Correlation and Regression Report

CS51: Formal Analyses
Minerva University

Prof. Volkan
Jan 30, 2022

1. Introduction

This paper analyzed the dataset on North Carolina Births. To answer “Does the number of gestational weeks affect a baby’s weight?” I analyzed two variables: gestational birth weeks and the baby’s weight. Through a linear regression model, I evaluated if there is a relationship between the variables—important since premature babies can have serious sequelae for life.

2. Dataset

The dataset is from OpenIntro, a platform that offers books on mathematics. The dataset was built from a random sample of 150 babies in North Carolina, where 50 had mothers who smoked and the rest did not. The database has information such as the baby’s father and mother’s age, and weight gained by the mother during pregnancy.

The gestational birth week is a quantitative, discrete variable since it is a number that can be represented in a finite enumerable way, and an independent variable since we are analyzing its influence on the baby’s weight. The baby’s weight is a quantitative, continuous variable since we can change the precision with which we measure it, and a dependent variable, being analyzed to infer if it is affected by the number of gestational weeks[1].

Based on this information the following research question and hypotheses were drawn:

Research Question: “Does the number of gestational weeks affect a baby’s weight?

[1] **#variables:** I identified and classified the used variables on the paper, defining their characteristics such as dependent and independent nature, explaining the possible relation between both and how one may affect the other.

H0: There is no linear relationship between the babies' weight and the number of pregnancy weeks (ß = 0)

HA: There is some linear relationship between the babies' weight and the number of pregnancy weeks (ß != 0)

The slope of the regression line will serve to test the hypotheses.

3. Methods

Summary statistics

Table 1: Summary statistics comparing the weight of premature and non-premature babies

	Weight of premature babies	Weight of non-premature babies
Mean	$\bar{x}_1 = 4.134$	$\bar{x}_2 = 7.393$
Median	4.5	7.44
Standard Deviation	$s_1 = 1.567$	$s_2 = 1.032$

From the summary statistics (Appendix A), we can infer from the mean that non-premature babies weigh more at birth than premature, with little deviation—observed by the calculated standard deviations.
The histograms (Appendix A) show non-premature babies tend to weigh between 6 and 9 pounds, while premature concentrate around 5 pounds. There is slight left-skew in both histograms, since the median is greater than the mean in both cases.

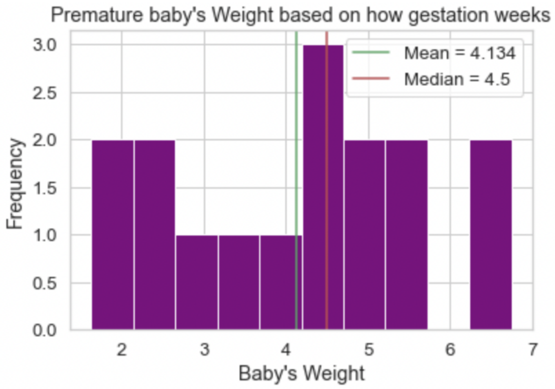


Figure 1: Histogram of premature babies' weight. The histogram shows data slightly skewed to the left.

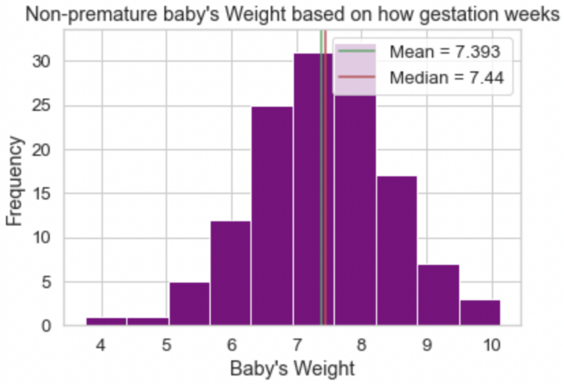


Figure 2: Histogram of non-premature babies' weight. We can also see the data slightly skewed to the left.

Correlation

To analyze the correlation between the variables, Pearson's correlation coefficient was calculated. The correlation coefficient is a quantitative measure of association—it tells us whether a scatterplot tilts up or down, and how tightly the data cluster around a straight line—it is a measure of linear association or how nearly a scatterplot follows a straight line. Based on r being 0.683 we can infer there is a considerable positive correlation between the variables (Appendix B) following the formula:

$$r = \frac{1}{n}\Sigma\left(\frac{x_i - \bar{x}}{S_x}\right)\left(\frac{y_i - \bar{y}}{S_y}\right)$$

The summation is subtracting the x and y data points by its corresponding means while dividing the results by their standard deviations and finally multiplying the results and dividing it by the sample size. The standard deviations here are the ones corresponding to the number of weeks and babies' weight[2] (Appendix B).

