Ross Lauterbach

STAT 706 HW #2

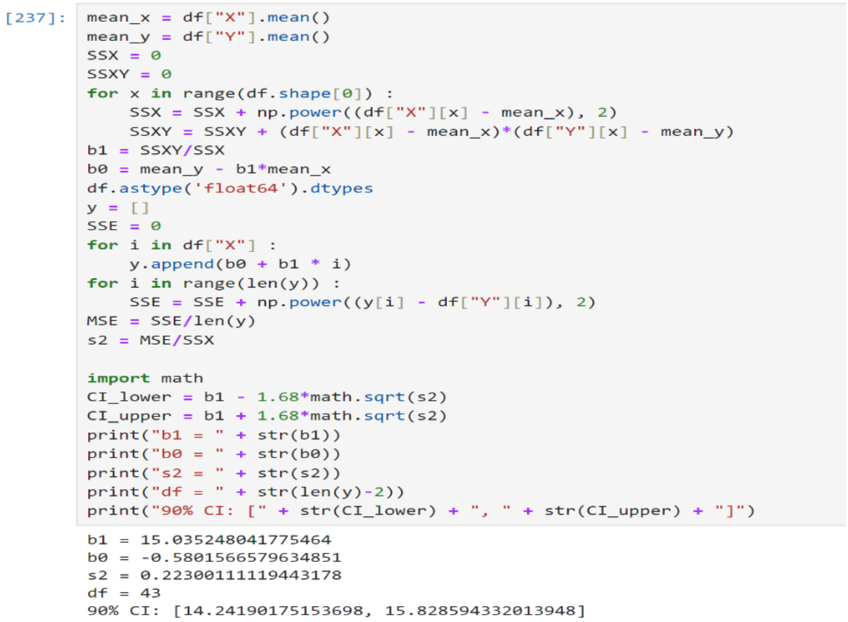
2.5) A. To compute a 90% confidence interval for β_1 , we need to use the following:

$$CI = b_1 \pm t_{(1-\alpha,n-2)} s\{b_1\}$$

Where b1 is the OLS estimate of β_1 and s{b_1} is calculated using the following formula:

$$s^2\{b_1\} = \frac{MSE}{\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

Using the following code in Python, I was able to load in the dataset and compute the OLS estimates b_1 and b_0 , the MSE, and the sum of squared deviations from the mean for x. I then used those calculations to build a confidence interval for b_1 using the fact that $t_{(0.9,43)} \approx \ 1.68\ according\ to\ a\ t - table.$



```
[237]: mean_x = df["X"].mean()
       mean_y = df["Y"].mean()
       SSX = 0
       SSXY = 0
       for x in range(df.shape[0]) :
           SSX = SSX + np.power((df["X"][x] - mean_x), 2)
           SSXY = SSXY + (df["X"][x] - mean_x)*(df["Y"][x] - mean_y)
       b1 = SSXY/SSX
       b0 = mean_y - b1*mean_x
       df.astype('float64').dtypes
       y = []
       SSE = 0
       for i in df["X"] :
           y.append(b0 + b1 * i)
       for i in range(len(y)) :
           SSE = SSE + np.power((y[i] - df["Y"][i]), 2)
       MSE = SSE/len(y)
       s2 = MSE/SSX

       import math
       CI_lower = b1 - 1.68*math.sqrt(s2)
       CI_upper = b1 + 1.68*math.sqrt(s2)
       print("b1 = " + str(b1))
       print("b0 = " + str(b0))
       print("s2 = " + str(s2))
       print("df = " + str(len(y)-2))
       print("90% CI: [" + str(CI_lower) + ", " + str(CI_upper) + "]")
```

```
b1 = 15.035248041775464
b0 = -0.5801566579634851
s2 = 0.22300111119443178
df = 43
90% CI: [14.24190175153698, 15.828594332013948]
```

So the 90% confidence interval for β_1 is [14.242, 15.829]

This means we are 90% confident that, on average, for each additional copier serviced, the service time increases by approximately 14.242 to 15.829 minutes

b) To test the linear relationship between X and Y, we conduct a t-test with the following null and alternative hypotheses:

$H_0\colon \beta_1 = 0$

$H_1\colon \beta_1 \neq 0$

At a significance level of 0.1, we reject the null hypothesis if the p-value is less than 0.1.

We also know that the t-statistic $t = \frac{b_1}{s\{b_1\}}$ follows a t distribution with n – 2 degrees of freedom. We saw from before that the critical t-value for a significance level of 0.1 and 43 degrees of freedom yields is about 1.68. This means our decision rule is that we reject the null hypothesis is the absolute value of the t-statistic is greater than 1.68. Using the point estimate b_1 and standard error from before, we find our t statistic to be 31.8, which yields of p-value of effectively zero. This means that we can reject the null hypothesis and confirms a linear association between the number of copiers serviced and the service time

c) My results from part a) and b) are consistent. In part a, I was 90% confident that there was a positive increase in service time as a result of increasing the number of copiers serviced. In part b, I concluded there was a significant linear association between the two variables, which supports the relationship that was shown in part a

d)

To test the linear relationship between X and Y, we conduct a t-test with the following null and alternative hypotheses:

$H_0\colon \beta_1 \geq 14$

$H_1\colon \beta_1 < 14$

At a significance level of 0.05, we reject the null hypothesis if the p-value is less than 0.05.

We also know that the t-statistic $t = \frac{\widehat{\beta_1} - 14}{s\{b_1\}}$ follows a t distribution with n – 2 degrees of freedom. This means our decision rule is that we reject the null hypothesis if the t-statistic is greater than -1.68. Using our calculated values, we get a t statistic of 2.14, which means we fail to reject the null hypothesis. This means that the data doesn't support the claim that the mean increase in service time for each additional copier is less than 14 minutes.

e) In this context, b_0 would represent the estimated service time when zero copiers are serviced. However, we calculated b_0 as -0.58, which does not make sense as you can not have a negative time. This is not meaningful and shows is b_0 does not give us any relevant information about the "start up" time on calls

2.14)

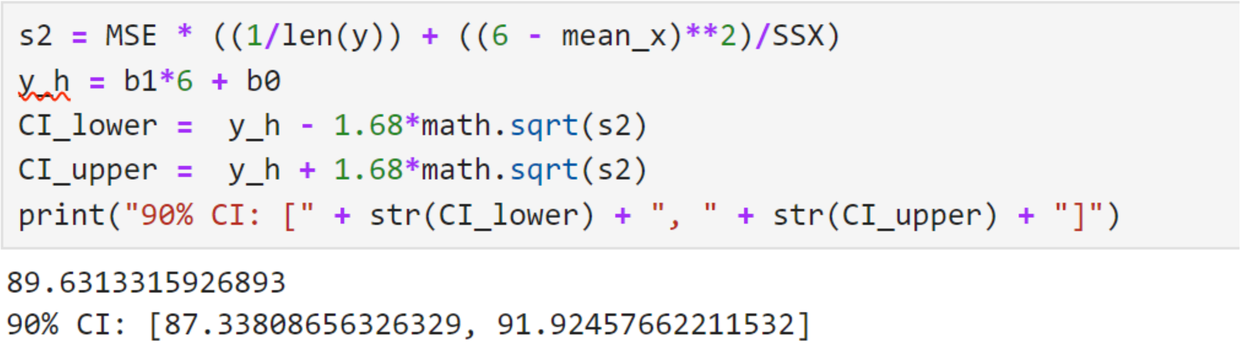
a) In order to construct a 90% confidence interval for $\widehat{Y_h}$ we use the following:

$$CI = \widehat{Y_h} \pm t_{(1-\alpha,n-2)} s\{\widehat{Y_h}\}$$

We must calculate the sample variance which is given by the formula:

$$s^2\{\widehat{Y_h}\} = MSE(\frac{1}{n} + \frac{(X_h - \bar{X})^2}{\sum_{i=i}^{n}(X_i - \bar{X})^2})$$

Using Python, I was able to use previous calculations and generate the 90% interval:



```
s2 = MSE * ((1/len(y)) + ((6 - mean_x)**2)/SSX)
y_h = b1*6 + b0
CI_lower =  y_h - 1.68*math.sqrt(s2)
CI_upper =  y_h + 1.68*math.sqrt(s2)
print("90% CI: [" + str(CI_lower) + ", " + str(CI_upper) + "]")
```

```
89.6313315926893
90% CI: [87.33808656326329, 91.92457662211532]
```

Where 1.68 was the value retrieved from the t-table. Therefore the 90% confidence interval for $\widehat{Y_h}$ is [87.34, 91.92]. This means we are 90% confident that the true mean service time for six copiers lies between 87.34 and 91.92 minutes

