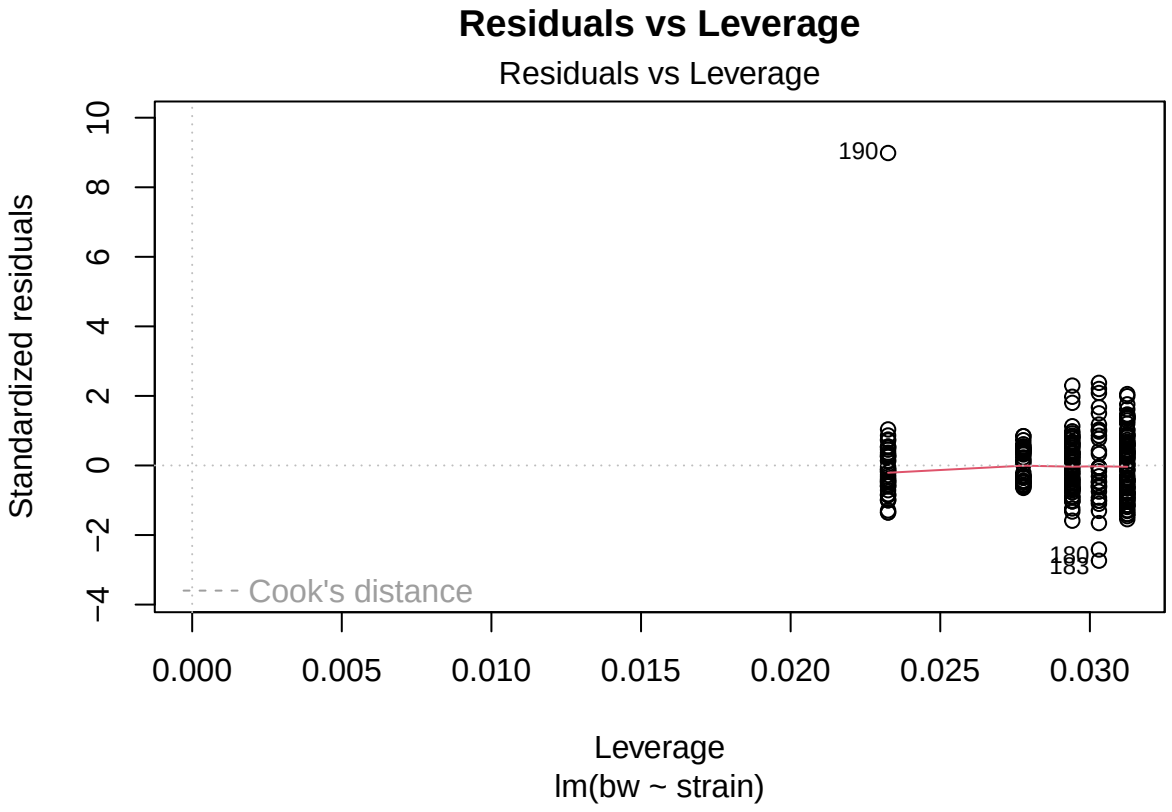

```
# Normal Q-Q Plot
qqnorm(lm_model$residuals,
       main = "Normal Q-Q Plot",
       pch = 19, col = "blue")
qqline(lm_model$residuals, col = "red", lwd = 2)
```