Regression

In a simple regression we use X to predict the Y value, X being the predictor variable, and Y the response. In Figure 3, the observations above the regression line are positive residuals and the ones below the regression line are negative residuals (Appendix C).

Correlation between gestational weeks and babies' weight

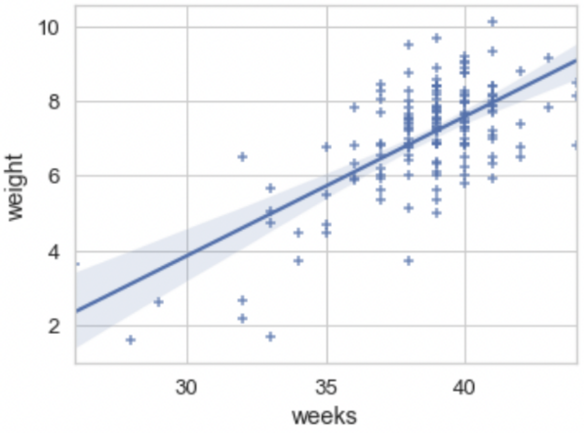


Figure 3: Correlation between gestational weeks and babies' weight. The graph shows a linear tendency and the data points.

Residuals are the leftover variation in the data after counting for the regression line.

DATA = FIT + RESIDUAL

[2] **#correlation:** I accurately computed and interpreted the correlation coefficient, providing a detailed explanation based on the dataset context later on differentiating between correlation and causation, while also identifying extraneous variables that could affect any links between the variables and explaining it.

To find the best fitting regression line, we want a line that has the least residuals. A common approach is choosing the line that minimizes the sum of the squared residuals (least squares line). To build it, we need to meet the following conditions:

1. Linearity: the data needs to show a linear trend.
2. Nearly normal residuals: the residuals' distribution needs to be nearly normal.
3. Constant variability: the variability of data around the least squares line needs to be roughly constant.
4. Independent observations: a rule of thumb to follow is guaranteeing the data analyzed represents less than 10% of the population.

Taking into account the dataset represents less than 10% of all the pregnant humans' population, and as we can observe in Figure 3 the data shows a linear trend with the datapoints somehow clustering together and following similar patterns; in Figure 4 (Appendix C) a near-normal distribution of the residuals since the residuals are centered around zero on the x-axis; and finally in Figure 5 a roughly constant distribution of the dataset around the least squared lines sharing a similar variance between themselves based on homoscedasticity (when the variance of Y is roughly the same for each data point), we can proceed with the regression analysis (Appendix D).

Normality of residuals (distribution)

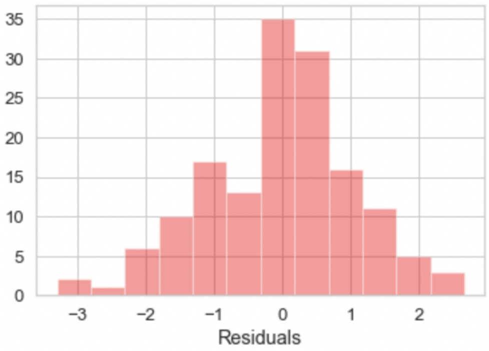


Figure 4: Normality of residuals (distribution). The distribution is almost normal with data centered around the x-axis zero[3].

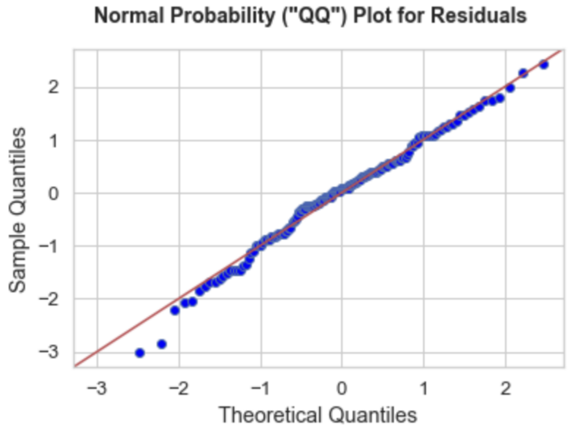


Figure 5: Normal Probability (QQ) Plot for Residuals. Most of the data points (residuals) follow the drawn line, indicating an almost equal variance.

