STA 106 Project 2: M. Pouokam

Dylan M Ang

5/28/2022

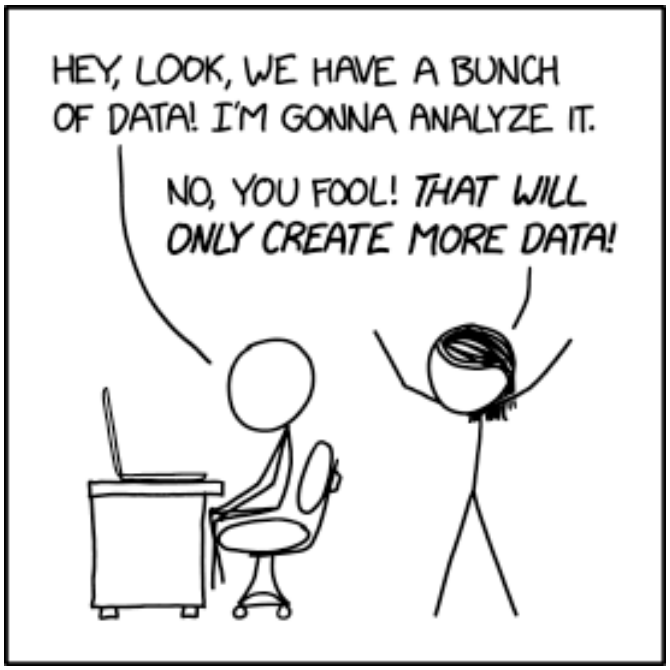


Figure 1: It's important to make sure your analysis destroys as much information as it produces. Credit: xkcd

Contents

Data Transformation

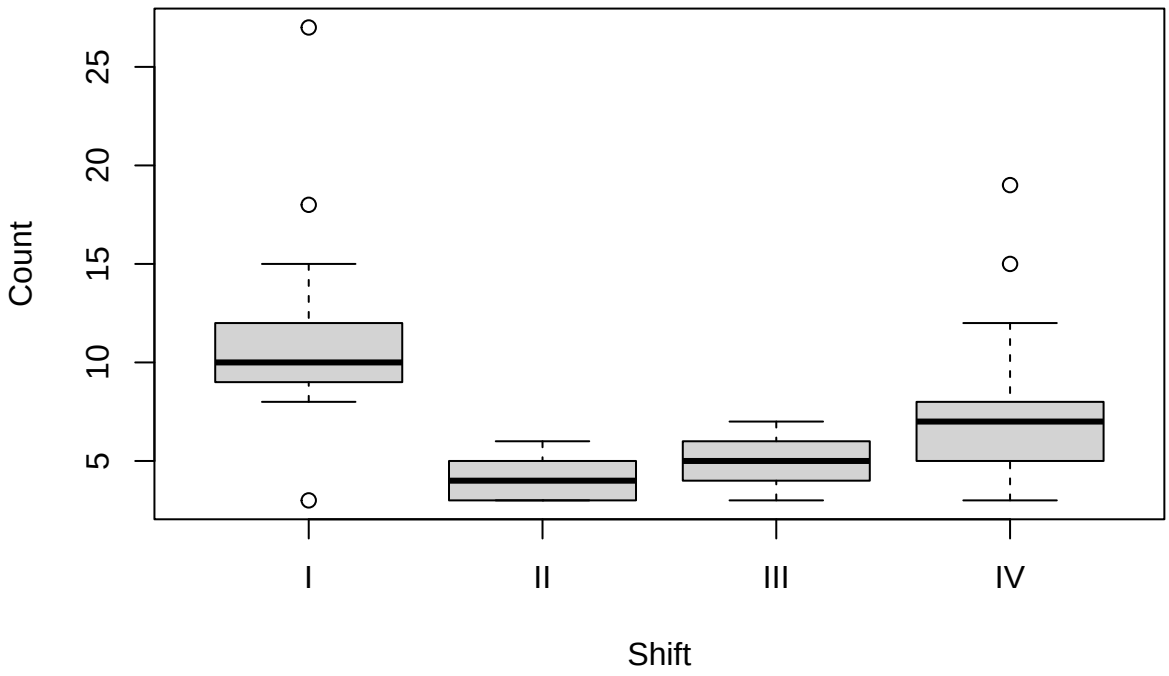
Introduction

The Helicopter data set tracks the number of helicopter requests for a sheriff's office and the time of day for 1 year. This project will examine the fitness of this data set to the ANOVA assumptions of normality and constant variance. In addition, this project will consider 3 types of data manipulation - removing outliers, transforming the dependent variable, and both - to see which new data set adheres to ANOVA assumptions better.

Original Data

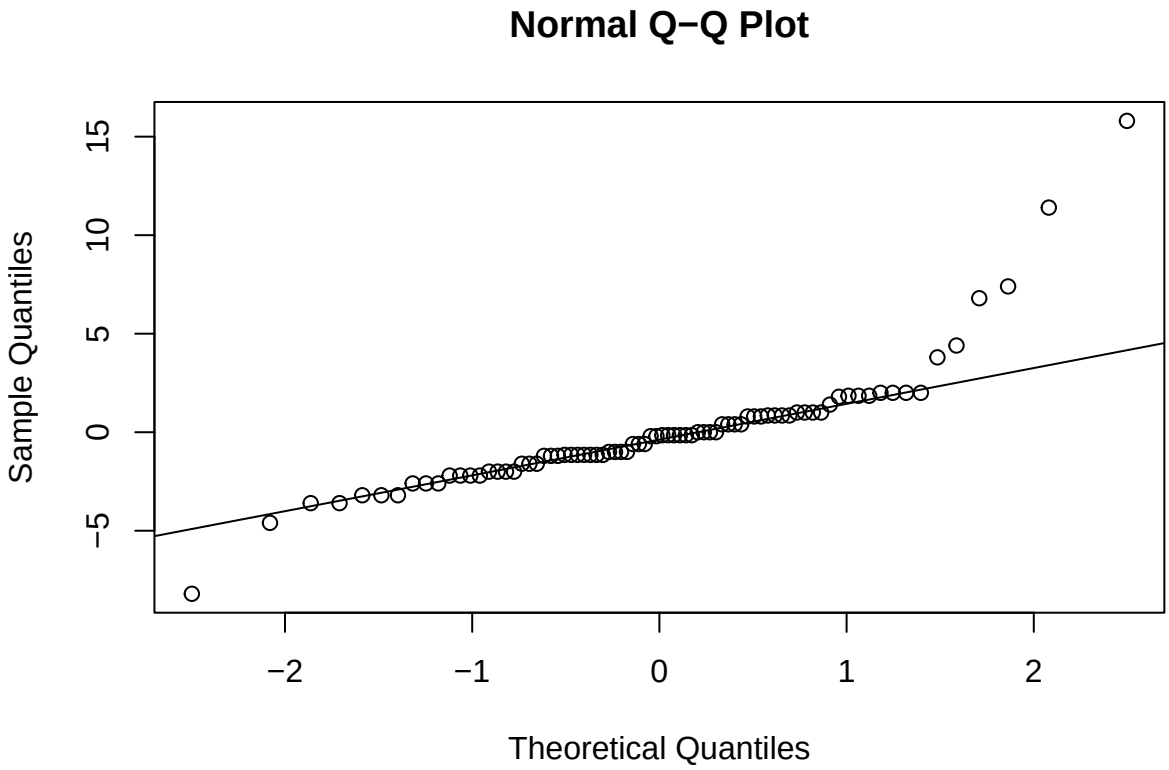
Table 1: Helicopter Data

Shift	Mean Count
I	11.20
II	4.15
III	5.00
IV	7.60



Based on the box plot, the data appears to have a few outliers on Shift I and IV. We will check the outliers with the standardized residuals in Section 3.

Assessing Normality



The QQ plot does not appear to be linear, we can use a Shapiro-Wilk test to be sure.

H_0 : The data is normally distributed.

H_A : The data is not normally distributed.

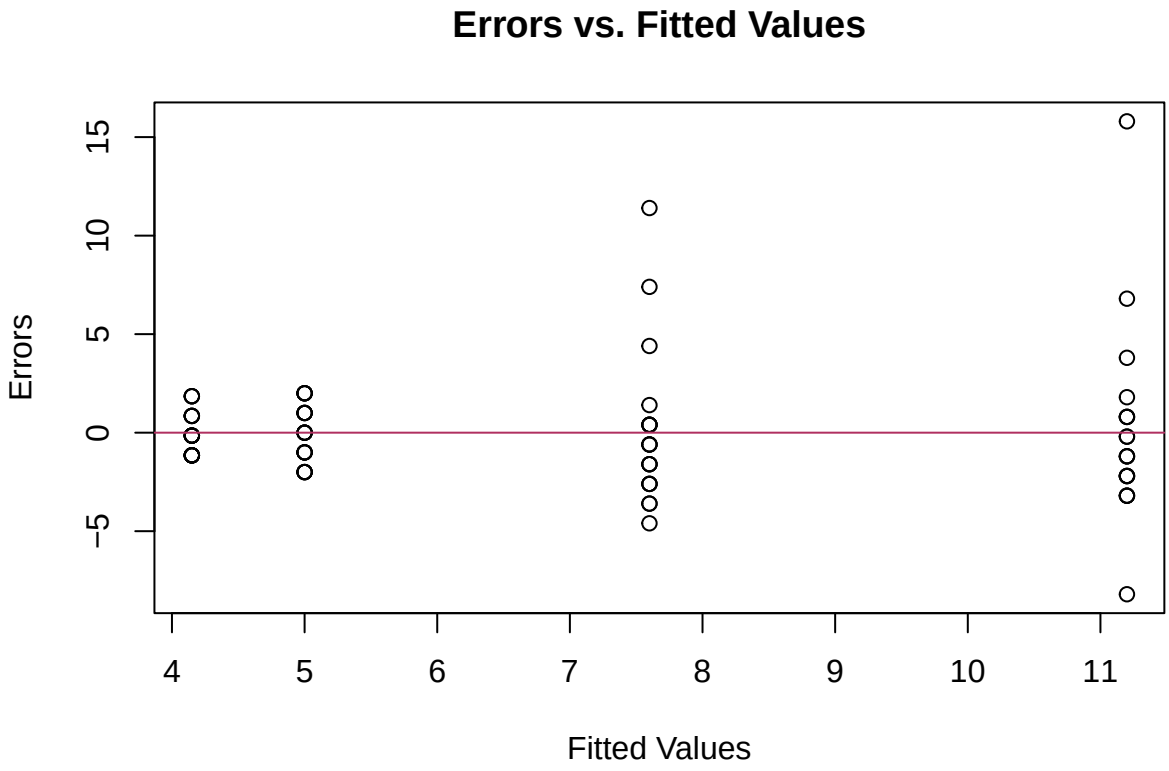
Table 2: Shapiro-Wilk's Test

W (test statistic)	0.8070844
p-value	0.0000000

$p < \alpha$ for any common significance level. Therefore we reject the null hypothesis and conclude that the data is not normally distributed.