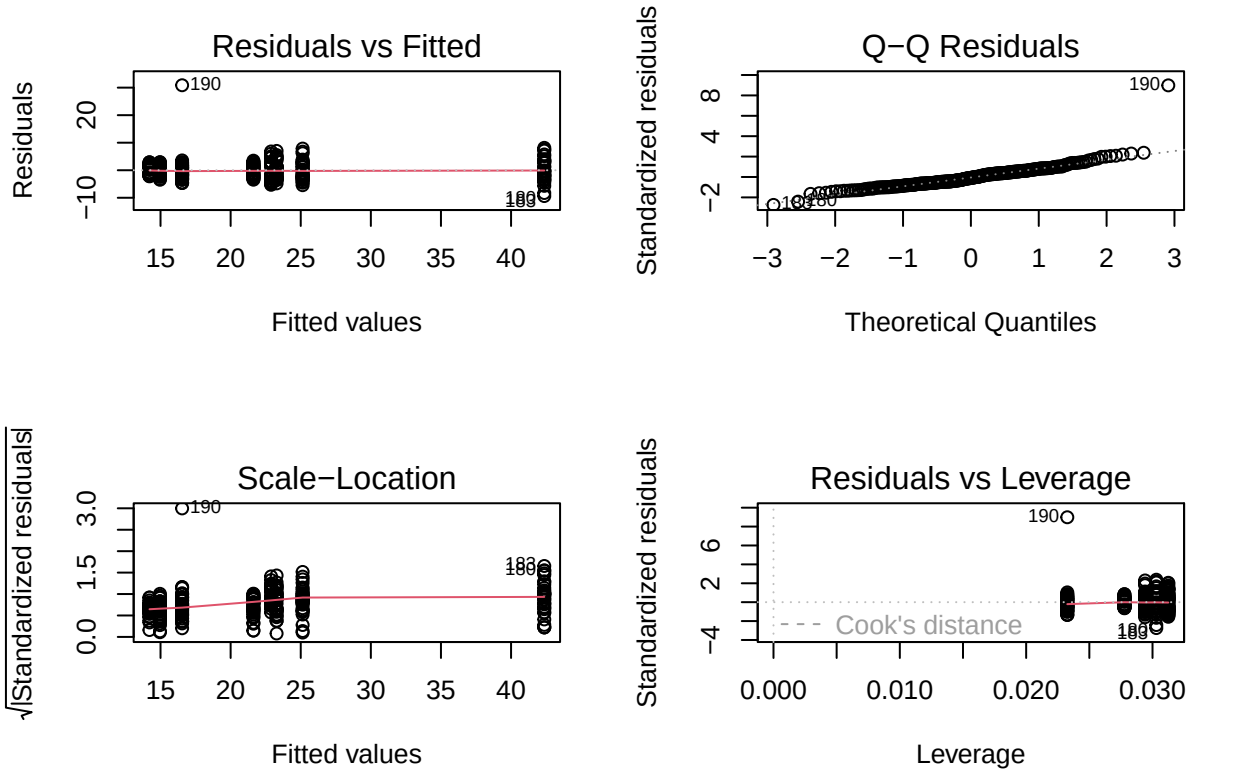

```r
# Scale-Location Plot
sqrt_abs_residuals <- sqrt(abs(lm_model$residuals))
plot(lm_model$fitted.values, sqrt_abs_residuals,
     main = "Scale-Location Plot",
     xlab = "Fitted Values",
     ylab = "Sqrt(|Residuals|)",
     pch = 19, col = "blue")
abline(h = 0, lty = 2, col = "red")
```




```r
# Residuals vs Leverage Plot
plot(lm_model, which = 5,
     main = "Residuals vs Leverage")
```



```r
# Generate all standard diagnostic plots
par(mfrow = c(2, 2))  # Arrange plots in a 2x2 grid
plot(lm_model)
```




```r
par(mfrow = c(1, 1))  # Reset to default
```

E

assumed to be true becuase we are able to plot the data

L

Based on the residuals vs fitted plot, we see that there are in fact 8 different 'columns' of data – each distirbuted randomly around 0. this means that our linearity assumption is reasonable. This shows that there is random variation that is not dependent on the intragroup variation. However, the weights are very clearly dependnet on the strain.

I

We do not have enough information about the data to confirm or deny that they are independent. Independence refers to the way that they data were collected. The table does not contain this

H

Based on the residuals vs fitted plot, we see that there are in fact 8 different 'columns' of data – each distirbuted randomly around 0. this means that our homoskedasticity and linearity assumptions are reasonable. note that there still is some violation of the homoskedasticity assumption because there is more spread at larger fitted values. This shows that there is random variation that is not dependent on the intragroup variation. However, the weights are very clearly dependnet on the strain. The data is NOT perfectly homoskedastic, but it is relatively close.

normality of errors

Normality assumption is achieved because the QQ plot demonstrates that the residuals fall along the theoretical quantiles so it is approximately normal.

(e) Did you actually need to check the linearity assumption in part (d)? Why or why not? Just provide a little intuition.