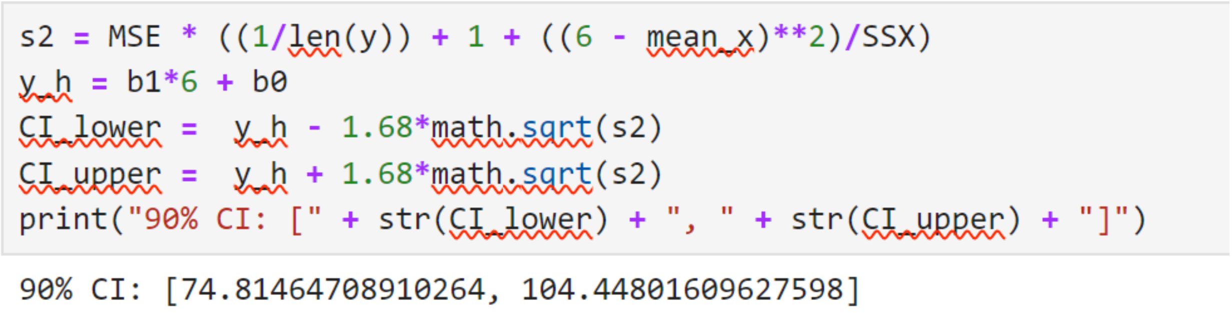
b) Instead of talking about the predicted true mean of already observed data, now we would like to find the service time on the next call in which six copiers are serviced. We use the following to construct a 90% confidence interval for $\widehat{Y_h}$

$$CI = \widehat{Y_h} \pm t_{(1-\alpha,n-2)} s\{\widehat{Y_h}\}$$

Where

$$s^2\{\widehat{Y_h}\} = MSE(\frac{1}{n} + 1 + \frac{(X_h - \bar{X})^2}{\sum_{i=i}^{n}(X_i - \bar{X})^2})$$

Using Python, I was again able to use previous calculations and generate the 90% interval:



```
s2 = MSE * ((1/len(y)) + 1 + ((6 - mean_x)**2)/SSX)
y_h = b1*6 + b0
CI_lower =  y_h - 1.68*math.sqrt(s2)
CI_upper =  y_h + 1.68*math.sqrt(s2)
print("90% CI: [" + str(CI_lower) + ", " + str(CI_upper) + "]")
```

```
90% CI: [74.81464708910264, 104.44801609627598]
```

This suggests that if Tri-City services six copiers on the next service call, the time required to complete the service is expected to fall between 74.81 and 104.45 minutes with 90% confidence. This interval is much wider than the confidence interval for the mean service time because there's more uncertainty about individual observations than about the average service time.

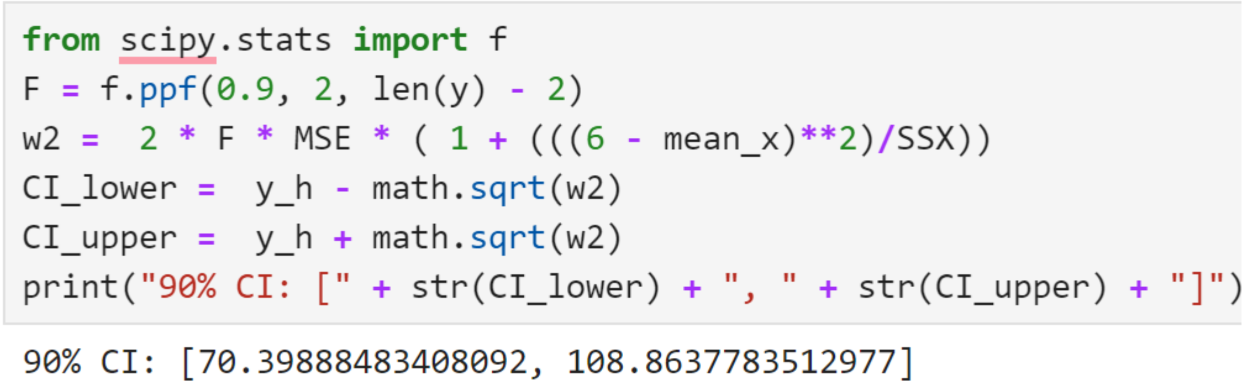
c) If we want to find a confidence interval for the average service time per one copier, we simply take the total time and divide it by 6, because we are assuming 6 copiers are serviced in that time. Since the 90% confidence interval for $\widehat{Y_h}$ is [87.34, 91.92], we divide each number by 6 and are left with the 90% interval for the average time to service one copier. This interval means that, with 90% confidence, the average time to service each copier on a call involving six copiers is expected to be between 14.55 and 15.33 minutes.

d) We know that the Work-Hotelling $1-\alpha$ confidence band for the regression line has the following two boundary values at any level X_h:

$$\hat{Y}_h \pm Ws\{\hat{Y}_h\}$$

$$W^2 = 2F(1-\alpha, 2, n-2)$$

I used Python to calculate the critical f value and construct the intervral:



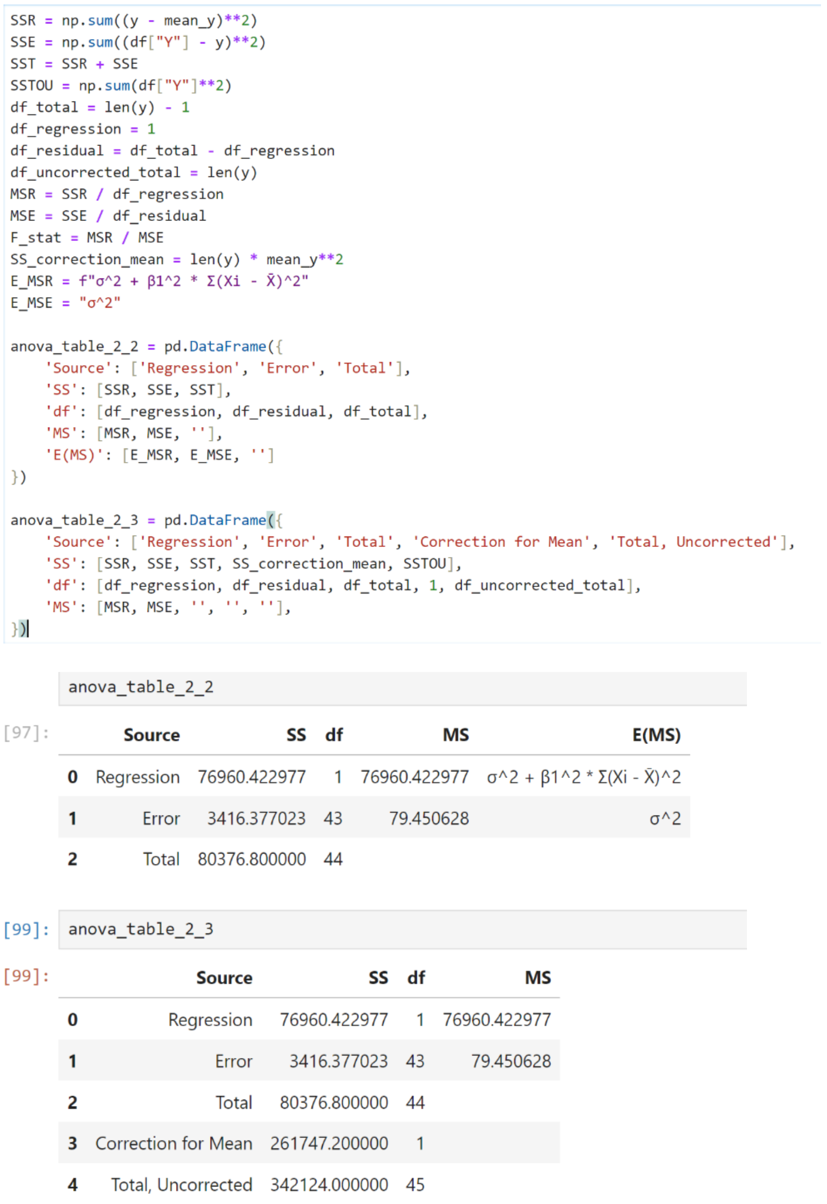
```
from scipy.stats import f
F = f.ppf(0.9, 2, len(y) - 2)
w2 =  2 * F * MSE * ( 1 + (((6 - mean_x)**2)/SSX))
CI_lower =  y_h - math.sqrt(w2)
CI_upper =  y_h + math.sqrt(w2)
print("90% CI: [" + str(CI_lower) + ", " + str(CI_upper) + "]")
```

```
90% CI: [70.39888483408092, 108.8637783512977]
```

This confidence band tells us that, with 90% confidence, the true mean service time when 6 copiers are serviced will lie between 70.40 and 108.86. Since this is based on Hotelling's method, it accounts for the uncertainty across the entire regression line, not just at x = 6