Assessing Constant Variance

If the errors have constant variance, we would see that the range of the errors in an Errors vs. Fitted Values plot would be roughly uniform.



We see in the plot that the range of the data increases as the mean increases. Therefore, the plot suggests that this data set does not have constant variance. We can conduct a Brown-Forsythe test to be sure.

$H_0 : \sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \sigma_4^2$

$H_A : \neg(\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \sigma_4^2)$

In words, this test will determine whether the variances of the data equal each other, or if at least one of the variances is not equal.

Table 3: BF Test

test-statistic	19.50259
p-value	0.00000

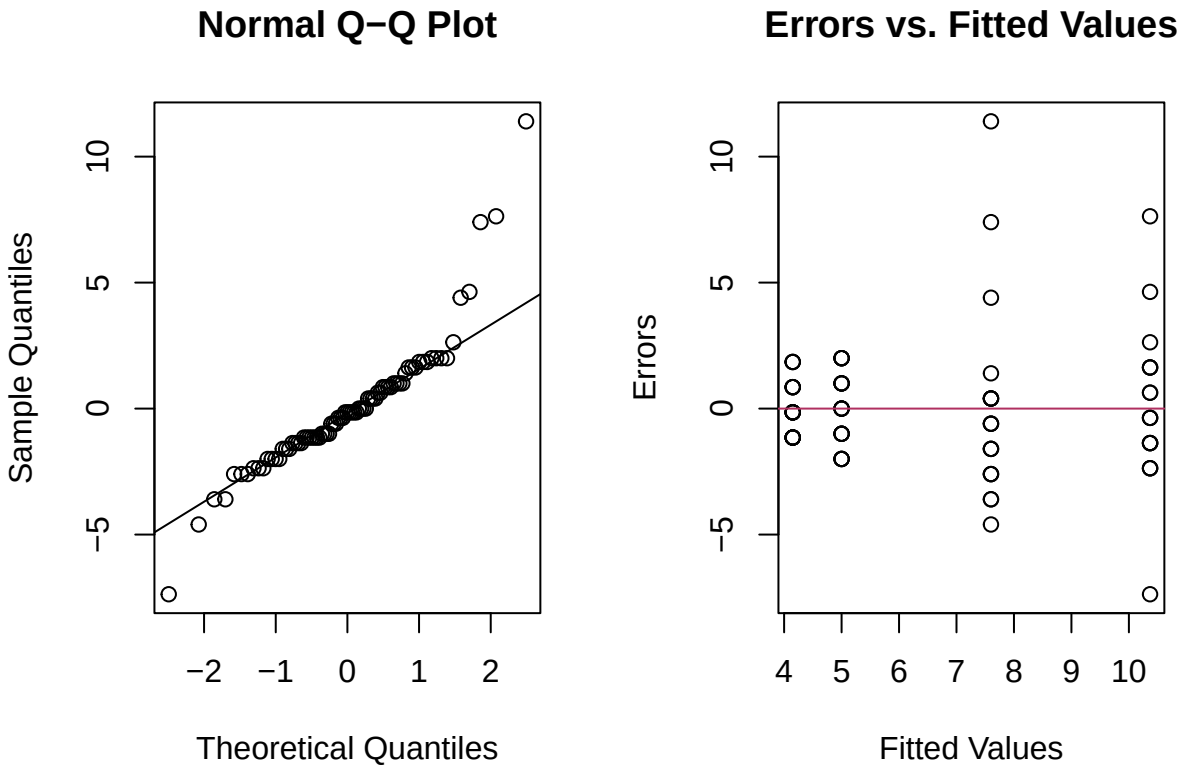
$p < \alpha$ for any common significance level, therefore we reject the null hypothesis. In conclusion, the data does not have constant variance between groups.

Transformations

Removing Outliers

Outliers will be removed by comparing the standardized residuals to a t cutoff value. Standardized residuals will be used as this method does not require the assumption of constant variance.

There was only one outlier removed, row 15 with Count: 27 and Shift: I. Now, we can check our diagnostic plots to get an idea of how effective the outlier removal was.



We see that the Errors vs. Fitted Values plot is slightly tighter, but ultimately didn't impact the spread very much. It still appears that the data is not normal nor does it have constant variance, we can check with our diagnostic tests.

Table 4: Diagnostic Tests

	Shapiro-Wilk	Brown-Forsythe
test statistic	0.8760140	21.95668
p value	0.0000015	0.00000

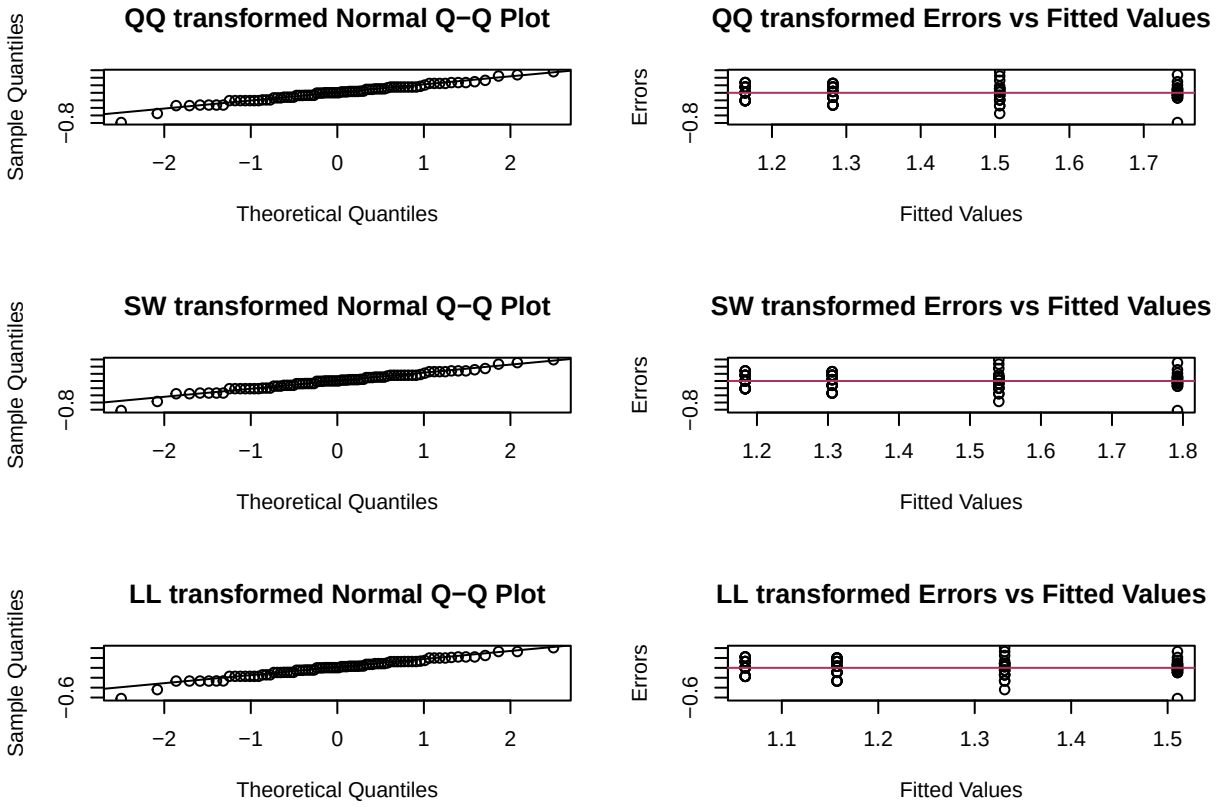
Both our Shapiro-Wilk's test and Brown-Forsythe test have very small p-values, less than any common alpha value. Therefore, we reject the null hypothesis in both cases and conclude that the data is not normal (Shapiro-Wilk) and does not have constant variance (Brown-Forsythe)

Transforming Y

Table 5: Transformation Results

	Lambda
Probability Plot	-0.2568296
Shapiro-Wilk	-0.2322012
Log-Likelihood	-0.3964138

Optimizing using the Q-Q plot, Shapiro-Wilk, and Log-Likelihood approaches results in different optimal lambda values. The Shapiro-Wilk method results in the greatest overall lambda value, followed by the Q-Q plot and then Log-Likelihood. We can check the diagnostic results of each to see which optimization method we should choose.



All three transformations seem to have improved the normality and constant variance of the data. However, from the Errors vs. Fitted Values it doesn't appear that the data has constant variance. It is still the case that the error appears to increase as the mean increases. To confirm our suspicions, we will perform a Shapiro-Wilk test for normality and a Brown-Forsythe test for constant variance.

Table 6: Shapiro-Wilk Test Results for Transformed Data

	Probability Plot	Shapiro-Wilk	Log-Likelihood
Test Statistics	0.9743466	0.9744398	0.9711696
p values	0.1075406	0.1090244	0.0673883

For a Shapiro-Wilk test, H_0 : The data is normally distributed and H_A : The data is not normally distributed. The Probability Plot and Shapiro-Wilk transformations have p-values that are insignificant at common significance levels (90, meaning they can be considered normally distributed. The Log-Likelihood transformation has a p-value of 0.067, which is insignificant at the 95 significance level but significant at the 90 significance level, therefore it may be considered normally distributed but less normally distributed than the other transformed data sets. Now we will use the Brown-Forsythe test to assess constant variance.