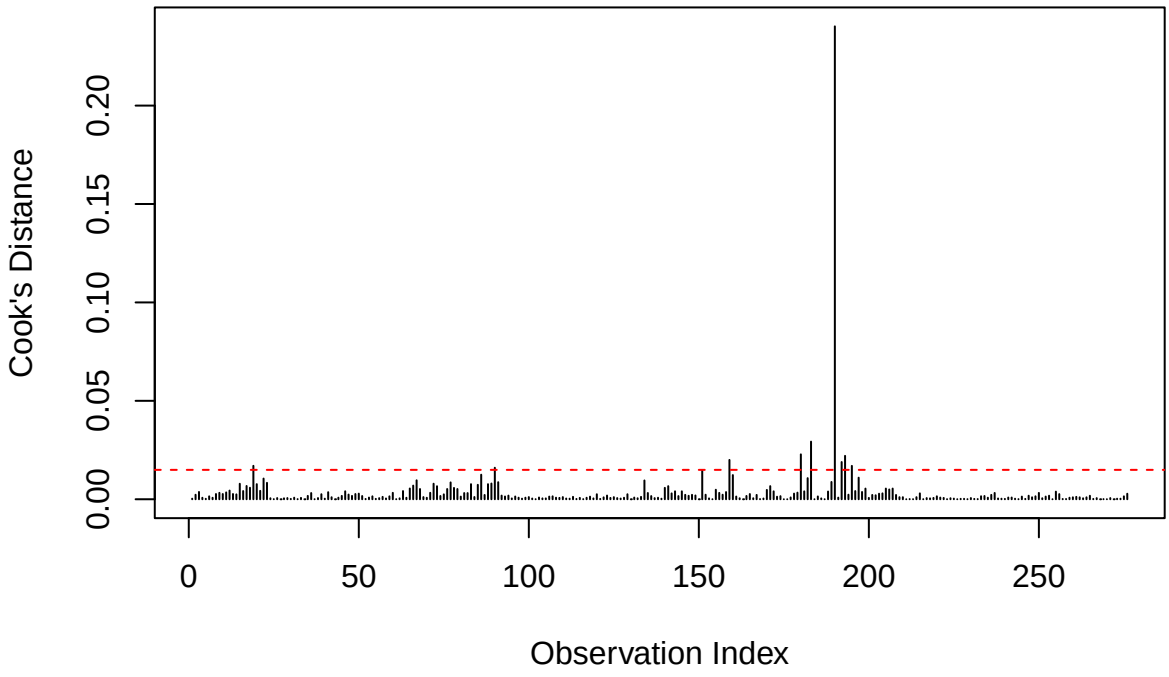
No we did not becuase we have categorical predictors. In this case, we interpret the idea of linearity differently. In ANOVA, we are not worried about if the data forms a linear line (because that doesn't seem to have an intuitive interpretation) – we are concerned about intergroup variation and intragroup variation. This means that as long as we have collected data that captures the group means, we should have accurate results. If we do not capture the true mean, we will have a biased analysis. In the linear regresssion, the coefficeints do not represent the "slope"–rather they represent the deviation from the baseline mouse strain in mean weight. Thus, linearity does not need to be chceked.

(f) Address any issues in the data as you see fit (you can discuss with a TF in office hours and they will pretend to be your collaborator). Refit the model from part **(b)**. If your overall F-test is significant (it should be very significant!), proceed to pairwise tests for the differences in mean weights between every pair of strains (hint: you saw a function in class that should make this very easy). In your pairwise t-tests, use the Bonferroni-adjusted p-values. Does there seem to be some structure or grouping to the strains?

Data cleaning and better practices: - We should properly handle outliers (perhaps removing them) - We should find a way to handle heteroskedasticity or account for this difference (instead of assuming perfect heteroskedasticity) - We do not know for sure if we have captured the mean of the data for each mouse strain. Ideally, we could consult other data sources to validate that our estimates are reasonable (and that we are not observing some other variation) - We would like to see some of the methods that were used to collect the data–this could help us identify if our independence assumption is valid. For example, it would be VERY relevant to know if the different mouse groups were weighed on different scales, or at different times in their lives, or had different diets etc. There are a lot of potential confounding variables that I would like to see explained. - To handle heteroskedasticity, we would do a log transform.

Now, lets code the solution (what we can lol)

```
# Fidn outliers
cooks_d <- cooks.distance(lm_model)

# Plot Cook's Distance
plot(cooks_d, type = "h",
     main = "Cook's Distance for Each Observation",
     ylab = "Cook's Distance",
     xlab = "Observation Index")
abline(h = 4/(nrow(data)-length(lm_model$coefficients)), col = "red", lty = 2)
```