2.24

a) I used Python to take care of the remaining calculations and table formatting:



```
SSR = np.sum((y - mean_y)**2)
SSE = np.sum((df["Y"] - y)**2)
SST = SSR + SSE
SSTOU = np.sum(df["Y"]**2)
df_total = len(y) - 1
df_regression = 1
df_residual = df_total - df_regression
df_uncorrected_total = len(y)
MSR = SSR / df_regression
MSE = SSE / df_residual
F_stat = MSR / MSE
SS_correction_mean = len(y) * mean_y**2
E_MSR = f"σ^2 + β1^2 * Σ(Xi - X̄)^2"
E_MSE = "σ^2"

anova_table_2_2 = pd.DataFrame({
    'Source': ['Regression', 'Error', 'Total'],
    'SS': [SSR, SSE, SST],
    'df': [df_regression, df_residual, df_total],
    'MS': [MSR, MSE, ''],
    'E(MS)': [E_MSR, E_MSE, '']
})

anova_table_2_3 = pd.DataFrame({
    'Source': ['Regression', 'Error', 'Total', 'Correction for Mean', 'Total, Uncorrected'],
    'SS': [SSR, SSE, SST, SS_correction_mean, SSTOU],
    'df': [df_regression, df_residual, df_total, 1, df_uncorrected_total],
    'MS': [MSR, MSE, '', '', ''],
})
```

```
anova_table_2_2
```

[97]:

	Source	SS	df	MS	E(MS)
0	Regression	76960.422977	1	76960.422977	σ^2 + β1^2 * Σ(Xi - X̄)^2
1	Error	3416.377023	43	79.450628	σ^2
2	Total	80376.800000	44		

[99]:

```
anova_table_2_3
```

[99]:

	Source	SS	df	MS
0	Regression	76960.422977	1	76960.422977
1	Error	3416.377023	43	79.450628
2	Total	80376.800000	44	
3	Correction for Mean	261747.200000	1	
4	Total, Uncorrected	342124.000000	45	

The basic table includes the Sum of Squares (SS), Degrees of Freedom (df), Mean Squares (MS), and the Expected Mean Squares (E(MS)) for both regression and error. On the other hand, the modified table adds the Correction for Mean and the Total Uncorrected Sum of Squares (SSTOU). The sum of squares are additive as seen in the table.

b)

To test the linear relationship between X and Y, we conduct a t-test with the following null and alternative hypotheses:

$H_0: \beta_1 = 0$

$H_1: \beta_1 \neq 0$

If the calculated F-statistic is greater than the critical value from the F-distribution for a 0.1 significance level, we reject the null hypothesis.

```
F = f.ppf(0.9, 1, len(y) - 1)
F_stat = MSR/MSE
F, F_stat
```

```
(2.8231727741715362, 968.6571959790681)
```

We see that the critical f-value is 2.82 and our calculated f-statistic is 968.66. This means our decision rule is that we reject the null hypothesis if our f-statistic is greater than 2.82, which it is in this case. This means that we can reject the null hypothesis and confirms a linear association between the number of copiers serviced and the service time.

c) To compute the total variation in number of minutes spent on a call reduced when the number of copiers serviced is introduced into the analysis, we calculate:

$$R^2 = \frac{SSR}{SST}$$

```
r2 = SSR / SST
r2
```

```
0.95749548347908
```

This means that about 95.75% of the total variation in service time is explained by the number of copiers serviced. This is a large reduction in total variation, meaning the number of copiers serviced explains a substantial portion of the variability in service time.

d) To compute the correlation coefficient, we have to consider two things: the sign of the point estimate for slope of the regression line, and the square root of the total variation in number of minutes spent on a call reduced when the number of copiers serviced is introduced into the analysis. By simply taking the square root of R^2 and observing that b_1 is a positive estimator, we find r = 0.979, which indicates a strong positive linear relationship between the number of copiers serviced and the service time.

2.55

We know from class that SSR is defined as follows:

$$SSR = \sum\left(\hat{Y}_i - \bar{Y}\right)^2$$

Where $\hat{Y}_i$ are the fitted values and $\bar{Y}$ is the mean of the observed y-values. We also know that our fitted values are of the form $\hat{Y}_i = b_0 + b_1 X_i$, and that our OLS estimate b_0 is defined as $b_0 = \bar{Y} - b_1\bar{X}$. We can substitute these directly into the given expression:

$SSR = \sum\left(\hat{Y}_i - \bar{Y}\right)^2 = \sum(b_0 + b_1X_i - \bar{Y})^2 = \sum((\bar{Y} - b_1\bar{X}) + b_1X_i - \bar{Y})^2 =$

$\sum(-b_1\bar{X} + b_1X_i)^2 = \sum(b_1x_i - b_1\bar{X})^2 = {b_1}^2\sum(X_i - \bar{X})^2$

Which gives us the desired expression, $SSR = {b_1}^2\sum(X_i - \bar{X})^2$

2.56

We saw from our ANOVA table that we can directly compute the expected value of MSE and MSR given the observed x-values and the true mean of the point estimates. The equations are:

$$E(MSE) = \sigma^2, \;\; E(MSR) = \; \sigma^2 + \beta_1 \sum_{i=i}^{n} (X_i - \bar{X})^2$$

The mean of the dataset {1, 4, 10, 11, 14}, $\bar{X}$, is 8.

We are given σ^2 = 0.36, $\beta_0 = 5$, and $\beta_1 = 3$, and can directly compute desired expectations

$$E(MSE) = \sigma^2 = 0.36,$$

$$E(MSR) = \; \sigma^2 + \beta_1 \sum_{i=i}^{n} (X_i - \bar{X})^2 = 0.36 + 3 \sum_{i=i}^{n} (X_i - 8)^2 = 1026.36$$

b) It would have been worse to take the observations at X = 6, 7, 8, 9, 10 because this set has a smaller spread compared our original data. A wider spread of x values allows us to better detect a relationship, and also want maximize E[MSR] ideally, so maximizing the sum of squares and therefore spread would allow us to do that

For X = 8, It would have been better to take observations around X = 8 because the closer the observations are, the more precise the estimate for the mean response at that specific point.

Proof From Slides

We know that from OLS estimation, we have the following two estimates, b_0 and b_1 :

$b_0 = \bar{Y} - b_1\bar{X}$ and $b_1 = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sum(X_i - \bar{X})^2}$

First, lets compute $E[b_0]$

$$E[b_0] = E[\bar{Y} - b_1\bar{X}] = E[\bar{Y}] - \bar{X}E[b_1]] = \beta_0 + \beta_1\bar{X} + \beta_1\bar{X} = \beta_0$$

Using the fact that $E[\bar{Y}] = \beta_0 + \beta_1\bar{X}$ and $E[b_1] = \beta_1$

Now we can derive the variance of b_0:

$$Var(b_0) = Var(\bar{Y} - b_1\bar{X}) = Var(\bar{Y}) + \bar{X}^2 Var(b_1)$$

We know from previous proofs that $Var(\bar{Y}) = \frac{\sigma^2}{n}$ and $Var(b_1) = \frac{\sigma^2}{\sum(X_i - \bar{X})^2}$

So we get that

$Var(b_0) = Var(\bar{Y}) + \bar{X}^2 Var(b_1) = \frac{\sigma^2}{n} + \frac{\sigma^2}{\sum(X_i - \bar{X})^2}\overline{X^2} = \sigma^2(\frac{1}{n} + \frac{\overline{X^2}}{\sum(X_i - \bar{X})^2})$

As desired.

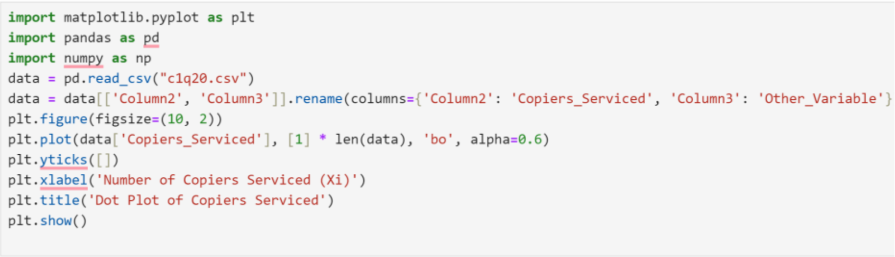
Ross Lauterbach

STAT 706 HW #3

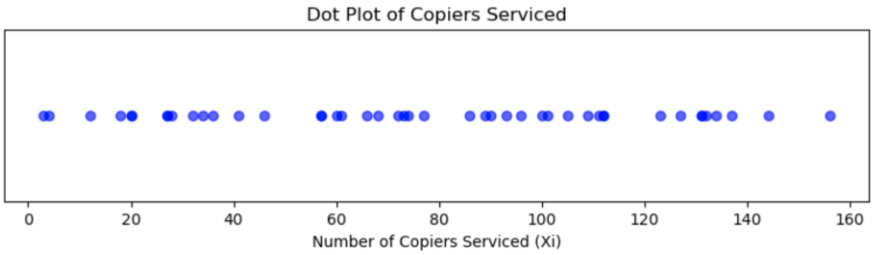
3.4)

a)

I used Python to all of the visualizations for this question.