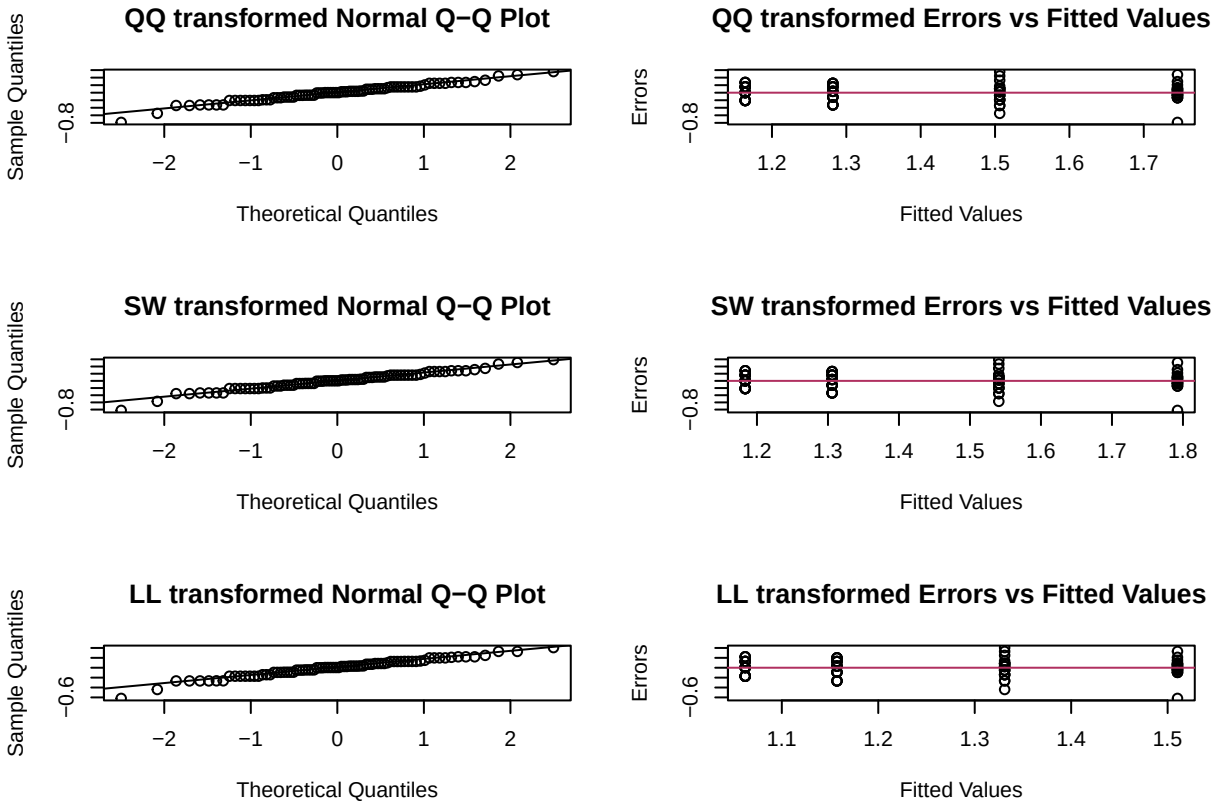
Table 7: Brown-Forsythe Test Results for Transformed Data

	Probability Plot	Shapiro-Wilk	Log-Likelihood
Test Statistics	25.77492	25.87002	25.11209
p values	0.00000	0.00000	0.00000

For a BF test, H_0 : The data has constant variance between groups and H_A : The data does not have constant variance between groups. All three transformed data sets have a p-value of 0 or very small (essentially 0), and therefore we can conclude that the data does not have constant variance between groups.

Both

Now, we will examine the assumptions of ANOVA for a data set which has had both outliers removed and transformations applied. This data set was created by applying the three transformations above to the data set found in the first part of this section.



The diagnostic plots above look similar to those of the outlier-remaining transformed data sets. We can use our empirical tests to see if the adherence to ANOVA assumptions improved at all by removing the outlier.

Table 8: Transformation Results (Outliers Removed)

	Lambda
Probability Plot	-0.1732219
Shapiro-Wilk	-0.0279367
Log-Likelihood	-0.2991460

Table 9: Shapiro-Wilk Test Results for Transformed Data (Outliers Removed)

	Probability Plot	Shapiro-Wilk	Log-Likelihood
Test Statistics	0.9738590	0.9687875	0.9737919
p values	0.1001033	0.0475116	0.0991198

Shapiro-Wilk tests measure normality, and we want to see high p-values, as they represent the probability of seeing our results if the data is normally distributed. We want this number to be as close to 1 as possible. For this data, the PPCC transformation has the highest p-value at > 0.1, meaning it is insignificant at the 3 common significance levels, so in that sense it is the "most" normal of the three transformations. The Log-Likelihood transformation has the next highest p-value at 0.099, which is just under 0.1. Then, the Shapiro-Wilk transformation has the lowest p-value at 0.0475, which is significant at $\alpha = 0.05$. We can say that both the Shapiro-Wilk transformed data and the Log-Likelihood data are not normally distributed with 90% confidence.

Table 10: Brown-Forsythe Test Results for Transformed Data (Outliers Removed)

	Probability Plot	Shapiro-Wilk	Log-Likelihood
Test Statistics	26.06773	26.35497	25.59535
p values	0.00000	0.00000	0.00000

A Brown-Forsythe test measures whether the data has pulled data from groups with constant variance, this helps measure if the different groups come from the same population. For a BF test, we want to see high p-values, as that tells us we likely have constant variance and can continue with our analysis. In this case, we see that all 3 p-values are very low or zero, and thus are all less than α at any common α level. Unfortunately this means that the data does not have constant variance, and any further analysis on this data may be misleading.

Best transformation

None of the transformations succeeded in making the data have constant variance, but the Box Cox transformations did make the data adhere better to a normal distribution. In particular, the Shapiro-Wilk optimized Box-Cox transformation on the data set without outliers removed had the highest p-value at 0.109. Therefore, we can say that this data is quantitatively the likely best data set to use, since it is the most normal.

Conclusion

Of the three manipulations that we tried - removing outliers, transforming Y, both - the most successful was transforming Y. In particular, using a Box-Cox transformation optimizing for Shapiro-Wilk scores resulted in the data set that best adhered to a normal distribution. There was only one outlier out of 80 data points, so removing it would mean removing 1.25 of the data, which isn't that bad, but removing the outlier did little to improve the normality or constant variance of the data, and doing both worked but not as well as just transforming Y. So ultimately, removing the outliers would not be worth the loss of the extra data point. Also, considering the shape of the data points in the Errors vs. Fitted Values plot, this data has increasing variance as the mean increases, so it is likely that this point is not actually an outlier. We would have to gather more data to be sure of that however.

One risk of using this new data set is that it is transformed, which means that interpretation will be more difficult. In addition, the data does not have constant variance, which may lead to uncertainty in the accuracy

of any future results with this data. However, considering that no other transformation had significant evidence of constant variance, this is still the best form for the data as it significantly improves the normality.

Two Factor ANOVA

Introduction

This report will be working with a data set containing salary information for tech employees working from either San Francisco or Seattle to determine what combinations of factors lead to changes in one's salary. Further, we will be examining the differences in salary by region for each of these types of tech employee - Data Scientist, Software Engineer, and Bioinformatics Engineer. The goal is to inform tech employees of potential salary increases or decreases by living in one region over another.

For this analysis, we will be fitting our data to a two factor ANOVA model. The factors we are considering are Profession (Factor A) and Region (Factor B) and their effect on Annual Salary. In addition, we will also consider any potential interaction effects, where combinations of these two factors lead to additional changes to Annual Salary. F-tests will be used to measure the significance of any interaction or factor effects.

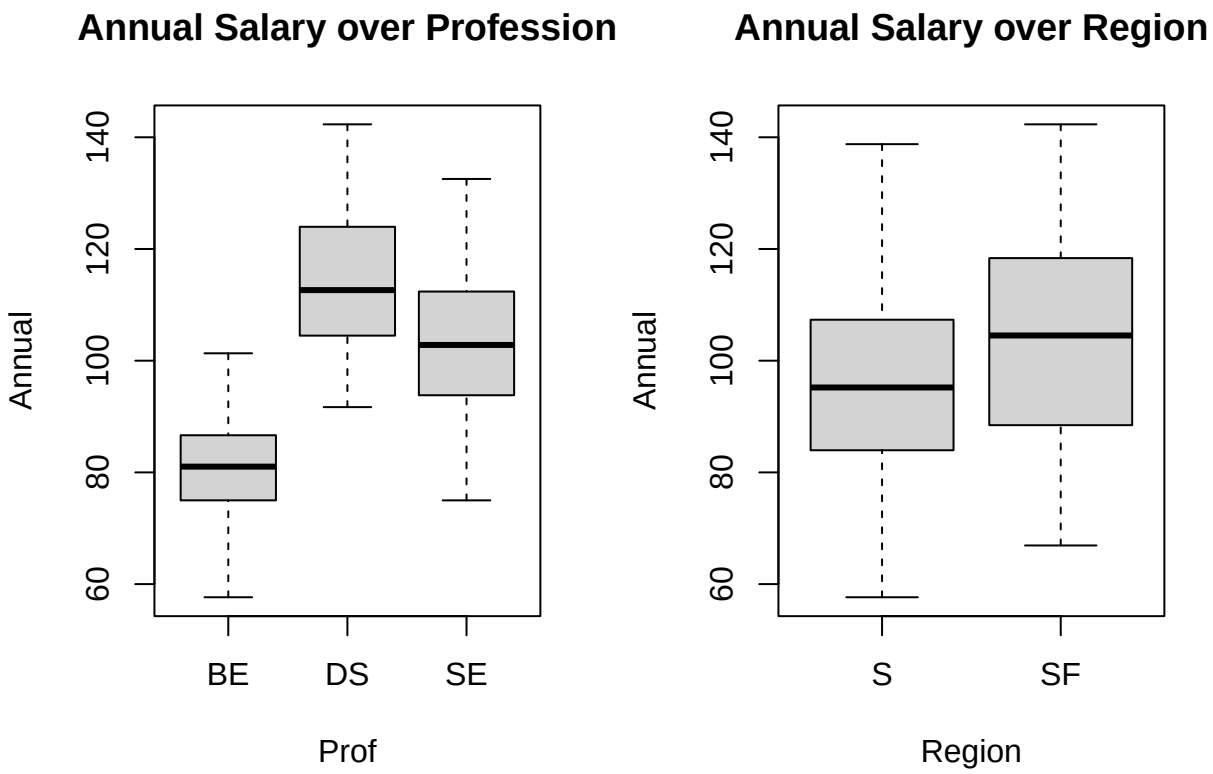
To ensure that the model conforms to the assumptions of our analysis (Two Factor ANOVA), we will make diagnostic plots (Q-Q and Errors vs. Fitted Values) and conduct diagnostic tests (Shapiro-Wilk and Brown-Forsythe). Transformations and outliers will be considered, but will not affect our analysis in a significant way, as this is outside of the scope of this report.