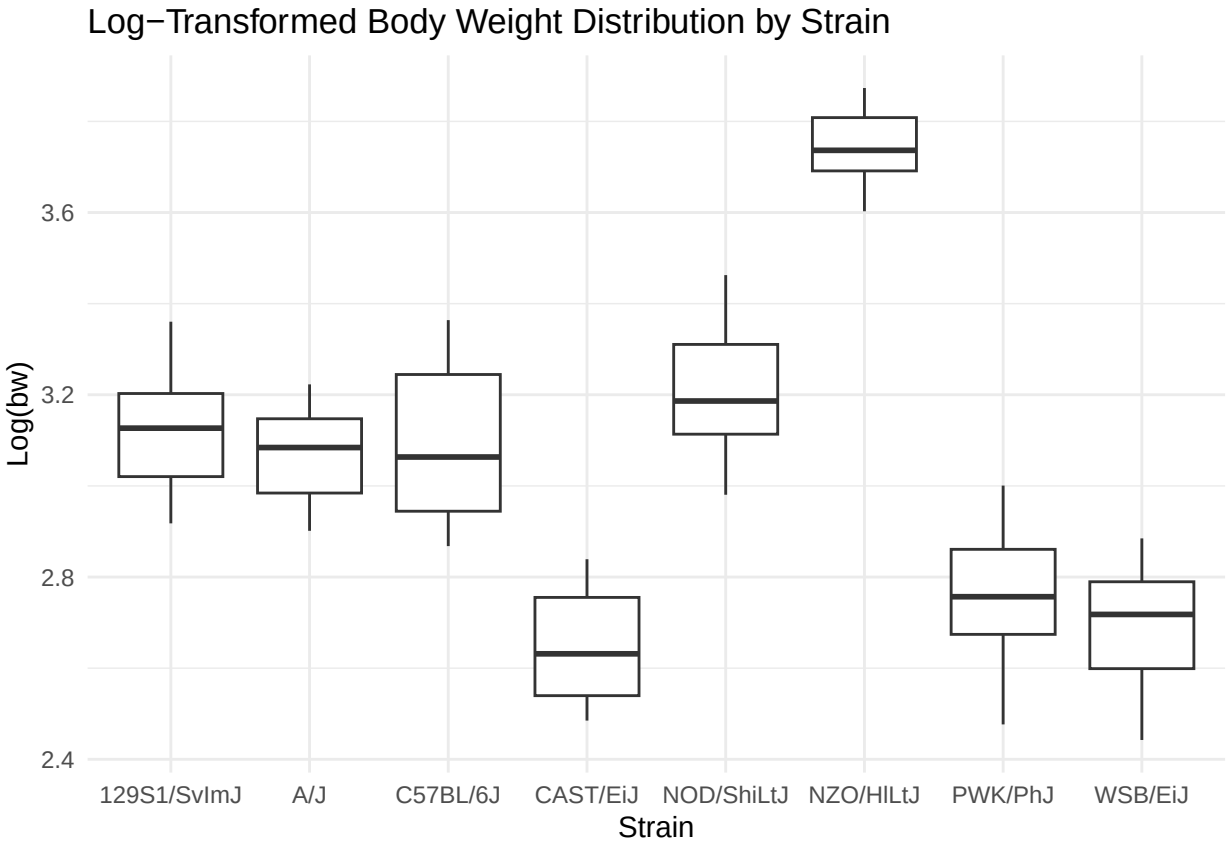

```
# Identify observations with Cook's Distance greater than the threshold
influential_threshold <- 4/(nrow(data) - length(lm_model$coefficients))
influential_points <- which(cooks_d > influential_threshold)
cat("Influential Observations (Cook's D > ", influential_threshold, "): ", influential_points, "\n")
```

```
## Influential Observations (Cook's D >  0.01492537 ):  19 90 159 180 183 190 192 193 195
```

```
# Remove influential observations
data_refined <- data[-influential_points, ]

cat("Number of observations after removing influential points:", nrow(data_refined), "\n")
```

```
## Number of observations after removing influential points: 267
```

```
# Apply log transformation to body weight
data_refined$log_bw <- log(data_refined$bw)

ggplot(data_refined, aes(x = strain, y = log_bw)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Log-Transformed Body Weight Distribution by Strain",
       x = "Strain", y = "Log(bw)")
```




```r
# Fit the refined linear model with log-transformed body weight
lm_refined <- lm(log_bw ~ strain, data = data_refined)

# Summary of the refined model
summary(lm_refined)
```

```
##
## Call:
## lm(formula = log_bw ~ strain, data = data_refined)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.274608 -0.103532  0.002882  0.099500  0.256286
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.13007    0.02216 141.232   <2e-16 ***
## strainA/J        -0.06079    0.03110  -1.955   0.0517 .
## strainC57BL/6J   -0.02252    0.03134  -0.718   0.4732
## strainCAST/EiJ   -0.48255    0.03023 -15.960   <2e-16 ***
## strainNOD/ShiLtJ  0.07722    0.03086   2.502   0.0130 *
## strainNZO/HlLtJ   0.60947    0.03217  18.944   <2e-16 ***
## strainPWK/PhJ    -0.37892    0.02922 -12.969   <2e-16 ***
## strainWSB/EiJ    -0.43176    0.03064 -14.090   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1234 on 259 degrees of freedom
## Multiple R-squared:  0.8761, Adjusted R-squared:  0.8727
## F-statistic: 261.6 on 7 and 259 DF,  p-value: < 2.2e-16
```