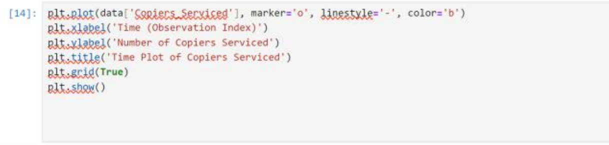


```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
data = pd.read_csv("c1q20.csv")
data = data[['Column2', 'Column3']].rename(columns={'Column2': 'Copiers_Serviced', 'Column3': 'Other_Variable'}
plt.figure(figsize=(10, 2))
plt.plot(data['Copiers_Serviced'], [1] * len(data), 'bo', alpha=0.6)
plt.yticks([])
plt.xlabel('Number of Copiers Serviced (Xi)')
plt.title('Dot Plot of Copiers Serviced')
plt.show()
```

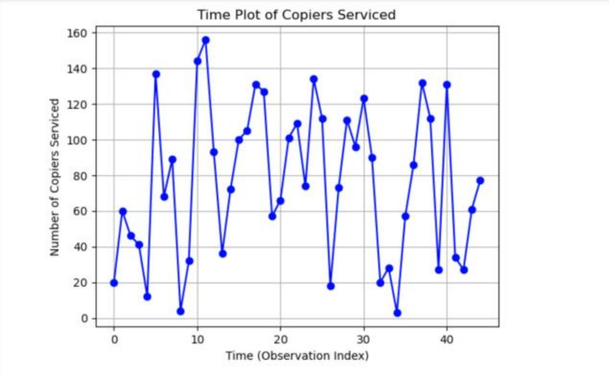


The dot plot shows that no extreme outliers appear for X_i

b)



```
[14]: plt.plot(data['Copiers_Serviced'], marker='o', linestyle='-', color='b')
      plt.xlabel('Time (Observation Index)')
      plt.ylabel('Number of Copiers Serviced')
      plt.title('Time Plot of Copiers Serviced')
      plt.grid(True)
      plt.show()
```



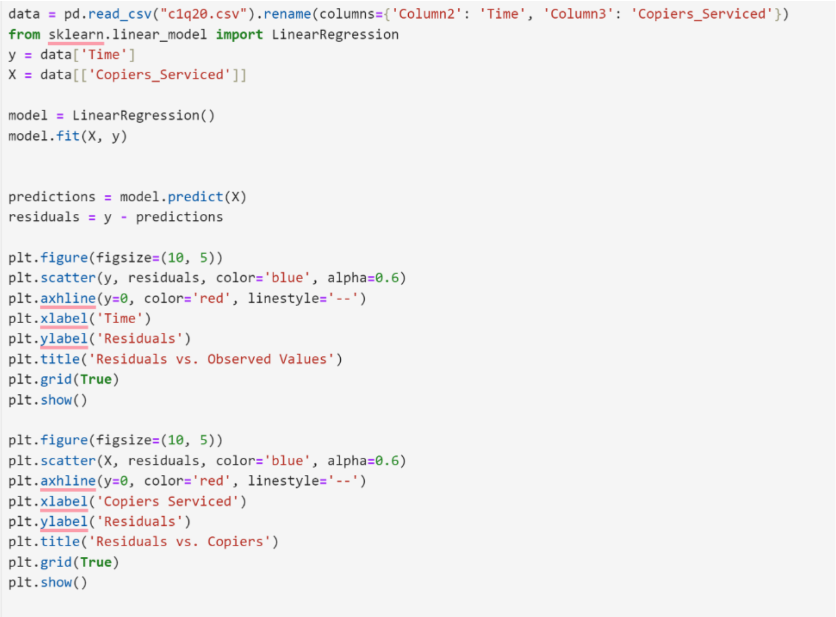
This time plot the amount of copiers serviced does not generally depend on the time or order of servicing

c) The following is the stem and leaf plot for the residuals, with H indicating hinges and M indicating the median:

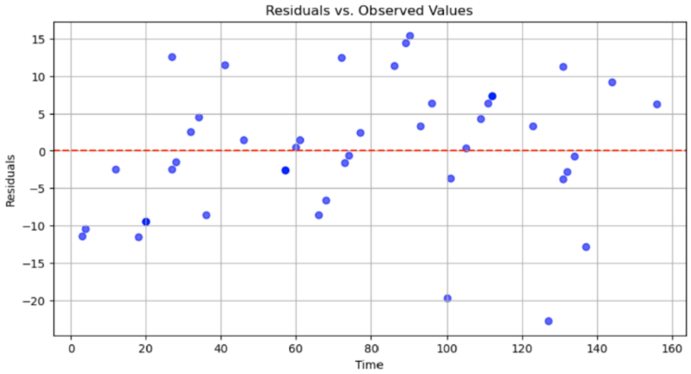
```
-23   | 2
 -20  | 3
 -13  | 2
 -12  | 5 5
 -11  | 5
 -10  | 5 5
 -9   | 4 5
 -7   | 4
 -4H  | 3 3
 -3   | 3 4 4 5 5
 -2   | 4 5
 -1   | 3 4
  0M  | 3 4
  1   | 4 5
  2   | 4 5
  3   | 3 4
  4   | 3 5
  6H  | 2 3 4
  7   | 3 3
  9   | 3
 11   | 3 4 5
 12   | 4 5
 14   | 4
 15   | 4
```

The residuals look relatively symmetric around the median, and there do not seem to be major outliers. In this case, the stems are ones and the leaves are tenths places for more clarity.

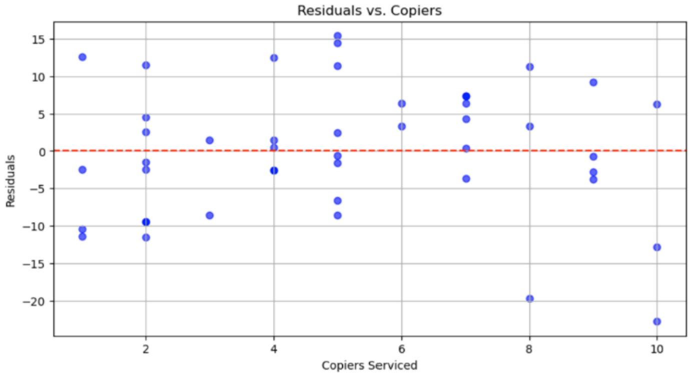
d)



```
data = pd.read_csv("c1q20.csv").rename(columns={'Column2': 'Time', 'Column3': 'Copiers_Serviced'})
from sklearn.linear_model import LinearRegression
y = data['Time']
X = data[['Copiers_Serviced']]

model = LinearRegression()
model.fit(X, y)


predictions = model.predict(X)
residuals = y - predictions

plt.figure(figsize=(10, 5))
plt.scatter(y, residuals, color='blue', alpha=0.6)
plt.axhline(y=0, color='red', linestyle='--')
plt.xlabel('Time')
plt.ylabel('Residuals')
plt.title('Residuals vs. Observed Values')
plt.grid(True)
plt.show()

plt.figure(figsize=(10, 5))
plt.scatter(X, residuals, color='blue', alpha=0.6)
plt.axhline(y=0, color='red', linestyle='--')
plt.xlabel('Copiers Serviced')
plt.ylabel('Residuals')
plt.title('Residuals vs. Copiers')
plt.grid(True)
plt.show()
```



Residuals vs. Observed Values

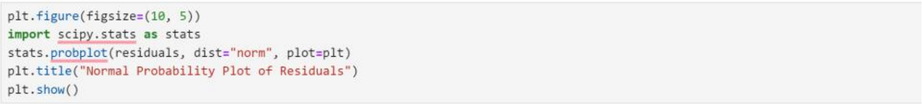


Residuals vs. Copiers

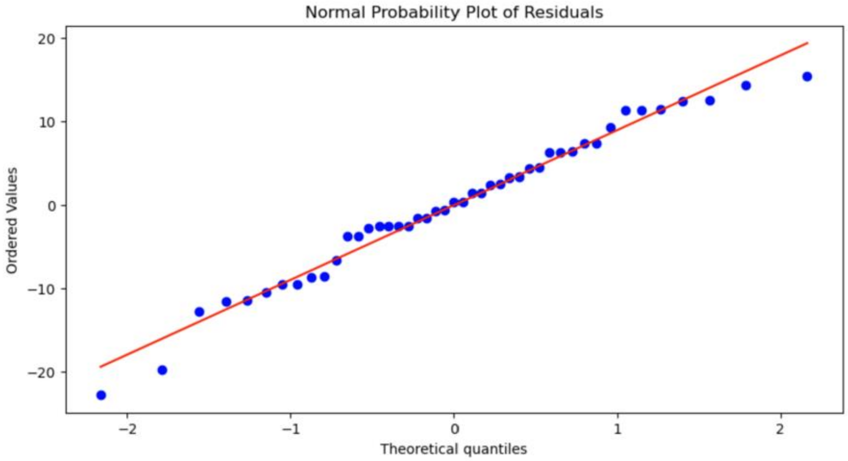
These plots show a relatively even scatter around zero without a clear pattern. This also supports the assumption of homoscedasticity with respect to the predictor variable because the spread looks pretty constant as we move up the graph. Generally, we also can assess linearity, normality of errors, and independence.

e)

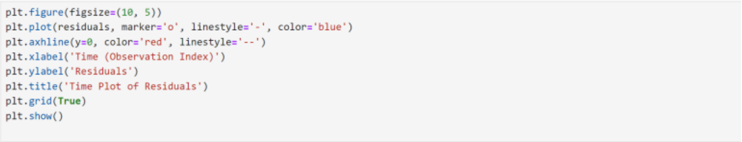
I calculated the correlation coefficient and normal plot using Python. We are looking to test the normality of residuals.