Summary of Data

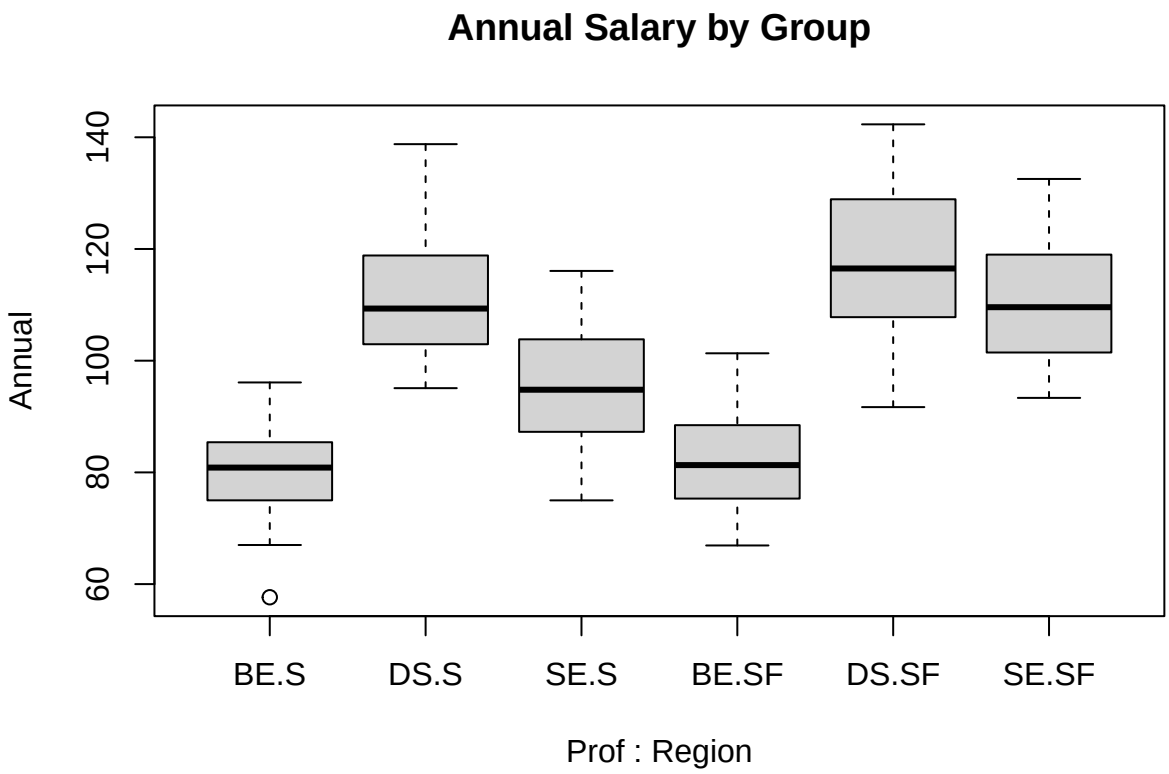
Table 11: Mean Annual Salary (in thousands) by Job Title and Region

	BE	DS	SE
San Francisco	82.41914	117.7688	110.26412
Seattle	79.75485	112.5272	95.54875

Note: BE represents Bioinformatics Engineer, DS represents Data Scientist, and SE represents Software Engineer.



Based on the Annual Salary over Region plot, there is not a lot of variation in salary between Seattle and San Francisco. There is a lot more variation in the Salary by Profession. As a general trend, we see that Bioinformatics Engineers are paid the least and Data Scientists are paid the most.

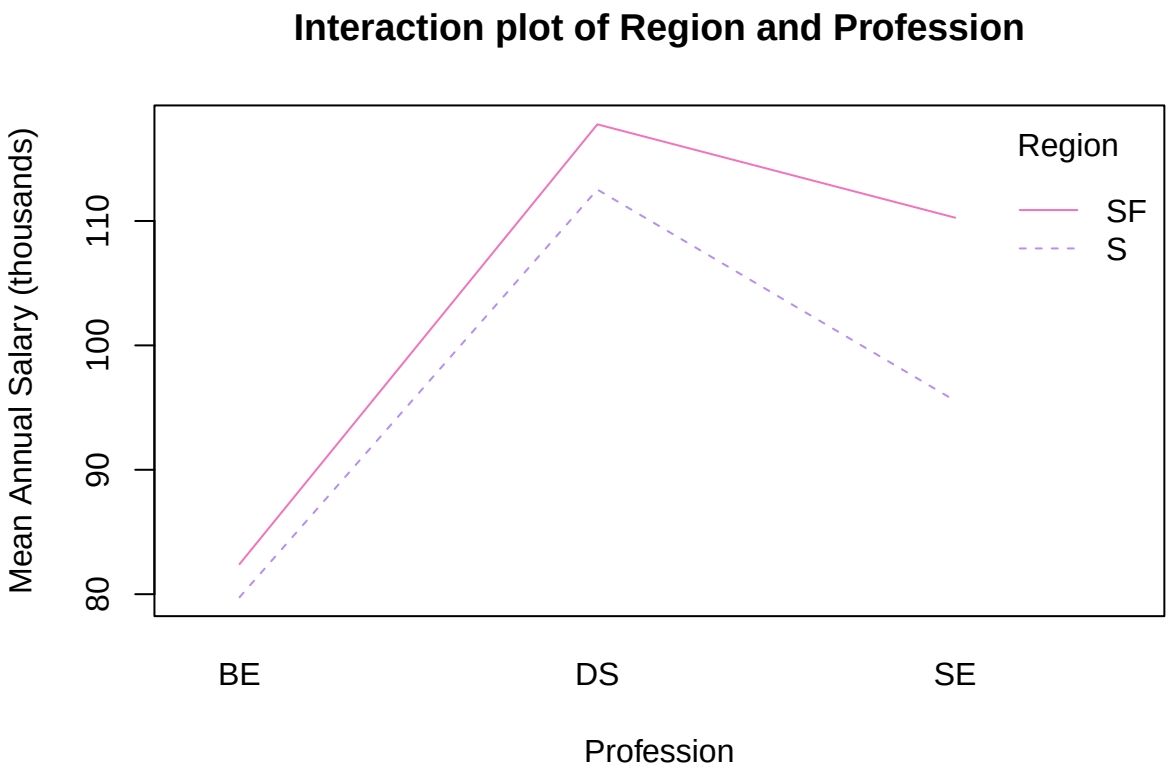


We can come to a similar conclusion with this box plot as the two plots for each factor. There is much more variation within the profession than within the region. This suggests that profession is a more significant factor in determining annual salary than region.

Interaction and Factor Effect Testing

Test for Interactions

In this section of the report, we will be examining the Effect of each factor and their interactions to reach a verdict on the best model to use for two factor anova.



We see from the interaction plot that the lines are not perfectly parallel. In general, salaries in SF are higher than Seattle, but particularly Software Engineers in SF see a higher proportional increase in salary over their Seattle counterparts than any other profession. This suggests that their may be an interaction effect to being a software engineer in San Francisco. We can be sure by performing an F test for Interactions

For an F test for Interactions with $\alpha = 0.05$.

Full Model:

$$Y_{ijk} = \mu_{..} + \gamma_i + \delta_j + (\gamma\delta)_{ij} + \epsilon_{ijk} \quad df\{SSE\} = n_T - ab$$

Reduced Model:

$$Y_{ijk} = \mu_{..} + \gamma_i + \delta_j + \epsilon_{ijk} \quad df\{SSE\} = n_T - a - b + 1$$

H_0 : All $(\gamma\delta)_{ij} = 0$

H_A : At least one $(\gamma\delta)_{ij} \neq 0$

Our test statistic is,

$$F_s = \left[\frac{SSE_R - SSE_F}{df_R - df_F}\right] \Big/ MSE_F \tag{1}$$

Table 12: Interaction Test Results

F-stat	3.0097976
p	0.0532358

$p < \alpha \implies$ Fail to reject the null. In conclusion, there is not a significant interaction effect on the mean annual salary. Therefore going forward we will be using a model with interaction.

Test for Factor A (Profession)

To test for the effect of Factor A, we will be comparing the model with Factor A and B, and the model with only Factor B.

Full Model:

$$Y_{ijk} = \mu_{..} + \gamma_i + \delta_j + \epsilon_{ijk} \quad df\{SSE\} = n_T - a - b + 1$$

Reduced Model:

$$Y_{ijk} = \mu_{..} + \delta_j + \epsilon_{ijk} \quad df\{SSE\} = n_T - b$$

$H_0 : \gamma_i = 0 \forall i$

H_A : At least one $\gamma_i \neq 0$

The test statistic is the same as before (1).

Table 13: Factor A test results

F-stat	86.0143
p-value	0.0000

$p < \alpha$ for $\alpha = 0.05$ or any common significance level. We should reject the null hypothesis and conclude that Factor A is significant. In other words, Profession has a significant effect on mean annual salary.

Test for Factor B (Region)

To test for the effect of Factor B, we will be comparing the model with Factor A and B, and the model with only Factor A.

Full Model:

$$Y_{ijk} = \mu_{..} + \gamma_i + \delta_j + \epsilon_{ijk} \quad df\{SSE\} = n_T - a - b + 1$$

Reduced Model:

$$Y_{ijk} = \mu_{..} + \gamma_i + \epsilon_{ijk} \quad df\{SSE\} = n_T - b$$

$H_0 : \delta_j = 0 \forall j$

H_A : At least one $\delta_j \neq 0$

The test statistic is the same as before (1).

Table 14: Factor A test results

F-stat	12.3217684
p-value	0.0006385

$p < \alpha$ for any common significance level, therefore we reject the null and conclude that Factor B is significant. In conclusion, Region has a significant effect on annual salary.