From this we see that the F statistic is huge (even bigger than the previous lm we fit!) Now let's do some pairwise analysis to check the inter-strain differences and then correct with pepperoni correction

```
#  pairwise t-tests with Bonferroni
pairwise_results <- pairwise.t.test(data_refined$log_bw, data_refined$strain,
                                    p.adjust.method = "bonferroni",
                                    pool.sd = FALSE, # Use separate variances
                                    paired = FALSE)


print(pairwise_results)
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  data_refined$log_bw and data_refined$strain
##
##            129S1/SvImJ A/J     C57BL/6J CAST/EiJ NOD/ShiLtJ NZO/HlLtJ PWK/PhJ
## A/J        1.00000     -       -        -        -          -         -
## C57BL/6J   1.00000     1.00000 -        -        -          -         -
## CAST/EiJ   < 2e-16     < 2e-16 < 2e-16  -        -          -         -
## NOD/ShiLtJ 0.44121     0.00014 0.24511  < 2e-16  -          -         -
## NZO/HlLtJ  < 2e-16     < 2e-16 < 2e-16  < 2e-16  < 2e-16    -         -
## PWK/PhJ    < 2e-16     < 2e-16 2.4e-12  0.01509  < 2e-16    < 2e-16   -
## WSB/EiJ    < 2e-16     < 2e-16 1.9e-14  1.00000  < 2e-16    < 2e-16   1.00000
##
## P value adjustment method: bonferroni
```

data: CAST/EiJ vs. All Other Strains: All comparisons involving CAST/EiJ have p $<$ 2e-16, indicating highly significant differences in mean body weight compared to every other strain. NOD/ShiLtJ vs. A/J: p = 0.00014 — Significantly different. PWK/PhJ vs. C57BL/6J: p = 2.4e-12 — Highly significantly different. PWK/PhJ vs. NOD/ShiLtJ: p = 0.01509 — Significantly different. CAST/EiJ and NZO/HlLtJ strains have distinctly different mean body weights compared to most other strains. NOD/ShiLtJ differs significantly from A/J, CAST/EiJ, and NZO/HlLtJ. PWK/PhJ differs significantly from C57BL/6J and NOD/ShiLtJ.

Interpretations: After refining our dataset by removing influential outliers and applying a log transformation to stabilize variance, we refitted our regression model. The ANOVA remains highly significant, confirming that mouse strain is a strong predictor of body weight. Through pairwise comparisons with Bonferroni-adjusted p-values, we identified specific strains with significantly different mean body weights.

(g) If you are interested, read a little bit online about the Collaborative Cross and see if your results from part **(f)** make sense scientifically.[2]

very cool!! :)

[2]This part will not be graded, although you might find it interesting!

Question 4: Interpretation of Parameter Estimates

The included dataset `harvardsqhomes.csv` contains data on a sample of homes in the Harvard Square area that were on the market in 2022. Among others, it includes the following variables:

`price`: the price of the home (in dollars)