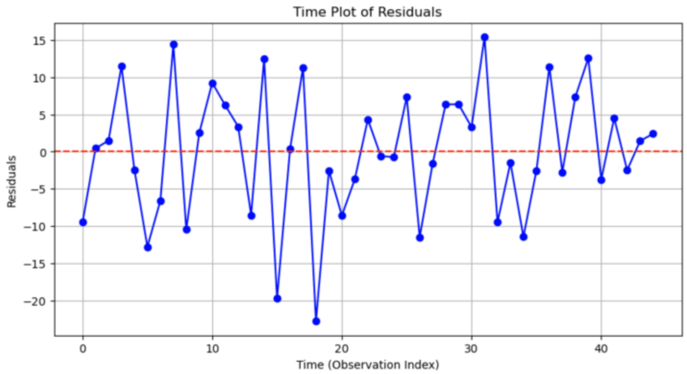
```
plt.figure(figsize=(10, 5))
import scipy.stats as stats
stats.probplot(residuals, dist="norm", plot=plt)
plt.title("Normal Probability Plot of Residuals")
plt.show()
```



The calculated correlation coefficient was 0.989, which supports normality of the residuals. Using table B6, we find a critical value of 0.977. Since our calculated correlation coefficient of 0.989 exceeds this critical value, we fail to reject the normality assumption, supporting that the residuals are likely normally distributed f)



```
plt.figure(figsize=(10, 5))
plt.plot(residuals, marker='o', linestyle='-', color='blue')
plt.axhline(y=0, color='red', linestyle='--')
plt.xlabel('Time (Observation Index)')
plt.ylabel('Residuals')
plt.title('Time Plot of Residuals')
plt.grid(True)
plt.show()
```



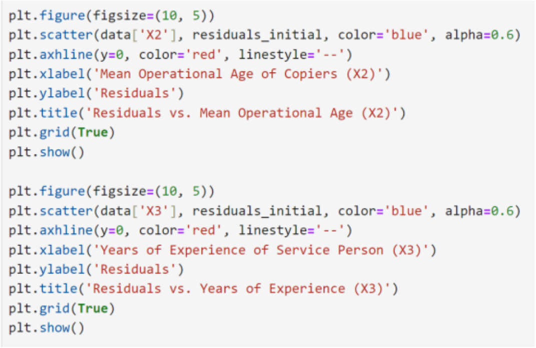
The time plot no clear pattern or trend, suggesting that there is likely no autocorrelation, and the residuals appear independent over time.

g) The Breusch-Pagan tests the following from the transformation given in 3.10

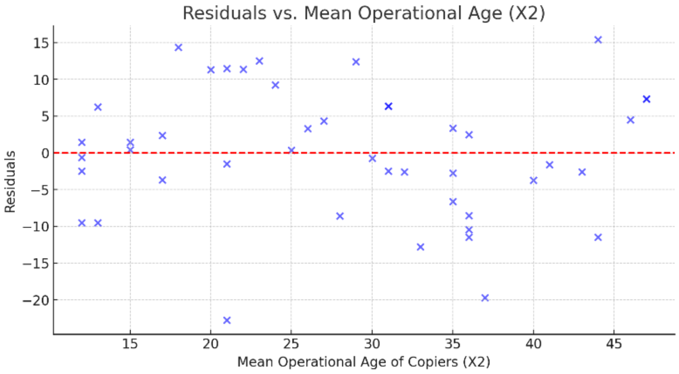
$$H_0\colon \gamma_1 = 0$$

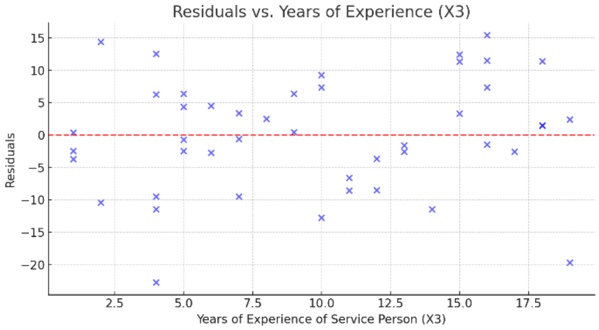
$$H_1\colon \gamma_1 \neq 0$$

If H_o is true, then the error variance does not vary with the level of X. Our decision rule rejects H_0 if the p-value is less than 0.05. Using Python we get a p-value of 0.85 and a test statistic of 0.033. Because of this, we fail to reject the null hypothesis and conclude that heteroscedasticity is not present h)



```
plt.figure(figsize=(10, 5))
plt.scatter(data['X2'], residuals_initial, color='blue', alpha=0.6)
plt.axhline(y=0, color='red', linestyle='--')
plt.xlabel('Mean Operational Age of Copiers (X2)')
plt.ylabel('Residuals')
plt.title('Residuals vs. Mean Operational Age (X2)')
plt.grid(True)
plt.show()

plt.figure(figsize=(10, 5))
plt.scatter(data['X3'], residuals_initial, color='blue', alpha=0.6)
plt.axhline(y=0, color='red', linestyle='--')
plt.xlabel('Years of Experience of Service Person (X3)')
plt.ylabel('Residuals')
plt.title('Residuals vs. Years of Experience (X3)')
plt.grid(True)
plt.show()
```





The residual plots against mean operational age of copiers and years of experience of the service person reveal no strong pattern. This suggests that these variables do not show significant relationships with the residuals, implying that adding them might not improve the model by a significant amount

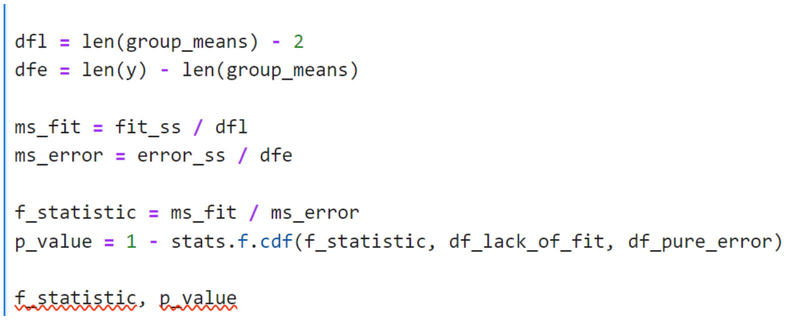
3.13)

a)

In testing for lack of fit in a linear regression function, we are checking whether a linear model is adequate or if a different model better describes the relationship between the predictor and the outcome. The null hypothesis states that the linear model fits the data adequately, and the alternative states that a linear model does not fit the data adequately, and that the error is not due to randomness.

b)

I used Python to carry out the test



```
dfl = len(group_means) - 2
dfe = len(y) - len(group_means)

ms_fit = fit_ss / dfl
ms_error = error_ss / dfe

f_statistic = ms_fit / ms_error
p_value = 1 - stats.f.cdf(f_statistic, df_lack_of_fit, df_pure_error)

f_statistic, p_value
```

I calculated an f_statistic of .914 and a p-value of .516. I calculated the critical value for Type I Error probability of 0.05 as 2.217. Since our F-statistic is less than 2.217 and equivalently our p-value is greater than 0.05, we fail to reject the null hypothesis, meaning there is no significant lack of fit.

c)

The lack-of-fit test specifically checks whether a linear model adequately describes the relationship between the predictor and response variables, but it does not detect other departures from the regression assumptions like non-constant variance or non-normality in the error terms. For example, if the error variance changes with the level of the predictor, the test may work well because it assumes constant variance, and will produce an inaccurate result.

3.19)

Remember that residuals are defined by the following equation, and can be rearranged:

$$e_i = Y_i - \hat{Y}_i$$

$$Y_i = e_i + \hat{Y}_i$$

We see here that the observed values are the sum of the predicted values and residuals. Since our fitted values are not a random variable, this relationship indicates that residuals are essentially a part of the observed values. Therefore, the sum shows that plotting residuals versus observed Y generally shows a positive relation, while plotting residuals versus fitted values avoids this issue because fitted values are deterministic. The relationship between residuals and observed values creates a misleading correlation, therefore plotting residuals vs. fitted values is more meaningful.