Section Conclusion

To summarize our results above, at a 95 significance level,

- Factor AB Interaction has no effect on the mean.
- Factor A has an effect on the mean.
- Factor B has an effect on the mean.

Based on our results, we will be using the model with Factor A (Profession) and Factor B (Region) but with no interaction term.

$$Y_{ijk} = \mu_{..} + \gamma_i + \delta_j + \epsilon_{ijk} \quad df\{SSE\} = n_T - a - b + 1$$

The partitioned variance values can be seen below.

Table 15: (A+B) ANOVA Model Summary

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Prof	2	23814.600	11907.300	86.01430	0.0000000
Region	1	1705.751	1705.751	12.32177	0.0006385
Residuals	116	16058.340	138.434	NA	NA

$\gamma_i = [-18.6268119, 15.4341832, 3.1926286]$ where $i = [BE, DS, SE]$

$\delta_j = [-3.7702246, 3.7702246]$ where $j = [S, SF]$

Now that we have settled on a model, we must review our ANOVA assumptions.

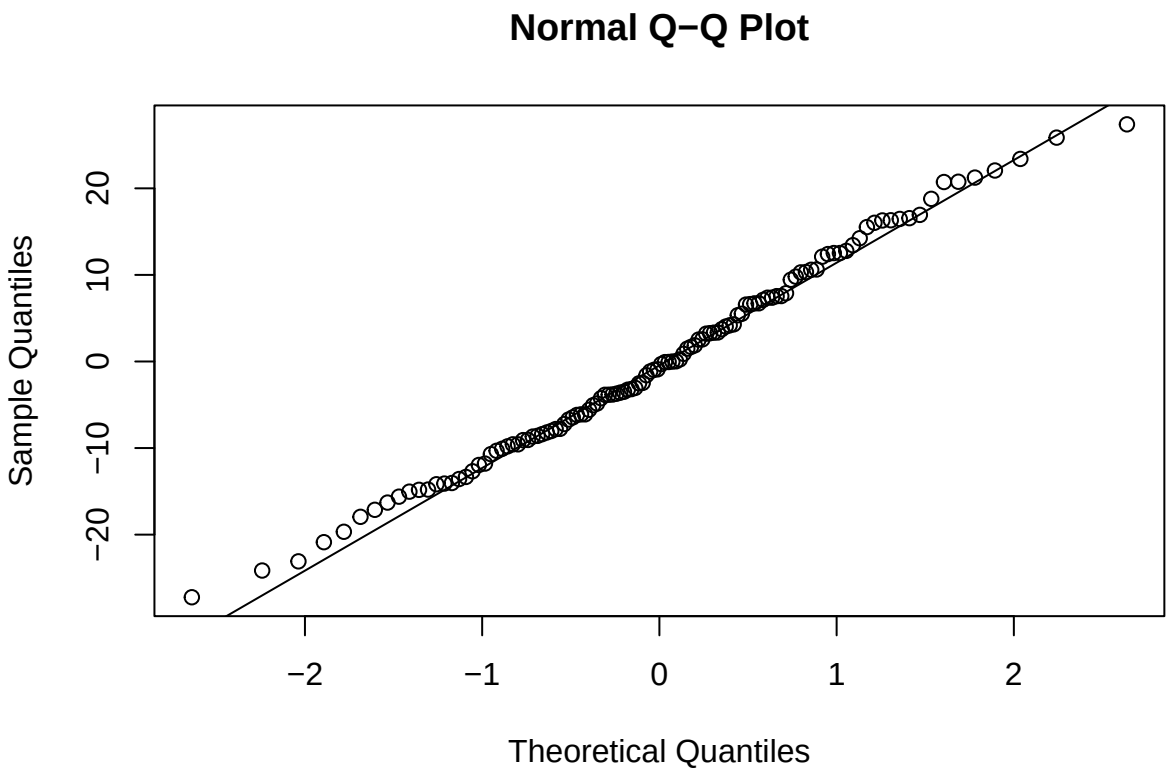
Diagnostics

In this section we will be testing for two assumptions we need for variance analysis, normality and constant variance. We will start by checking our data for outliers by comparing the standardized residuals to a t cutoff.

Fortunately, there are no outliers in this data set, so we can proceed with out analysis without removing any data points.

Normality

We can use a Q-Q plot to get an idea of the normality of the data. Q-Q plots measure how the well the Sample Quantiles match up to Theoretical Quantiles. If the data is normally distributed, we expect to see data points in a linear pattern.



In this case, we see that there is a little deviation from the line near the end points, but overall there is strong adherence to the line. This plot suggests that the data is sufficiently normally distributed. To be sure, we will perform a Shapiro-Wilk test.

A Shapiro-Wilk test measures the normality of the data.

H_0 : The data is normally distributed.

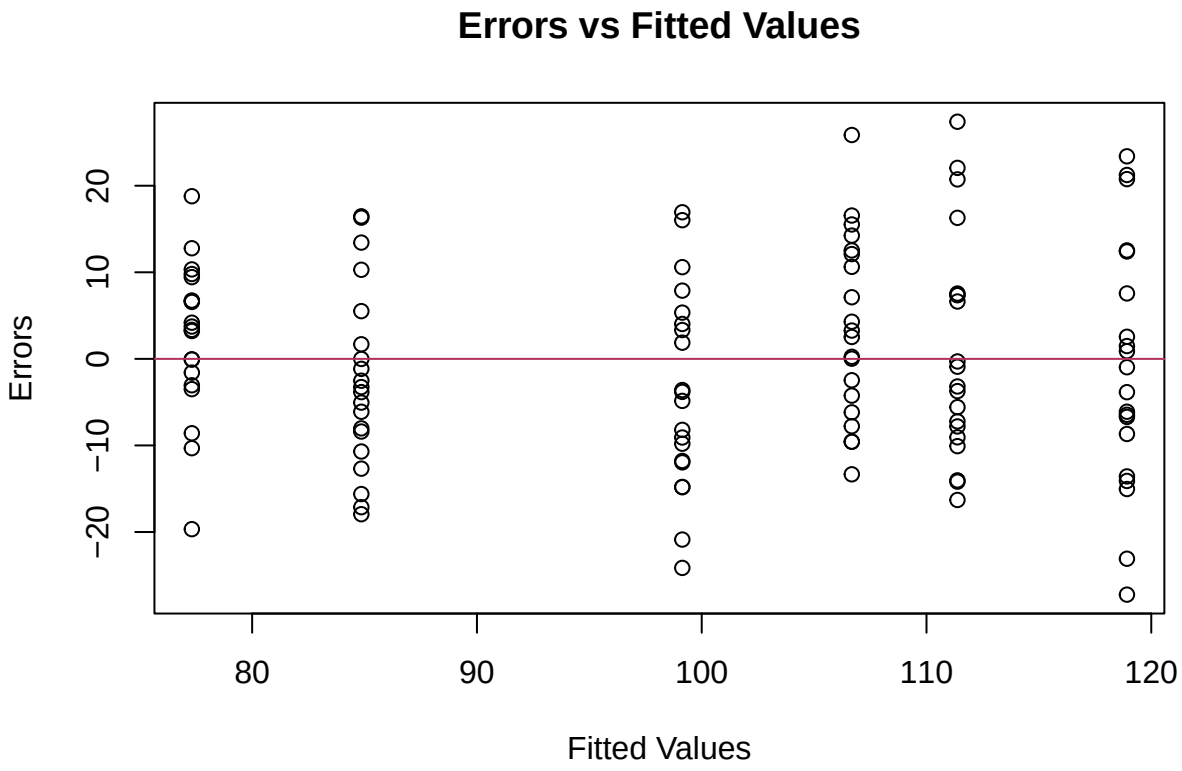
H_A : The data is not normally distributed.

Table 16: Shapiro-Wilk's Test

W (test statistic)	0.9914583
p-value	0.6697801

Our test has a high p-value greater than α at any common significance level, therefore we fail to reject the null hypothesis. We conclude that the data is normally distributed and it is safe to continue with our analysis.

Constant Variance



The errors and variance plot shows that the range of errors is roughly uniform. There isn't any specific patterns to the data, so the plot suggests there is constant variance, although we will use a Brown-Forsythe test to be sure.

H_0 : The data has constant variance.

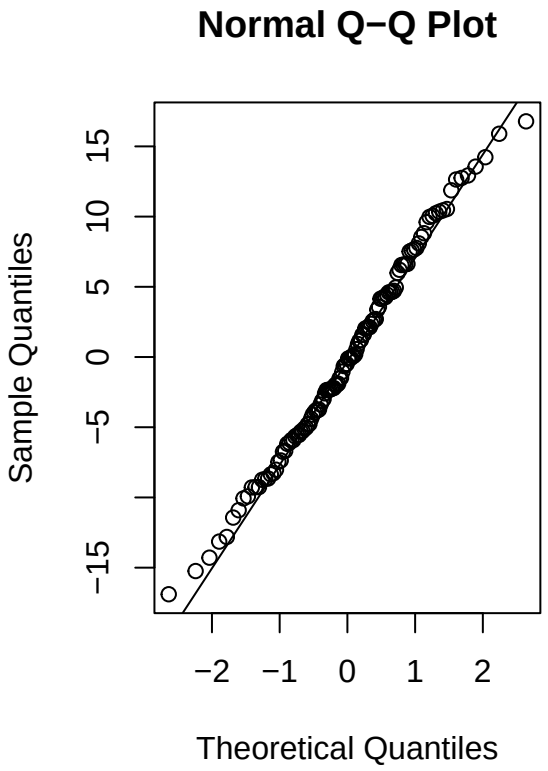
H_A : The data does not have constant variance.

	test-statistic	p-value
Profession	78.425297	0.0000000
Region	5.048001	0.0265336

The p-value for constant variance within Factor A (Profession) is 0, so at any significance level we reject the null hypothesis and conclude that there is not constant variance between the levels of Factor A.

The p-value for constant variance within Factor B (Region) is $> \alpha$ at a 99 significance level, therefore we fail to reject the null and conclude that Factor B has constant variance.

Since we lack constant variance for both Factors, our data does not satisfy assumptions required for ANOVA. To see if we can achieve constant variance, we will apply a box cox transformation optimizing for Shapiro-Wilk scores.