Respecting those conditions, it is expected a regression line between both variables, asserting its correlation. R-squared then can be calculated using the following formula[4]:

$$r^2 = 1 - \frac{Error\ sum\ of\ squares\ (SSE)}{Total\ sum\ of\ square\ (SSTO)}$$

SSE quantifies how much data points Y vary around the estimated regression line $\hat{y}$. SSTO quantifies how much the data points y vary around their mean $\hat{y}$.

R-squared tells how accurate our model's predictions are—represented as a proportion (a number between 0 and 1). When the value gets closer to 1, it means the predictor variable accounts for the variation in the

[3] **#dataviz:** I interpreted, analyzed, and created a befitting data visualization that contains what I want to explain. I appropriately chose the types of graphs that best showcased the data and provided the needed information and interpretation about them.

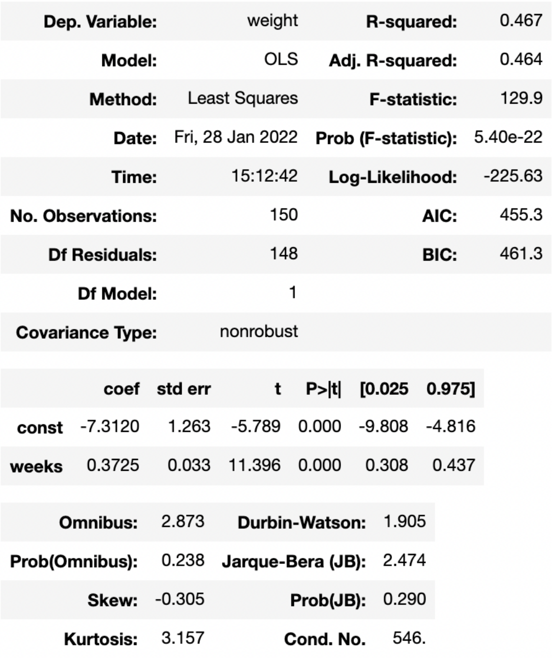
[4] **#organization:** I organized the paper following a well-established structure, explaining further needed points and visually presenting them in a way that would be easier to understand and interpret.

response variable better—the contrary happens when the value gets closer to 0. In this scenario, we have an R-squared of 0.467, meaning 46.7% of the data can be explained through the model:

WEIGHT (pounds) = 0.372 * WEEKS - 7.312

0.372 is the slope coefficient of the model, and - 7.312 is a constant (the value the model will assume when weeks = 0). Taking that into account, there is an increase of 0.372 per week[5] (Appendix D).

Table 2: OLS Regression Results



Dep. Variable:	weight	**R-squared:**	0.467
Model:	OLS	**Adj. R-squared:**	0.464
Method:	Least Squares	**F-statistic:**	129.9
Date:	Fri, 28 Jan 2022	**Prob (F-statistic):**	5.40e-22
Time:	15:12:42	**Log-Likelihood:**	-225.63
No. Observations:	150	**AIC:**	455.3
Df Residuals:	148	**BIC:**	461.3
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-7.3120	1.263	-5.789	0.000	-9.808	-4.816
weeks	0.3725	0.033	11.396	0.000	0.308	0.437

Omnibus:	2.873	**Durbin-Watson:**	1.905
Prob(Omnibus):	0.238	**Jarque-Bera (JB):**	2.474
Skew:	-0.305	**Prob(JB):**	0.290
Kurtosis:	3.157	**Cond. No.**	546.

P-value

The p-value serves to indicate the statistical significance of the regression's slope. Here, a 0.05 significance level was used—a good balance between committing a Type I or Type II error. Defining this, the resulting t-score was 11.39, leading to a two-tailed p-value of almost 0 (Appendix E).

[5] **#regression:** I constructed an interpreted regression model, explaining the relation between the dependent and independent variable, while also computing and explaining the regression formula, explaining the prediction results.

The p-value means analyzing the probability we can observe a t-score value greater or equal to the calculated value when the null hypothesis is true, and if the p-value is lower than the significance level, we proceed to reject the null hypothesis. In our model, the p-value is close enough to 0, indicating the null hypothesis can be rejected in favor of the alternative hypothesis[6].