3.21) We are aiming to split the SSE up into components. Let's start by rewriting the residual as:

$$e_i = y_i - \hat{y}_i = \left(y_i - \bar{y}_{x_i}\right) + (\bar{y}_{x_i} - \ \hat{y}_i)$$

For observation i, where $\bar{y}_{x_i}$ is the mean of observed y values at level x_i. The SSPE corresponds to the first part, which is the variation within group, while the SSLF is the second part representing between group variation. Squaring both sides yields

$$e_i^2 = \left(\left(y_i - \bar{y}_{x_i}\right) + \left(\bar{y}_{x_i} - \ \hat{y}_i\right)\right)^2 = \left(y_i - \bar{y}_{x_i}\right)^2 + 2\left(y_i - \bar{y}_{x_i}\right)\left(\bar{y}_{x_i} - \ \hat{y}_i\right) + \left(\bar{y}_{x_i} - \ \hat{y}_i\right)^2$$

Summing over both sides gives:

$$\sum e_i^2 = \sum(\left(y_i - \bar{y}_{x_i}\right)^2 + 2\left(y_i - \bar{y}_{x_i}\right)\left(\bar{y}_{x_i} - \ \hat{y}_i\right) + \left(\bar{y}_{x_i} - \ \hat{y}_i\right)^2$$

Taking a look at the middle term, we can divide it up into two sums, first iterating over each group and then each observation in that group:

$$2\sum_{G}\left(\bar{y}_{x_i} - \ \hat{y}_i\right)\sum_{i\in G}\left(y_i - \bar{y}_{x_i}\right)$$

We know $\sum_{i\in G}\left(y_i - \bar{y}_{x_i}\right) = 0$ since the sum of deviations within each group is 0 because we are using the group mean and only data within that group. Therefore we have that

$$\sum e_i^2 = \sum\left(y_i - \bar{y}_{x_i}\right)^2 + \sum\left(\bar{y}_{x_i} - \ \hat{y}_i\right)^2$$

Or $SSE = SSPE + SSLR$ using our definitions.

Ross Lauterbach

STAT 706 HW #4

4.3)

a) In this model for copier maintenance, the errors will tend to go in opposite directions for b_0 and b_1. For example, if the slope $b_1$we calculate is an over-estimate of our true slope parameter β_1, then for each copier we add, we are adding additional time that does not reflect the true regression line. However, this needs to be balanced out to achieve least squares, so each outcome expected by the regression line is shifted down to make up for this. In other words, b_0 would then effectively be underestimated to account for the unjustified increase in service time caused by overestimating b_1.

b) For Bonferroni joint confidence intervals, we need to widen our individual confidence intervals for our desired coverage. If we want a 95% family-wise confidence interval for β_1 and β_0, each individual interval is computed with total significance $\frac{\alpha}{2}$ as opposed to α, which means we have a two-sided individual confidence level of $1-\frac{\alpha}{4}$ rather than $1-\frac{\alpha}{2}$.

In this specific case, $1-\frac{\alpha}{4}=0.975,$ so we construct individual 97.5% confidence intervals.

For β_0 we get $\widehat{\beta_0}\ \pm t_{\left(1-\frac{\alpha}{4},\ n-2\right)}s\{\widehat{\beta_0}\}$ and similarly $\widehat{\beta_1}\ \pm t_{\left(1-\frac{\alpha}{4},\ n-2\right)}s\{\widehat{\beta_1}\}$

Using code posted below I was easily able to calculate these intervals and found the following:

$\beta_0 \in [-6.56, 6.77]$ and $\beta_1 \in [13.81, 16.08]$

This means we are 95% confident that the true intercept lies between −6.56 and 6.77 AND that the true slope is between 13.81 and 16.08

c)

The consultant proposed that the intercept should 0 and the slope should be 14. To check this, we can use the Bonferroni confidence intervals we just found. For intercept, the suggested value of zero falls within the confidence interval for the intercept. This means that having an intercept of zero is plausible according to our model, which would be the time for no copiers serviced, which makes sense logically. For slope, the suggested value of fourteen also falls within the confidence interval for the slope. This indicates that an additional fourteen minutes per copier serviced is a feasible estimate according to the data. Therefore, both the intercept and slope values suggested by the consultant are consistent with our family-wise CI, and I would support the recommendation.

4.7

a) To estimate the mean response for levels of 3, 5, and 7 copiers, we can use the Working-Hotelling Procedure to achieve a 90% family-wise confidence level. The equation for the critical value is as follows: $W = \sqrt{2F_{1-\alpha,g,n-2}}$. Here, g =3. We can construct individual confidence intervals using Python, and find the following:

- The CI for the mean time to service 3 copiers is [113.64, 129.22]
- The CI for the mean time to service 5 copiers is [196.08, 208.29]
- The CI for the mean time to service 7 copiers is [275.63, 290.25]

These intervals give us simultaneous estimates for the mean with 90% family-wise confidence.

b) From class and the textbook, we can calculate the critical values using Scheffe's and Bonferroni's methods respectively:

$$S = \sqrt{gF_{1-\alpha,g,n-2}}, B = t_{1-\frac{\alpha}{2g},\ n-2}$$

Where g is the number of parameters and n is the sample size. Scheffé's method is better for predicting many means because it uses an F-distribution to account for the broader range of parameters. This then means Scheffé's produces wider intervals for smaller numbers of predictions. Bonferroni's method, on the other hand, is more efficient with a smaller g value. So in our case where g = 2, we would opt to use the Bonferroni procedure

c)

In our case, since we want a 90% family confidence level, we take $\alpha = 0.1$ and g = 2 since we are predicting two values. Again we have:

$$S = \sqrt{gF_{1-\alpha,g,n-2}}, B = t_{1-\frac{\alpha}{2g}, \ n-2}$$

Using the code below, I calculated the following intervals:

4 Copiers: Scheffe interval: [98.67, 165.55], Bonferroni interval: [84.15, 157.55]

7 copiers: Scheffe interval: [197.49, 264.55], Bonferroni interval: [174.52, 248.13]

It seems as though, in this case, the Scheffe intervals are wider than the Bonferroni intervals. So Bonferroni provides tighter prediction limits. Here is the code I used for all of my confidence and prediction intervals:

```
def calculate_confidence_interval(x_values, confidence_level=0.90):
    critical_value = np.sqrt(2 * f.ppf(confidence_level, 2, df_resid))
    intervals = []
    for x in x_values:
        y_hat = model.predict([1, x])[0]
        se = s * np.sqrt((1 / n) + ((x - X_mean) ** 2 / SXX))
        margin = critical_value * se
        intervals.append((y_hat - margin, y_hat + margin))
    return intervals

def calculate_prediction_interval(x_values, confidence_level=0.90, method="Bonferroni"):
    k = len(x_values)
    alpha = 1 - confidence_level
    if method == "Bonferroni":
        critical_value = t.ppf(1 - alpha / (2 * k), df_resid)
    elif method == "Scheffé":
        critical_value = np.sqrt(k * f.ppf(confidence_level, k, df_resid))
    intervals = []
    for x in x_values:
        y_hat = model.predict([1, x])[0]
        se = s * np.sqrt(1 + (1 / n) + ((x - X_mean) ** 2 / SXX))
        margin = critical_value * se
        intervals.append((y_hat - margin, y_hat + margin))
    return intervals

x_values_a = [3, 5, 7]
confidence_intervals_a = calculate_confidence_interval(x_values_a)

x_values_c = [4, 7]
prediction_intervals_c = calculate_prediction_interval(x_values_c, method="Bonferroni")
```

4.16)

a) The estimated regression function for predicting service time, Y, based on the number of copiers serviced, X, is $Y_i = 14.9472X_i$ using the fact that $\beta_1 = \frac{\sum(X_i-\bar{X})(Y_i-\bar{Y})}{\sum(X_i-\bar{X})^2}$ for this model