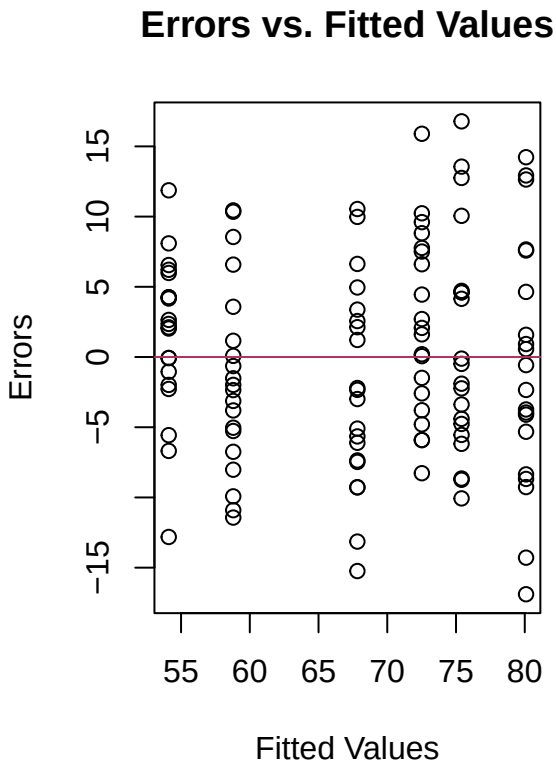


Table 18: BF test results for transformed data

	test-statistic	p-value
Profession	79.26434	0.0000000
Region	5.01266	0.0270555

Based on the results of the bf test, our constant variance did not improve in any meaningful way. Since transformations make our interpretation more difficult, and there is no benefit in this case, we will continue our analysis with our untransformed data.

Analysis

We already know that both Factor A and B have a significant effect on mean annual salary and will be using the $(A + B)$ model, shown below.

$$Y_{ijk} = \mu_{..} + \gamma_i + \delta_j + \epsilon_{ijk} \tag{2}$$

Since we have already tested for Factor Effects, we will just find confidence intervals in this section.

The average salaries in SF are all higher than their counterparts in Seattle, so we will find the difference in salary with a pairwise confidence interval.

$$\mu_{SF} - \mu_S$$

If we do find a significant salary difference between tech employees working in San Francisco vs Seattle, it would make sense that some employees would want to move. To see if it would be "worth it" to move to San Francisco, we should test to see if the salary increases are significant with the following intervals.

$$\mu_{SE.SF} - \mu_{SE.S}$$

$$\mu_{DS.SF} - \mu_{DS.S}$$

$$\mu_{BE.SF} - \mu_{BE.S}$$

Since Software Engineers are in the middle, we will compare the salaries of software engineers to the average salary of data scientists and bioinformatics engineers.

$$\mu_{SE} - \frac{\mu_{DS} + \mu_{BE}}{2}$$

We also consider a couple, one is a software engineer and the other is a data scientist. Our confidence interval will examine the difference in the amount of money that this couple could make working out of San Francisco vs working out of Seattle.

$$(\mu_{DS.SF} + \mu_{SE.SF}) - (\mu_{DS.S} + \mu_{SE.S})$$

In summary, we will be examining 6 simultaneous confidence intervals.

- $\mu_{SF} - \mu_S$
- $\mu_{SE.SF} - \mu_{SE.S}$
- $\mu_{DS.SF} - \mu_{DS.S}$
- $\mu_{BE.SF} - \mu_{BE.S}$
- $\mu_{SE} - \frac{\mu_{DS} + \mu_{BE}}{2}$
- $(\mu_{DS,SF} + \mu_{SE,SF}) - (\mu_{DS,S} + \mu_{SE,S})$

We will be considering 3 types of corrections: Tukey, Bonferroni, and Scheffe. We want to find the smallest intervals, as they allow us to be more accurate in our interpretation. The three correction methods have the same format except for their multiplier, and the smallest interval results from the smallest multiplier. Before finding the intervals, we will find the smallest multiplier and use the smallest one to create the intervals.

Table 19: Simultaneous CI Correction Multipliers

	Multiplier
Tukey	2.898
Bonferroni	2.684
Scheffe	3.386

The Bonferroni correction results in the smallest multiplier, and will therefore result in the tightest intervals, therefore we will continue with the Bonferroni correction.

Table 20: 95% Bonferroni Corrected Confidence Intervals

	Labels	Estimate	lower.bound	upper.bound
I	SF - S	7.540449	1.774866	13.30603
II	SE.SF - SE.S	14.715373	4.729091	24.70166
III	DS.SF - DS.S	5.241682	-4.744600	15.22796
IV	BE.SF - BE.S	2.664292	-7.321990	12.65057
V	(BE.SF + SE,SF) - (BE.S + SE.S)	17.379665	3.256930	31.50240

	Labels	Estimate	lower.bound	upper.bound
VI	(DS.SF + SE.SF) - (DS.S + SE.S)	19.957056	5.834320	34.07979

We will examine these intervals in-depth in the next section.

Interpretation

I: $\mu_{SF} - \mu_S$

With 95% overall confidence, the difference in average salary between tech employees in San Francisco and in Seattle, regardless of Profession, is between roughly $\$1,774$ and $\$13,306$ annually.

II: $\mu_{SE.SF} - \mu_{SE.S}$

With 95% overall confidence, the difference in average salary for a Software Engineer in San Francisco and in Seattle is between roughly $\$4,729$ and $\$24,702$ annually.

III: $\mu_{DS.SF} - \mu_{DS.S}$

With 95% overall confidence, the difference in average salary for a Data Scientist in San Francisco and in Seattle is between roughly $-\$4,745$ and $\$15,228$ annually.

IV: $\mu_{BE.SF} - \mu_{BE.S}$

With 95% overall confidence, the difference in average salary for a Bioinformatics Engineers in San Francisco and in Seattle is between roughly $-\$7,322$ and $\$12,651$ annually.

V: $(\mu_{BE,SF} + \mu_{SE,SF}) - (\mu_{BE,S} + \mu_{SE,S})$

With 95% overall confidence, the difference in average combined salary of a Bioinformatics Engineer and a Software Engineer working out of San Francisco vs Seattle is between roughly $\$3,257$ and $\$31,502$ annually.

VI: $(\mu_{DS,SF} + \mu_{SE,SF}) - (\mu_{DS,S} + \mu_{SE,S})$

With 95% overall confidence, the difference in average combined salary of a Data Scientist and a Software Engineer working out of San Francisco vs Seattle is between roughly $\$5,834$ and $\$34,080$ annually.

Conclusion

While the specific values can be found elsewhere in the report, we will summarize our general findings here. Overall, regardless of your profession, salaries in San Francisco are generally higher than salaries in Seattle. A working Software Engineer can make on average between roughly $\$5,000$ $\$25,000$ more by working in San Francisco versus Seattle. However, Data Scientists and Bioinformatics Engineers can make more or less the same, no matter where they live. Data Scientists and Bioinformatics Engineers may not see a significant pay increase by working in San Francisco over Seattle.

However, many people are not only considering themselves in their move. Many people are married or in relationships, or otherwise live and share income with another person. We can see from the combined salary confidence intervals that with 95% overall confidence, a couple moving to San Francisco from Seattle will see a total pay increase, so long as at least one of the partners is a Software Engineer.