4. Results and Conclusions

Through the slope analysis, we accept the alternative hypothesis that there is some linear relationship between the baby's weight and the number of pregnancy weeks since ß is not zero. From the football shape of Figure 3 and the r-squared results, we can infer correlation between the variables.

Through the regression model, we can predict 46.7% of the babies' weights correctly, improving this percentage by changing the model's coefficient and constant based on the confidence interval defined in the study.

Inductively on a greater scale, we can make inferences about babies in general and their conditions depending on how many gestational weeks they have, since we have strong premises as our regression model and significance analysis to draw this conclusion. We can affirm with some level of confidence that babies will be born with less weight when they are premature. This can be demonstrated by the model, but also explained by biology. It is logical to think babies that have longer gestational periods receive more nutrients, develop organs and structures further than premature babies. However, a non-premature baby can still be born with lower than normal weight if their generator does not offer them enough nutrients or in cases where the baby has an illness (extraneous variables). Also, premature babies can be born with a good weight, especially those being generated by a healthy person and that were born close to what is

[6] **#significance:** To assess the significance of the test the p-value and other significance tests were drawn. I explained what the results meant given the context, adequating it to a regression analysis.

considered a normal gestational period. Based on that, we can infer in most cases there is a causal relationship between gestational week number and the weight of a newborn[7].

5. Reflection

During the course I learned when to test hypotheses and how to analyze data and draw information from it. Combining correlation analysis and the evaluation of statistical significance I have more tools to prove the strength of a hypothesis, showing data about its correlative nature and statistical significance, maybe inferring causal relationships[8].

Word count: 1404 words.

[7] **#induction:** I analyzed the premisses that led to the overall conclusion about the population, pointing the strong link we could draw but also possible pitfalls given the context.

[8] **#professionalism:** I communicated and presented my work appropriately, following the advised template and APA rules, following nuanced conventions of a report.