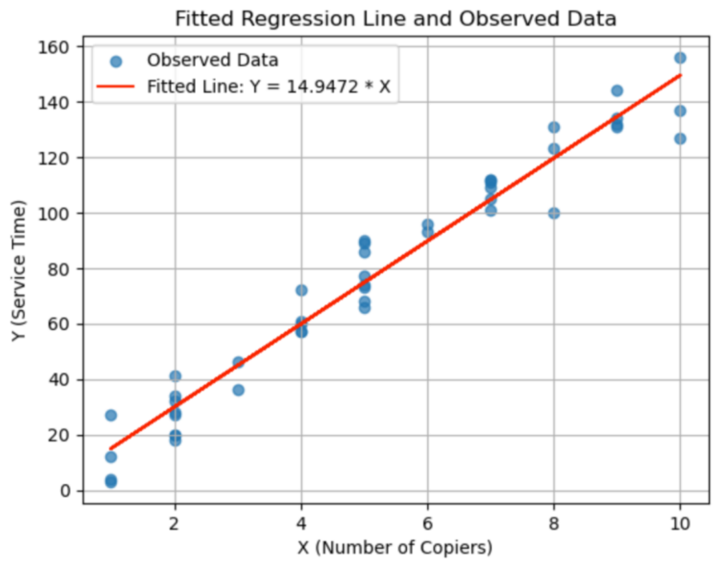
This means that, on average, the service time increases by approximately 14.95 minutes for each additional copier serviced. The model assumes a linear relationship through the origin, which implies no fixed time is required if no copiers are serviced.

b) We can use Python to construct the confidence interval from the functions made for the last question. I found that the 90% confidence interval for the slope $\beta_1$of the regression line is [14.5668,15.3277] using the following equation: $\widehat{\beta_1} \pm t_{\left(1-\frac{\alpha}{2},\ n-2\right)}s\{\widehat{\beta_1}\}$ which is for an individual confidence interval. This suggests that the true average service time per copier is likely between 14.57 and 15.33 minutes with 90% confidence.

c) I can also use the above functions to construct a prediction interval. The equation stays the same, but we use a different standard error: $\widehat{\beta_1} \pm t_{\left(1-\frac{\alpha}{2},\ n-2\right)}s\{pred\}$

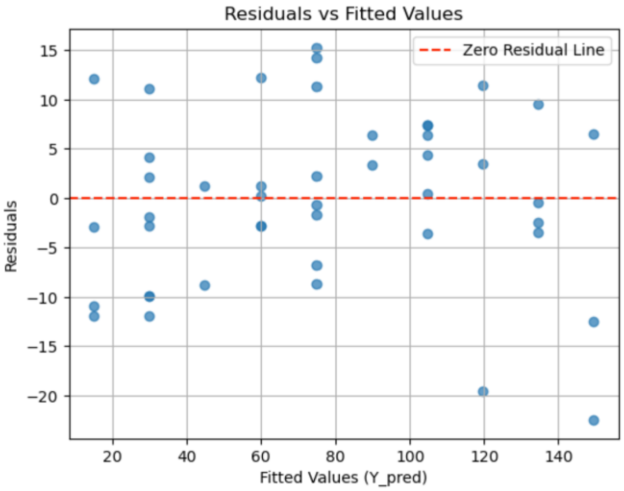
For a service call involving six copiers, the predicted service time is expected to fall within the interval of [74.6956 to 104.67121] minutes, with 90% confidence. The prediction interval is a bit wider because it accounts for more uncertainty with new data.

4.17: I plotted the regression line in Python:



The plot of the observed data vs the fitted regression line indicates that the linear regression function through the origin provides a reasonable fit, as data points are generally close to the line. We do see some deviations across the line, though.

b) After calculating residuals, I found that they summed to -5.863. However, this does not indicate poor model performance, because we are asserting that this model is going through the origin. We can also plot the residuals vs. fitted values:



Based on this plot, it seems that the residuals are randomly distributed around the red dashed line and I can not identify any clear pattern. Based on this, this regression model seems to be a good fit.

c) For the regression model through the origin, our F-test for lack of fit will be a bit different. To start, we have a full and reduced model. Since we have no intercept, it looks like this:

$$Full\ Model: Y_i = \beta_1 X_i + \varepsilon_i$$

$$Reduced\ Model: Y_i = \varepsilon_i$$

With our hypotheses being that:

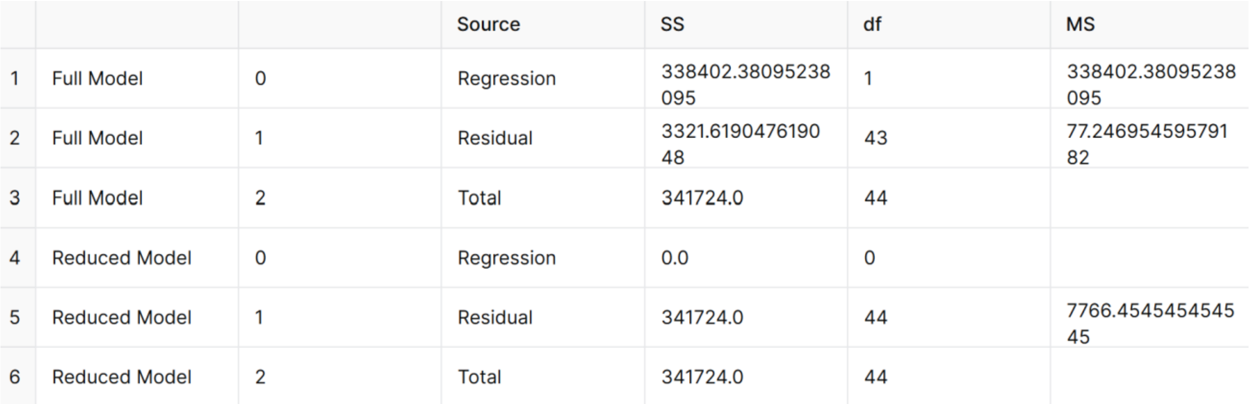
$$H_0: \beta_1 = \ 0$$

$$H_A: \beta_1 \neq 0$$

We then calculate the F-statistic using the following formula:

$$F^* = \frac{SSE(R) - SSE(F)}{df_r - df_f} \div \frac{SSE(F)}{df_f}$$

Using Python, I produced both models and generated the following ANOVA tables:



			Source	SS	df	MS
1	Full Model	0	Regression	338402.38095238095	1	338402.38095238095
2	Full Model	1	Residual	3321.619047619048	43	77.24695459579182
3	Full Model	2	Total	341724.0	44	
4	Reduced Model	0	Regression	0.0	0	
5	Reduced Model	1	Residual	341724.0	44	7766.454545454545
6	Reduced Model	2	Total	341724.0	44	

Using this, we can calculate the F statistic as:

$$F^* = \frac{341724 - 3321.62}{44 - 43} \div \frac{3321.62}{43} = 4380.78$$

For significance $\alpha = 0.01$, we reject the null hypothesis when our F statistic exceeds the critical value of approximately 7.26. Because of this, we reject the null hypothesis, supporting the fact that there is a significant linear relationship between the two variables. My calculated p-value using Python was $p = 1.1 \times 10^{-16}$, which is far less than our significance of $\alpha = 0.01$, supporting our rejection of the null hypothesis

Ross Lauterbach

STAT 706 HW #5

5.15 We are given the following simultaneous linear equations:

$$5Y_1 + 2Y_2 = 8$$

$$23Y_1 + 7Y_2 = 28$$

This system can be expressed as a matrix in the form AY = B as shown:

$$A = \begin{pmatrix} 5 & 2 \\ 23 & 7 \end{pmatrix} \; Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \; B = \begin{pmatrix} 8 \\ 28 \end{pmatrix}$$

To solve for Y, we can alter the above equation to get $Y = A^{-1}B$, where $A^{-1}$is the inverse matrix of A

I know that if $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ then we have $A^{-1} = \frac{1}{\det(A)}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$, so in our case

$$\det(A) = ad - bc = (5)(7) - (23)(2) = -11$$

It follows that $A^{-1} = \frac{-1}{11}\begin{pmatrix} 7 & -2 \\ -23 & 5 \end{pmatrix} = \begin{pmatrix} -\frac{7}{11} & \frac{2}{11} \\ \frac{23}{11} & -\frac{5}{11} \end{pmatrix}$

We can now calculate Y, which is given by

$$Y = A^{-1}B = \begin{pmatrix} -\frac{7}{11} & \frac{2}{11} \\ \frac{23}{11} & -\frac{5}{11} \end{pmatrix}\begin{pmatrix} 8 \\ 28 \end{pmatrix} = \begin{pmatrix} -\frac{112}{11} \\ \frac{44}{11} \end{pmatrix}$$

Therefore, $Y_1 = -\frac{112}{11}, Y_2 = \frac{44}{11}$

5.20

Given the following quadratic form: $7Y_1^2 - 8Y_1Y_2 + 8Y_2^2$

We can represent this in matrix form by taking $Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix}$ and $A = \begin{pmatrix} 7 & -4 \\ -4 & 8 \end{pmatrix}$. This means that Y^TAY will give us the quadratic form as desired, which I will show below:

$$Y^T A == \begin{pmatrix} Y_1 & Y_2 \end{pmatrix} \begin{pmatrix} 7 & -4 \\ -4 & 8 \end{pmatrix} == \begin{pmatrix} 7Y_1 - 4Y_2 & -4Y_1 + 8Y_2 \end{pmatrix}$$