Appendix

```
knitr::opts_chunk$set(echo = FALSE)
suppressMessages(library(tidyverse, warn.conflicts = FALSE))
heli <- read.csv("../datasets/Helicopter.csv")

library(knitr)
mbg <- aggregate(Count ~ Shift, data = heli, mean)
colnames(mbg)[2] = "Mean Count"
kable(mbg, caption = "Helicopter Data")
boxplot(Count ~ Shift, data = heli)
fit <- lm(Count ~ Shift, data = heli)
qqnorm(fit$residuals)
qqline(fit$residuals)
# Function to output shapiro wilk tests in a table
print_shapiro_wilk <- function(numeric_values) {
  shap <- shapiro.test(numeric_values)
  W = shap$statistic
  p = shap$p.value
  shap_res <- c(W,p, use.names=FALSE)
  shap_labels <- c("W (test statistic)", "p-value")
  kable(data.frame(shap_labels, shap_res), col.names = c("",""),
        caption="Shapiro-Wilk's Test")
}

print_shapiro_wilk(heli$Count)
plot(fit$residuals ~ fit$fitted.values,
     main="Errors vs. Fitted Values",
     xlab="Fitted Values", ylab="Errors")
abline(h=0, col="maroon")
library(onewaytests)
res <- bf.test(Count ~ Shift, data=heli, alpha=0.01, verbose=FALSE)
ret <- c(res$statistic, res$p.value)
names(ret) <- c("test-statistic", "p-value")
kable(ret, col.names="", caption = "BF Test")
nt = nrow(heli)
a = nrow(mbg)
alpha = 0.01
t.cutoff = qt(1 - alpha/(2*nt), nt - a)

# Studentized DOES NOT ASSUME CONST VAR
rij <- rstandard(fit)
CO.rij <- which(abs(rij) > t.cutoff)

# Remove outliers
outliers = CO.rij
nout_heli <- heli[-outliers,]
nout_fit <- lm(Count ~ Shift, data = nout_heli)
print_diag_plots <- function(fit) {
  qqnorm(fit$residuals)
  qqline(fit$residuals)

  plot(fit$residuals ~ fit$fitted.values,
```

```
      main="Errors vs. Fitted Values",
      xlab="Fitted Values", ylab="Errors")
  abline(h=0, col="maroon")
}

par(mfrow = c(1,2))
print_diag_plots(nout_fit)
print_shapiro_wilk <- function(numeric_values, print = FALSE) {
  shap <- shapiro.test(numeric_values)
  W = shap$statistic
  p = shap$p.value
  shap_res <- c(W,p, use.names=FALSE)
  shap_labels <- c("W (test statistic)", "p-value")
  if (print == FALSE) {return(data.frame(shap_labels, shap_res))}
  kable(data.frame(shap_labels, shap_res), col.names = c("",""),
       caption="Shapiro-Wilk's Test")

}
print_bf_test <- function(formula, data, alpha, print = FALSE) {
  res <- bf.test(formula, data=data, alpha=alpha, verbose=FALSE)
  ret <- c(res$statistic, res$p.value)
  names(ret) <- c("test-statistic", "p-value")
  if (print == FALSE) {return(ret)}
  kable(ret, col.names="", caption = "BF Test")
}

# par(mfrow = c(1,2))
shap <- print_shapiro_wilk(nout_fit$residuals, print=FALSE)
bf <- print_bf_test(Count ~ Shift, data=nout_heli, alpha=0.01, print=FALSE)

print_diag_tests <- function(shap, bf) {
  test_stats <- unname(c(shap[1,2], bf[1]))
  p_vals <- unname(c(shap[2,2], bf[2]))
  ret <- rbind(test_stats, p_vals)
  row.names(ret) <- c("test statistic", "p value")
  kable(ret, col.names = c("Shapiro-Wilk", "Brown-Forsythe"),
        caption="Diagnostic Tests")
}

print_diag_tests(shap, bf)
library(EnvStats, warn.conflicts = FALSE)
T1 <- boxcox(fit, objective.name = "PPCC", optimize = TRUE)
T2 <- boxcox(fit, objective.name = "Shapiro-Wilk", optimize = TRUE)
T3 <- boxcox(heli$Count, objective.name = "Log-Likelihood", optimize = TRUE)

transforms <- list(T1, T2, T3)

get_lambda <- function(transformation) {transformation$lambda}

lambdas <- transforms %>% map(get_lambda) %>% unlist()
names(lambdas) <- c("Probability Plot", "Shapiro-Wilk", "Log-Likelihood")

kable(lambdas, col.names = "Lambda", caption = "Transformation Results")
```

```r
apply_boxcox <- function(lambda) {
  count <- (heli$Count^(lambda) - 1)/lambda
  trans <- data.frame(Count = count, Shift = heli$Shift)
  return(trans)
}

transformed <- lambdas %>% map(apply_boxcox)

fits <- transformed %>% map(function (data) {lm(Count ~ Shift, data=data)})
print_diag_plots <- function(fit, qqcaption = "Normal Q-Q Plot",
                             varcaption = "Errors vs. Fitted Values") {
  qqnorm(fit$residuals, main=qqcaption)
  qqline(fit$residuals)

  plot(fit$residuals ~ fit$fitted.values,
      main=varcaption,
      xlab="Fitted Values", ylab="Errors")
  abline(h=0, col="maroon")
}

par(mfrow = c(3,2))
print_diag_plots(fits$`Probability Plot`,
    qqcaption="QQ transformed Normal Q-Q Plot",
    varcaption="QQ transformed Errors vs Fitted Values")


print_diag_plots(fits$`Shapiro-Wilk`,
    qqcaption="SW transformed Normal Q-Q Plot",
    varcaption="SW transformed Errors vs Fitted Values")

print_diag_plots(fits$`Log-Likelihood`,
    qqcaption="LL transformed Normal Q-Q Plot",
    varcaption="LL transformed Errors vs Fitted Values")
shaps <- fits %>% map(function(fit) {fit$residuals}) %>%
              map(print_shapiro_wilk) %>%
              map(function (shap) {shap$shap_res})

stat <- rep(c(TRUE, FALSE), 3)
sw_stats <- unlist(shaps)[stat]
sw_ps <- unlist(shaps)[!stat]

sw_results <- rbind(sw_stats, sw_ps)
row.names(sw_results) <- c("Test Statistics", "p values")
kable(sw_results, col.names = c("Probability Plot", "Shapiro-Wilk", "Log-Likelihood"),
      caption = "Shapiro-Wilk Test Results for Transformed Data")
bfs <- transformed %>% map(function(tr) {
  print_bf_test(Count ~ Shift, tr, 0.01, print = FALSE)
})
bf_stats <- unlist(bfs)[stat]
bf_ps <- unlist(bfs)[!stat]

bf_results <- rbind(bf_stats, bf_ps)
row.names(bf_results) <- c("Test Statistics", "p values")
```

```
kable(bf_results, col.names = c("Probability Plot", "Shapiro-Wilk", "Log-Likelihood"),
      caption = "Brown-Forsythe Test Results for Transformed Data")
par(mfrow = c(3,2))
print_diag_plots(fits$`Probability Plot`,
    qqcaption="QQ transformed Normal Q-Q Plot",
    varcaption="QQ transformed Errors vs Fitted Values")


print_diag_plots(fits$`Shapiro-Wilk`,
    qqcaption="SW transformed Normal Q-Q Plot",
    varcaption="SW transformed Errors vs Fitted Values")

print_diag_plots(fits$`Log-Likelihood`,
    qqcaption="LL transformed Normal Q-Q Plot",
    varcaption="LL transformed Errors vs Fitted Values")

T1 <- boxcox(nout_fit, objective.name = "PPCC", optimize = TRUE)
T2 <- boxcox(nout_fit, objective.name = "Shapiro-Wilk", optimize = TRUE)
T3 <- boxcox(nout_heli$Count, objective.name = "Log-Likelihood", optimize = TRUE)
transforms <- list(T1, T2, T3)

get_lambda <- function(transformation) {transformation$lambda}

lambdas <- transforms %>% map(get_lambda) %>% unlist()
names(lambdas) <- c("Probability Plot", "Shapiro-Wilk", "Log-Likelihood")

kable(lambdas, col.names = "Lambda",
      caption = "Transformation Results (Outliers Removed)")

transformed <- lambdas %>% map(apply_boxcox)

fits <- transformed %>% map(function (data) {lm(Count ~ Shift, data=data)})
shaps <- fits %>% map(function(fit) {fit$residuals}) %>%
              map(print_shapiro_wilk) %>%
              map(function (shap) {shap$shap_res})

stat <- rep(c(TRUE, FALSE), 3)
sw_stats <- unlist(shaps)[stat]
sw_ps <- unlist(shaps)[!stat]

sw_results <- rbind(sw_stats, sw_ps)
row.names(sw_results) <- c("Test Statistics", "p values")
kable(sw_results,
      col.names = c("Probability Plot", "Shapiro-Wilk", "Log-Likelihood"),
      caption = "Shapiro-Wilk Test Results for
      Transformed Data (Outliers Removed)")
bfs <- transformed %>% map(function(tr) {
  print_bf_test(Count ~ Shift, tr, 0.01, print = FALSE)
})
bf_stats <- unlist(bfs)[stat]
bf_ps <- unlist(bfs)[!stat]

bf_results <- rbind(bf_stats, bf_ps)
```

```r
row.names(bf_results) <- c("Test Statistics", "p values")
kable(bf_results,
      col.names = c("Probability Plot", "Shapiro-Wilk", "Log-Likelihood"),
      caption = "Brown-Forsythe Test Results for
      Transformed Data (Outliers Removed)")
rm(list = ls()) # remove previous variables
library(tidyverse, warn.conflicts = FALSE)
library(knitr)
library(onewaytests)
library(EnvStats, warn.conflicts = FALSE)
sals <- read.csv("../datasets/Salary.csv")
mean_by_prof <- aggregate(Annual ~ Prof, data=sals, mean)
mean_by_regn <- aggregate(Annual ~ Region, data=sals, mean)
mbg <- aggregate(Annual ~ Prof+Region, data=sals, mean)

# Make table
sf <- mbg[mbg$Region == "SF",]
profs <- sf$Prof
sf <- sf$Annual
names(sf) <- profs
sea <- mbg[mbg$Region == "S",]$Annual
table <- rbind(sf, sea)
row.names(table) <- c("San Francisco", "Seattle")
kable(table, caption="Mean Annual Salary (in thousands) by Job Title and Region")
par(mfrow = c(1,2))
boxplot(Annual ~ Prof, data=sals, main="Annual Salary over Profession")
boxplot(Annual ~ Region, data=sals, main="Annual Salary over Region")
boxplot(Annual ~ Prof+Region, data=sals, main="Annual Salary by Group")
interaction.plot(x.factor = sals$Prof,
                 trace.factor = sals$Region,
                 response = sals$Annual,
                 xlab = "Profession",
                 ylab = "Mean Annual Salary (thousands)",
                 trace.label = "Region",
                 col = c("#B78FF1","#F174BD"),
                 main = "Interaction plot of Region and Profession"
)
fitAB <- lm(Annual ~ Prof*Region, data=sals)
fitApB <- lm(Annual ~ Prof+Region, data=sals)
aov <- anova(fitApB, fitAB)
Fs = aov$F[2]
p = aov$`Pr(>F)`[2]
res <- c(Fs, p)
names(res) <- c("F-stat", "p")