`beds`: the number of bedrooms in the home

`sqft`: the square footage of the home

`baths`: the number of bathrooms in the home

`year`: the year the home was built

```
data <- read.csv('data/harvardsqhomes.csv')
summary(data)
```

```
##      date               type              address              city
##  Length:349         Length:349         Length:349         Length:349
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##       zip           price              beds            baths
##  Min.   :2138   Min.   : 291500   Min.   : 0.000   Min.   :1.000
##  1st Qu.:2139   1st Qu.: 750000   1st Qu.: 2.000   1st Qu.:1.000
##  Median :2139   Median :1050000   Median : 2.000   Median :2.000
##  Mean   :2141   Mean   :1291537   Mean   : 2.782   Mean   :2.107
##  3rd Qu.:2141   3rd Qu.:1650000   3rd Qu.: 3.000   3rd Qu.:2.500
##  Max.   :2414   Max.   :6850000   Max.   :17.000   Max.   :9.500
##
##  neighborhood.9          sqft         lotsize           year
##  Length:349         Min.   : 294   Min.   : 1025   Min.   :1805
##  Class :character   1st Qu.: 871   1st Qu.: 2092   1st Qu.:1894
##  Mode  :character   Median :1191   Median : 3049   Median :1915
##                     Mean   :1551   Mean   : 3608   Mean   :1931
##                     3rd Qu.:1897   3rd Qu.: 4042   3rd Qu.:1982
##                     Max.   :8737   Max.   :13873   Max.   :2022
##                                    NA's   :278
##       hoa              url                mls              latitude
##  Min.   :    81   Length:349         Min.   :72809964   Min.   :42.36
##  1st Qu.:   231   Class :character   1st Qu.:72953839   1st Qu.:42.37
##  Median :   341   Mode  :character   Median :72969397   Median :42.37
##  Mean   : 18858                      Mean   :72970006   Mean   :42.37
##  3rd Qu.:   505                      3rd Qu.:72989817   3rd Qu.:42.37
##  Max.   :999999                      Max.   :73026427   Max.   :42.38
##  NA's   :132                         NA's   :60
##    longitude
##  Min.   :-71.13
##  1st Qu.:-71.11
##  Median :-71.10
##  Mean   :-71.10
##  3rd Qu.:-71.10
##  Max.   :-71.08
```

##

(a) Fit a simple linear regression with price **in thousands of dollars** as the dependent variable and `beds` as the sole predictor. Interpret the coefficient estimate associated with `beds`.

```
# Convert price to thousands of dollars
data$price_thousands <- data$price / 1000

# Fit the linear regression model
model <- lm(price_thousands ~ beds, data = data)

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

```
##
## Call:
## lm(formula = price_thousands ~ beds, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1429.4  -299.6  -142.0   181.5  4760.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   601.59      49.25   12.21   <2e-16 ***
## beds          247.98      13.85   17.90   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 572.8 on 347 degrees of freedom
## Multiple R-squared:  0.4801, Adjusted R-squared:  0.4786
## F-statistic: 320.4 on 1 and 347 DF,  p-value: < 2.2e-16
```

the coefficients: Intercept: For a house with "no beds" will be about $600,000. This is the baseline Slope: For each additional bed, we perdict that there will be about a $250,000 price increase in house in our model given the common linear regression assumptions (which we have not tested yet!)

We are assuming that the assumptions for linear regression are met, but we did not actually check!

(b) Now fit a multiple linear regression by adding `sqft`, `baths` and `year` to your model. Interpret the estimate associated with `beds` again. Reconcile the estimate from this model with the one from part **(a)**. Does it make sense why and how it changed? No math required here, just some intuition/explanation.

```
# Fit the multiple linear regression model
multiple_model <- lm(price_thousands ~ beds + sqft + baths + year, data = data)

# Display the summary of the model
summary(multiple_model)
```

```
##
## Call:
## lm(formula = price_thousands ~ beds + sqft + baths + year, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1312.2  -217.0   -54.7   143.8  3421.2
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   77.3501   942.7410   0.082    0.935
## beds        -132.4948    28.8423  -4.594 6.12e-06 ***
## sqft           0.5847     0.0608   9.617  < 2e-16 ***
## baths        202.6394    38.0932   5.320 1.88e-07 ***
## year           0.1290     0.4870   0.265    0.791
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 453.1 on 344 degrees of freedom
## Multiple R-squared:  0.6775, Adjusted R-squared:  0.6737
## F-statistic: 180.6 on 4 and 344 DF,  p-value: < 2.2e-16
```

In this new model, the coefficients represent similar but different things. The coefficients now represent the price in house after changing each predictor by 1 unit while holding the other variables constant.

Now, the coefficients are now much less clear. The beds coefficient changed significantly – the sign and magnitude changed! This is odd at first because initially we saw that an increase in beds was linked to an increase in price, not a decrease. However, knowing more about multilinearregression gives us an explanation. Beds is highly correlated with the other variables so each coefficient actually means less in the real world application – it's hard to actually increase beds without increasing sq ft for example. This doesn't mean the model is inaccurate, it just shows that colinearity has an effect on the interpretation of our coefficients. so yes, it does make sense that it changed. In the original LM, we could consider beds to be a 'proxy' or 'representation' of the other factors that were affecting the price. It is possible that: an increase in bed count is linked to an increase in baths and sqft, which are positively associated with price.

yippee!! all done!!