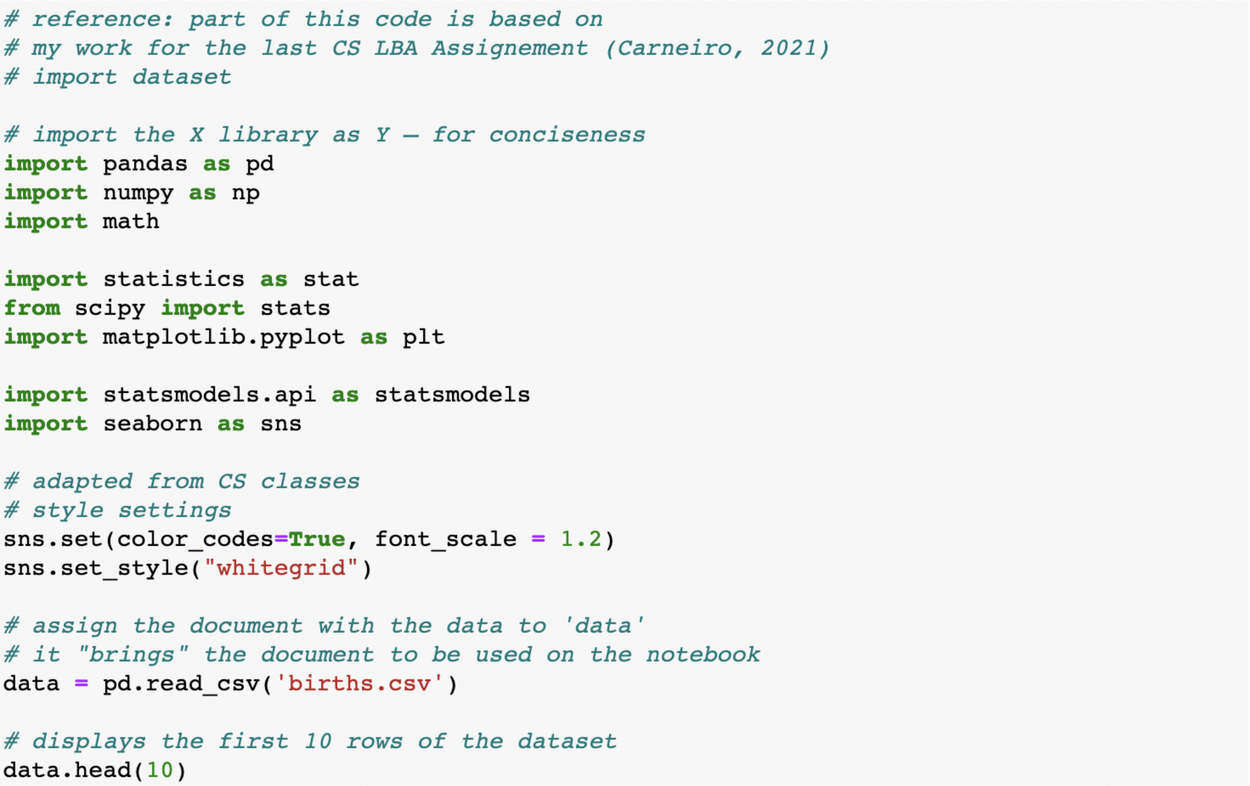
6. References

Carneiro, Maria. (2021, Dec). *Statistical Inferences.* [Unpublished assignment submitted for CS50]. Minerva University

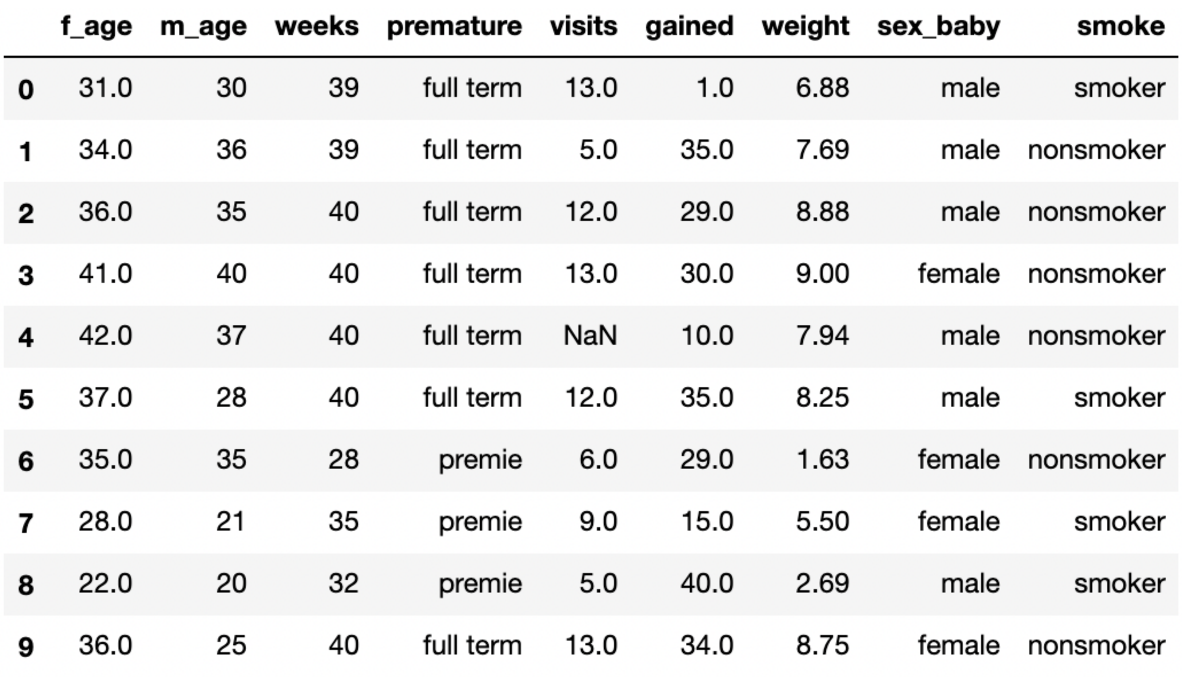
Data Sets. (n.d.). OpenIntro. https://www.openintro.org/data/index.php?data=births