kable(res, col.names = "", caption="Interaction Test Results")
fitB <- lm(Annual ~ Region, data=sals)
aov <- anova(fitB, fitApB)
Fs <- aov$F[2]
p <- aov$`Pr(>F)`[2]
res <- setNames(c(Fs, p), c("F-stat", "p-value"))
kable(res, col.names="", caption="Factor A test results")
fitA <- lm(Annual ~ Prof, data=sals)
aov <- anova(fitA, fitApB)
```

```r
Fs <- aov$F[2]
p <- aov$`Pr(>F)`[2]
res <- setNames(c(Fs, p), c("F-stat", "p-value"))
kable(res, col.names="", caption="Factor A test results")
fit <- fitApB
kable(anova(fit), caption = "(A+B) ANOVA Model Summary")
get.gamma.delta = function(the.model,the.data){
  nt = nrow(the.data)
  a = length(unique(the.data[,2]))
  b = length(unique(the.data[,3]))
  the.data$hat = the.model$fitted.values
  the.ns = find.means(the.data,length)
  a.vals = sort(unique(the.data[,2]))
  b.vals= sort(unique(the.data[,3]))
  muij = matrix(nrow = a, ncol = b)
  rownames(muij) = a.vals
  colnames(muij) = b.vals
  for(i in 1:a){
  for(j in 1:b){
  muij[i,j] = the.data$hat[which(the.data[,2] == a.vals[i] & the.data[,3] == b.vals[j])[1]]
  }
  }
  mi. = rowMeans(muij)
  m.j = colMeans(muij)
  mu.. = sum(muij)/(a*b)
  gammai = mi. - mu..
  deltaj = m.j - mu..
  gmat = matrix(rep(gammai,b),nrow = a, ncol = b, byrow= FALSE)
  dmat = matrix(rep(deltaj,a),nrow = a, ncol = b,byrow=TRUE)
  gamma.deltaij =round(muij -(mu.. + gmat + dmat),8)
  results = list(Mu.. = mu.., Gam = gammai, Del = deltaj, GamDel = gamma.deltaij)
  return(results)
}
find.means = function(the.data,fun.name = mean){
  a = length(unique(the.data[,2]))
  b = length(unique(the.data[,3]))
  means.A = by(the.data[,1], the.data[,2], fun.name)
  means.B = by(the.data[,1],the.data[,3],fun.name)
  means.AB = by(the.data[,1],list(the.data[,2],the.data[,3]),fun.name)
  MAB = matrix(means.AB,nrow = b, ncol = a, byrow = TRUE)
  colnames(MAB) = names(means.A)
  rownames(MAB) = names(means.B)
  MA = as.numeric(means.A)
  names(MA) = names(means.A)
  MB = as.numeric(means.B)
  names(MB) = names(means.B)
  MAB = t(MAB)
  results = list(A = MA, B = MB, AB = MAB)
  return(results)
}
gamdel <- get.gamma.delta(fit, sals)
gamma <- gamdel$Gam
delta <- gamdel$Del
```

```r
nt = nrow(sals)
a = nrow(mbg)
alpha = 0.05
t.cutoff = qt(1 - alpha/(2*nt), nt - a)

# Studentized DOES NOT ASSUME CONST VAR
rij <- rstandard(fit)
CO.rij <- which(abs(rij) > t.cutoff)

# Remove outliers
outliers = CO.rij
print_diag_plots <- function(fit, qqcaption = "Normal Q-Q Plot",
                             varcaption = "Errors vs. Fitted Values") {
  qqnorm(fit$residuals, main=qqcaption)
  qqline(fit$residuals)

  plot(fit$residuals ~ fit$fitted.values,
      main=varcaption,
      xlab="Fitted Values", ylab="Errors")
  abline(h=0, col="maroon")
}

print_shapiro_wilk <- function(numeric_values, print = FALSE) {
  shap <- shapiro.test(numeric_values)
  W = shap$statistic
  p = shap$p.value
  shap_res <- c(W,p, use.names=FALSE)
  shap_labels <- c("W (test statistic)", "p-value")
  if (print == FALSE) {return(data.frame(shap_labels, shap_res))}
  kable(data.frame(shap_labels, shap_res), col.names = c("",""),
       caption="Shapiro-Wilk's Test")

}
print_bf_test <- function(formula, data, alpha, print = FALSE) {
  res <- bf.test(formula, data=data, alpha=alpha, verbose=FALSE)
  ret <- c(res$statistic, res$p.value)
  names(ret) <- c("test-statistic", "p-value")
  if (print == FALSE) {return(ret)}
  kable(ret, col.names="", caption = "BF Test")
}

print_diag_tests <- function(shap, bf) {
  test_stats <- unname(c(shap[1,2], bf[1]))
  p_vals <- unname(c(shap[2,2], bf[2]))
  ret <- rbind(test_stats, p_vals)
  row.names(ret) <- c("test statistic", "p value")
  kable(ret, col.names = c("Shapiro-Wilk", "Brown-Forsythe"),
        caption="Diagnostic Tests")
}
qqnorm(fit$residuals)
qqline(fit$residuals)
print_shapiro_wilk(fit$residuals, print=TRUE)
plot(fit$residuals ~ fit$fitted.values,
```

```
    xlab="Fitted Values", ylab="Errors",
    main="Errors vs Fitted Values"
)
abline(h=0, col="maroon")
prof_var <- print_bf_test(Annual ~ Prof, data = sals, alpha=0.01)
reg_var <- print_bf_test(Annual ~ Region, data = sals, alpha=0.01)
res <- rbind(prof_var, reg_var)
row.names(res) <- c("Profession", "Region")
kable(res)
L <- boxcox(fit, objective.name = "Shapiro-Wilk", optimize=TRUE)$lambda
# L <- boxcox(sals$Annual, objective.name = "Log-Likelihood", optimize=TRUE)$lambda
annual <- (sals$Annual^L - 1)/L
tdata <- data.frame(Annual = annual, Prof = sals$Prof, Region = sals$Region)
tfit <- lm(Annual ~ Prof+Region, data=tdata)
par(mfrow=c(1,2))
print_diag_plots(tfit)
prof_var <- print_bf_test(Annual ~ Prof, data=tdata, alpha=0.0)
reg_var <- print_bf_test(Annual ~ Region, data=tdata, alpha=0.0)
res <- rbind(prof_var, reg_var)
row.names(res) <- c("Profession", "Region")
kable(res, caption="BF test results for transformed data")
alpha = 0.05
a = nrow(mean_by_prof)
b = nrow(mean_by_regn)
df = anova(fit)$Df[3]
g = 6
tuk = round(qtukey(1 - alpha, a*b, df)/sqrt(2), 3)
bon = round(qt(1 - alpha/(2*g), df), 3)
sch = round(sqrt((a * b - 1) * qf(1 - alpha, a * b - 1, df)), 3)
res <- setNames( c(tuk, bon, sch), c("Tukey", "Bonferroni", "Scheffe"))
kable(res, col.names = "Multiplier", caption="Simultaneous CI Correction Multipliers")
find.means = function(the.data,fun.name = mean){
  a = length(unique(the.data[,2]))
  b = length(unique(the.data[,3]))
  means.A = by(the.data[,1], the.data[,2], fun.name)
  means.B = by(the.data[,1],the.data[,3],fun.name)
  means.AB = by(the.data[,1],list(the.data[,2],the.data[,3]),fun.name)
  MAB = matrix(means.AB,nrow = b, ncol = a, byrow = TRUE)
  colnames(MAB) = names(means.A)
  rownames(MAB) = names(means.B)
  MA = as.numeric(means.A)
  names(MA) = names(means.A)
  MB = as.numeric(means.B)
  names(MB) = names(means.B)
  MAB = t(MAB)
  results = list(A = MA, B = MB, AB = MAB)
  return(results)
}

scary.CI = function(the.data,
                    MSE,
                    equal.weights = TRUE,
                    multiplier,
```

```
                    group,
                    cs) {
if (sum(cs) != 0 & sum(cs != 0) != 1) {
  return("Error - you did not input a valid contrast")
} else{
  the.means = find.means(the.data)
  the.ns = find.means(the.data, length)
  nt = nrow(the.data)
  a = length(unique(the.data[, 2]))
  b = length(unique(the.data[, 3]))
  if (group == "A") {
    if (equal.weights == TRUE) {
      a.means = rowMeans(the.means$AB)
      est = sum(a.means * cs)
      mul = rowSums(1 / the.ns$AB)
      SE = sqrt(MSE / b ^ 2 * (sum(cs ^ 2 * mul)))
      N = names(a.means)[cs != 0]
      CS = paste("(", cs[cs != 0], ")", sep = "")
      fancy = paste(paste(CS, N, sep = ""), collapse = "+")
      names(est) = fancy
    } else{
      a.means = the.means$A
      est = sum(a.means * cs)
      SE = sqrt(MSE * sum(cs ^ 2 * (1 / the.ns$A)))
      N = names(a.means)[cs != 0]
      CS = paste("(", cs[cs != 0], ")", sep = "")
      fancy = paste(paste(CS, N, sep = ""), collapse = "+")
      names(est) = fancy
    }
  } else if (group == "B") {
    if (equal.weights == TRUE) {
      b.means = colMeans(the.means$AB)
      est = sum(b.means * cs)
      mul = colSums(1 / the.ns$AB)
      SE = sqrt(MSE / a ^ 2 * (sum(cs ^ 2 * mul)))
      N = names(b.means)[cs != 0]
      CS = paste("(", cs[cs != 0], ")", sep = "")
      fancy = paste(paste(CS, N, sep = ""), collapse = "+")
      names(est) = fancy
    } else{
      b.means = the.means$B
      est = sum(b.means * cs)
      SE = sqrt(MSE * sum(cs ^ 2 * (1 / the.ns$B)))
      N = names(b.means)[cs != 0]
      CS = paste("(", cs[cs != 0], ")", sep = "")
      fancy = paste(paste(CS, N, sep = ""), collapse = "+")
      names(est) = fancy
    }
  } else if (group == "AB") {
    est = sum(cs * the.means$AB)
    SE = sqrt(MSE * sum(cs ^ 2 / the.ns$AB))
    names(est) = "someAB"
  }
  the.CI = est + c(-1, 1) * multiplier * SE
```

```
    results = c(est, the.CI)
    names(results) = c(names(est), "lower bound", "upper bound")
    return(results)
  }
}
SSE <- anova(fit)$`Sum Sq`[3]
dfSSE <- df
MSE <- SSE/dfSSE
groupmean <- find.means(sals)

I = scary.CI(sals, MSE, multiplier = bon, group = "B",cs = c(-1, 1))

cs <- cbind(c(0, 0, -1), c(0, 0, 1)) # SE.SF - SE.S
II = scary.CI(sals, MSE, multiplier = bon, group = "AB",cs = cs)

cs <- cbind(c(0, -1, 0), c(0, 1, 0)) # DS.SF - DS.S
III = scary.CI(sals, MSE, multiplier = bon, group = "AB",cs = cs)

cs <- cbind(c(-1, 0, 0), c(1, 0, 0)) # BE.SF - BE.S
IV = scary.CI(sals, MSE, multiplier = bon, group = "AB",cs = cs)

# V = scary.CI(sals, MSE, multiplier = bon, group = "A",cs = c(-1/2, -1/2, 1))

cs <- cbind(c(-1, 0, -1), c(1, 0, 1))
V = scary.CI(sals, MSE, multiplier = bon, group = "AB",cs = cs)

cs <- cbind(c(0, -1, -1), c(0, 1, 1))
VI = scary.CI(sals, MSE, multiplier = bon, group = "AB",cs = cs)

names(I)[1] <- "Estimate"
cis <- data.frame(rbind(I, II, III, IV, V, VI))
Labels <- c(
  "SF - S",
  "SE.SF - SE.S",
  "DS.SF - DS.S",
  "BE.SF - BE.S",
  "(BE.SF + SE,SF) - (BE.S + SE.S)",
  "(DS.SF + SE.SF) - (DS.S + SE.S)"
)
cis <- add_column(cis, Labels, .before="Estimate")
kable(cis, caption="95% Bonferroni Corrected Confidence Intervals")
```