Then, as desired:

$$Y^T A Y = \begin{pmatrix} 7Y_1 - 4Y_2 & -4Y_1 + 8Y_2 \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = 7Y_1^2 - 4Y_1Y_2 - 4Y_1Y_2 + 8Y_2^2 = 7Y_1^2 - 8Y_1Y_2 + 8Y_2^2$$

5.23

a) We can find estimated regression coefficients with the following formula $b = (X^T X)^{-1} X^T Y$, where b is the vector of estimated coefficients, X is the design matrix, and Y is the vector of observations. To start,

$$X^T X = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 8.0 & 4.0 & 0.0 & -4.0 & -8.0 \end{pmatrix} \begin{pmatrix} 1 & 8.0 \\ 1 & 4.0 \\ 1 & 0.0 \\ 1 & -4.0 \\ 1 & -8.0 \end{pmatrix} = \begin{pmatrix} 5 & 0 \\ 0 & 160 \end{pmatrix}$$

$$X^T Y = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 8.0 & 4.0 & 0.0 & -4.0 & -5.0 \end{pmatrix} \begin{pmatrix} 7.5 \\ 9.0 \\ 10.2 \\ 11.0 \\ 11.7 \end{pmatrix} = \begin{pmatrix} 49.4 \\ -64.0 \end{pmatrix} \text{ and}$$

$$(X^T X)^{-1} = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.00625 \end{pmatrix}$$

Therefore $b = (X^T X)^{-1} X^T Y = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.00625 \end{pmatrix} \begin{pmatrix} 49.4 \\ -64.0 \end{pmatrix} = \begin{pmatrix} 9.88 \\ -0.26 \end{pmatrix}$

This means $b_0 = 9.88 \ and \ b_1 = -0.4$

The residuals are given by $e = Y - Xb$. It follows that, in this case:

$$e = \begin{pmatrix} 7.5 \\ 9.0 \\ 10.2 \\ 11.0 \\ 11.7 \end{pmatrix} - \begin{pmatrix} 1 & 8.0 \\ 1 & 4.0 \\ 1 & 0.0 \\ 1 & -4.0 \\ 1 & -8.0 \end{pmatrix} \begin{pmatrix} 9.88 \\ -0.26 \end{pmatrix} = \begin{pmatrix} 7.5 \\ 9.0 \\ 10.2 \\ 11.0 \\ 11.7 \end{pmatrix} - \begin{pmatrix} 7.8 \\ 8.84 \\ 9.88 \\ 10.92 \\ 11.96 \end{pmatrix} = \begin{pmatrix} -0.30 \\ 0.16 \\ 0.32 \\ 0.08 \\ -0.26 \end{pmatrix}$$

In matrix form, SSR can be simplified to the following:

$$SSR = b^T(X^TY) - n\bar{Y}^2 = (9.88 \quad -0.26)\begin{pmatrix}49.4\\-64.0\end{pmatrix} - 5(9.88)^2 = 16.64$$

And SSE is given by

$$SSE = e^Te = \begin{pmatrix}-0.30\\0.16\\0.32\\0.08\\-0.26\end{pmatrix}(-0.30 \quad 0.16 \quad 0.32 \quad 0.08 \quad -0.26) = 0.292$$

To get the variance covariance matrix of b, we ue the following formula:

$$Var(b) = s^2(X^TX)^{-1} = \frac{SSE}{n-p}(X^TX)^{-1} = \frac{0.292}{5-2}\begin{pmatrix}0.2 & 0\\0 & 0.00625\end{pmatrix} = \begin{pmatrix}0.0195 & 0\\0 & 0.0006\end{pmatrix}$$

This tells us that $Var(b_1) = 0.0195\ and\ Var(b_2) = 0.0006$

For a prediction at X = -6, we simply plug into our model:

$$\hat{Y} = b_0 + b_1X = 9.88 + (-0.26)(-6) = 11.44$$

This tells us our prediction for number of weeks before flavor deterioration given a storage temperature of -6 degrees Fahrenheit is 11.44 weeks. To get the variance of this prediction, we use the following formula:

$Var(\hat{Y}) = s^2(1 + X_{new}^T(X^TX)^{-1}X_{new}$ where in this case, $X_{new} = \begin{pmatrix}1\\6\end{pmatrix}$, therefore

$Var(\hat{Y}) = s^2(1 + X_{new}^T(X^TX)^{-1}X_{new} = \frac{0.292}{5-2}(1 + (1 \quad 6)\begin{pmatrix}0.2 & 0\\0 & 0.00625\end{pmatrix}\begin{pmatrix}1\\6\end{pmatrix}$) = 0.1387

b) The X levels in this dataset are equally spaced and centered around 0. One simplification that is a result of this is the sum and mean of X is equal to 0. This is also why the matrix X^TX is diagonal, because each non-diagonal entry contains a scalar multiple of the mean of X, which is 0. Finally, this simplifies predictions, because our predictions will be centered around the mean of our observations, Y. We saw this before, when our fitted value matrix was $\begin{pmatrix} 7.8 \\ 8.84 \\ 9.88 \\ 10.92 \\ 11.96 \end{pmatrix}$

c) The hat matrix H is given by

$$H = X(X^T X)^{-1} X^T = \begin{pmatrix} 1 & 8.0 \\ 1 & 4.0 \\ 1 & 0.0 \\ 1 & -4.0 \\ 1 & -8.0 \end{pmatrix} \begin{pmatrix} 0.2 & 0 \\ 0 & 0.00625 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 8.0 & 4.0 & 0.0 & -4.0 & -8.0 \end{pmatrix}$$

$$= \begin{pmatrix} 0.6 & 0.4 & 0.2 & 0 & -0.2 \\ 0.4 & 0.3 & 0.2 & 0.1 & 0 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0 & 0.1 & 0.2 & 0.3 & 0.4 \\ -0.2 & 0 & 0.2 & 0.4 & 0.6 \end{pmatrix}$$

d) We can calculate the variance of residuals using the following formula:

$$s^2(e) = \frac{SSE}{n-p} = \frac{0.292}{5-2} = 0.0973$$

5.27

We want to find the expectation vector for residuals. We know residuals are given by

$e = Y - Xb$, where X only consists of one column of predictorsvsince this is a through the origin model. We can take the expectation of both sides

$$E[e] = E[Y - Xb] = E[Y] - E[Xb]$$

We know through properties of linear models that E[Y] = Xb. We also know that E[Xb] = Xb, since the X levels and regression coefficients are not random variables. Therefore,

$E[e] = E[Y - Xb] = E[Y] - E[Xb] = Xb - Xb = 0$, so $E[e] = 0$

Derive the inverse of X’X

We have $X = \begin{pmatrix} 1 & X_1 \\ \vdots & \vdots \\ 1 & X_n \end{pmatrix}$ and $X^T = \begin{pmatrix} 1 & \cdots & 1 \\ X_1 & \cdots & X_n \end{pmatrix}$

Therefore $X^T X = \begin{pmatrix} 1 & \cdots & 1 \\ X_1 & \cdots & X_n \end{pmatrix} \begin{pmatrix} 1 & X_1 \\ \vdots & \vdots \\ 1 & X_n \end{pmatrix} =$
$\begin{pmatrix} 1 + \cdots + 1 & X_1 * 1 + \cdots + X_n * 1 \\ X_1 * 1 + \cdots + X_n * 1 & X_1 * X_1 + \cdots + X_n * X_n \end{pmatrix} = \begin{pmatrix} n & \sum X_i \\ \sum X_i & \sum X_i^2 \end{pmatrix}$

To find the inverse, we must take the determinant:

$$Det(X^T X) = n \sum {X_i}^2 - \left(\sum X_i\right)^2$$

Therefore, $(X^T X)^{-1} = \frac{1}{n \sum {X_i}^2 - (\sum X_i)^2} \begin{pmatrix} \sum X_i^2 & = \sum X_i \\ - \sum X_i & n \end{pmatrix}$ given what we know about determinants

Prove the H matrix is symmetric

From before, the hat matrix H was given as $H = X(X^T X)^{-1} X^T$

As can be shown above, the design matrix X will be n x 2 dimensional, meaning that its transposed will be 2 x n dimensional. This implies that the product of these two, $X^T X$, which multiples a 2 x n by an n x 2, will result in a 2 x 2 matrix. Taking the inverse of a matrix does not change its dimensions, so $(X^T X)^{-1}$ will also be a 2 x 2 matrix. To start the calculation for H, we calculate $X(X^T X)^{-1}$, which multiples an n x 2 matrix by a 2 x 2 matrix, which will result in another n x 2 matrix. When we take this and multiply by the matrix X^T we get out final desired answer $X(X^T X)^{-1} X^T$. This calculation involves multiplying an n x 2 matrix by a 2 x n matrix, so the result will always be an n x n matrix. Therefore, the hat matrix H will always be a symmetric matrix of dimensions n x n, where n is the number of observations in the dataset.