7. Appendix

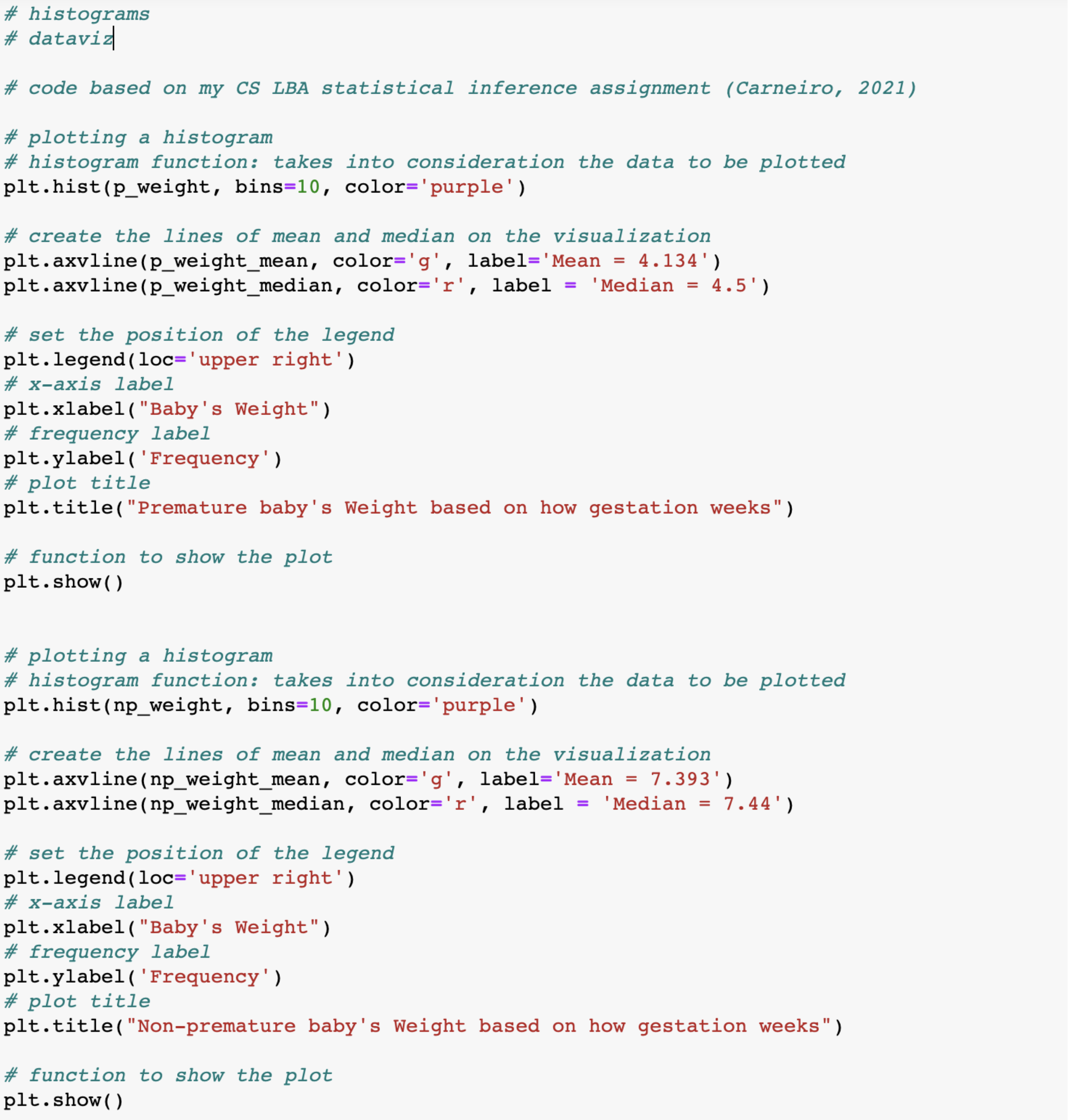
Appendix A



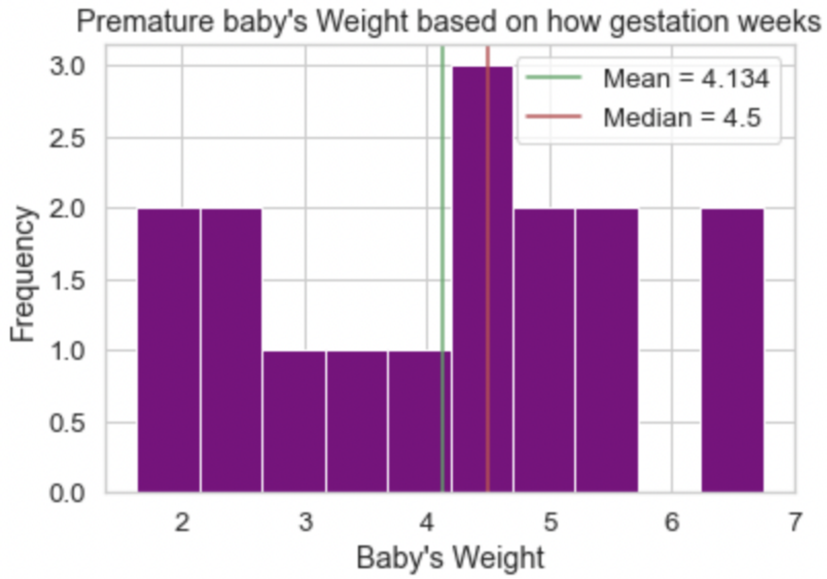
```
# reference: part of this code is based on
# my work for the last CS LBA Assignement (Carneiro, 2021)
# import dataset

# import the X library as Y — for conciseness
import pandas as pd
import numpy as np
import math

import statistics as stat
from scipy import stats
import matplotlib.pyplot as plt

import statsmodels.api as statsmodels
import seaborn as sns

# adapted from CS classes
# style settings
sns.set(color_codes=True, font_scale = 1.2)
sns.set_style("whitegrid")

# assign the document with the data to 'data'
# it "brings" the document to be used on the notebook
data = pd.read_csv('births.csv')

# displays the first 10 rows of the dataset
data.head(10)
```

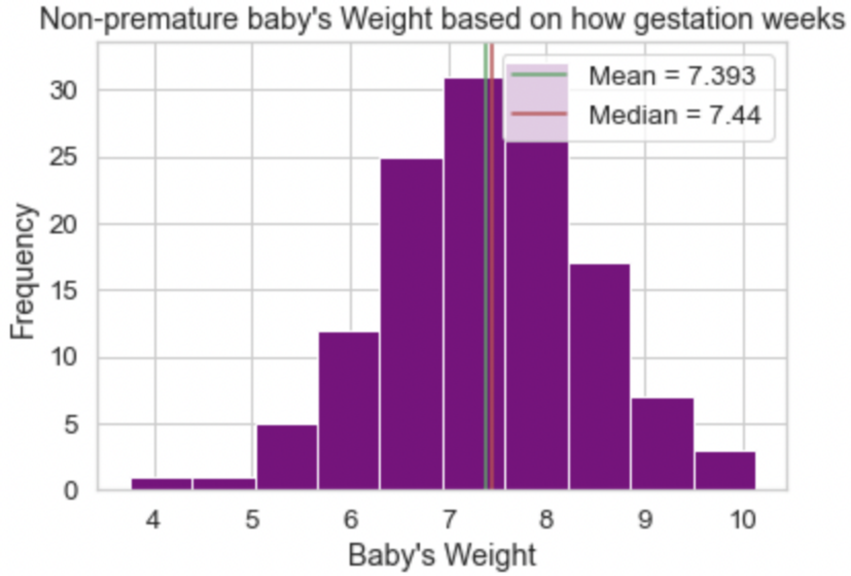


	f_age	m_age	weeks	premature	visits	gained	weight	sex_baby	smoke
0	31.0	30	39	full term	13.0	1.0	6.88	male	smoker
1	34.0	36	39	full term	5.0	35.0	7.69	male	nonsmoker
2	36.0	35	40	full term	12.0	29.0	8.88	male	nonsmoker
3	41.0	40	40	full term	13.0	30.0	9.00	female	nonsmoker
4	42.0	37	40	full term	NaN	10.0	7.94	male	nonsmoker
5	37.0	28	40	full term	12.0	35.0	8.25	male	smoker
6	35.0	35	28	premie	6.0	29.0	1.63	female	nonsmoker
7	28.0	21	35	premie	9.0	15.0	5.50	female	smoker
8	22.0	20	32	premie	5.0	40.0	2.69	male	smoker
9	36.0	25	40	full term	13.0	34.0	8.75	female	nonsmoker

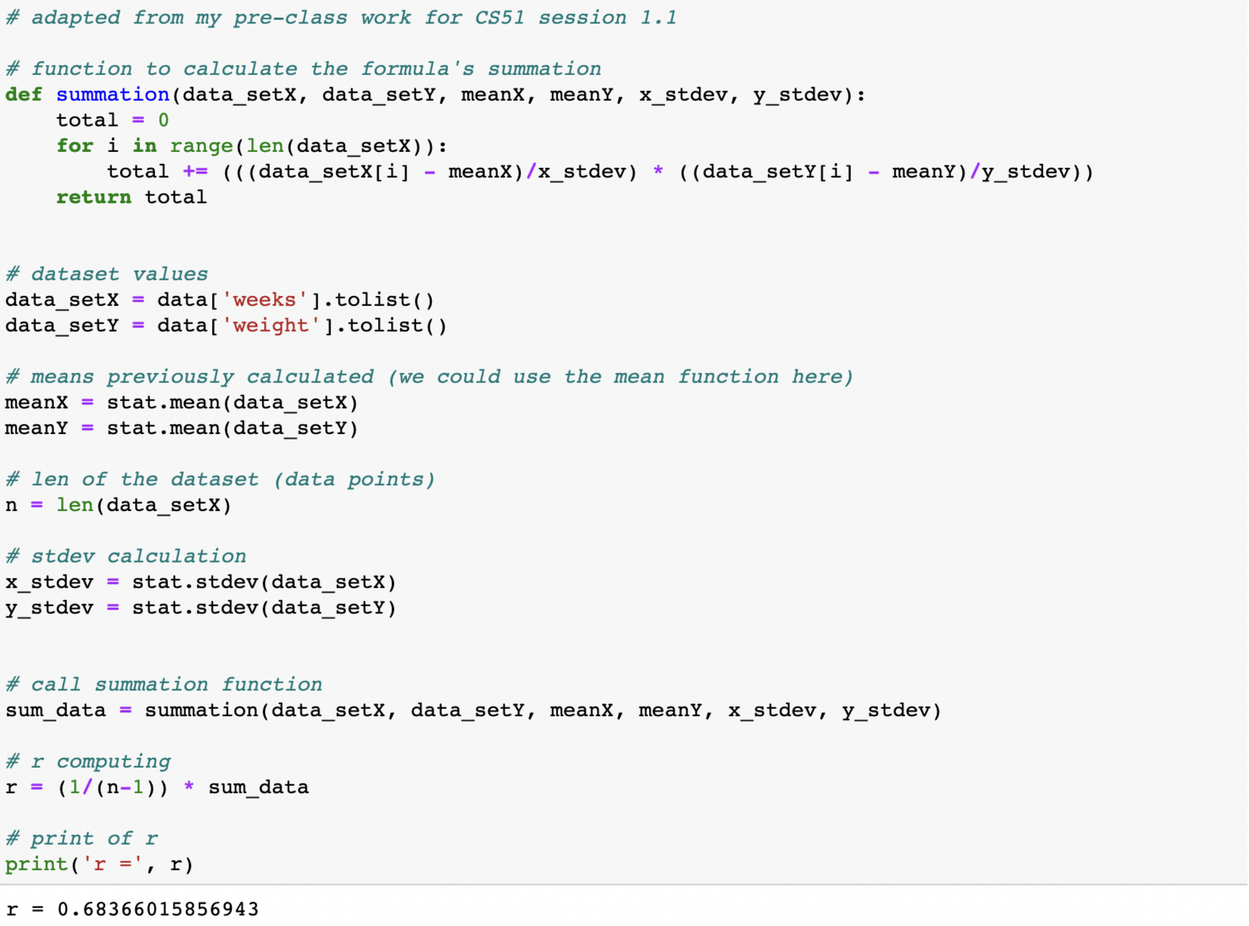


```
# histograms
# dataviz

# code based on my CS LBA statistical inference assignment (Carneiro, 2021)

# plotting a histogram
# histogram function: takes into consideration the data to be plotted
plt.hist(p_weight, bins=10, color='purple')

# create the lines of mean and median on the visualization
plt.axvline(p_weight_mean, color='g', label='Mean = 4.134')
plt.axvline(p_weight_median, color='r', label = 'Median = 4.5')

# set the position of the legend
plt.legend(loc='upper right')
# x-axis label
plt.xlabel("Baby's Weight")
# frequency label
plt.ylabel('Frequency')
# plot title
plt.title("Premature baby's Weight based on how gestation weeks")

# function to show the plot
plt.show()


# plotting a histogram
# histogram function: takes into consideration the data to be plotted
plt.hist(np_weight, bins=10, color='purple')

# create the lines of mean and median on the visualization
plt.axvline(np_weight_mean, color='g', label='Mean = 7.393')
plt.axvline(np_weight_median, color='r', label = 'Median = 7.44')

# set the position of the legend
plt.legend(loc='upper right')
# x-axis label
plt.xlabel("Baby's Weight")
# frequency label
plt.ylabel('Frequency')
# plot title
plt.title("Non-premature baby's Weight based on how gestation weeks")

# function to show the plot
plt.show()
```


Premature baby's Weight based on how gestation weeks
Mean = 4.134
Median = 4.5
Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
2
3
4
5
6
7
Baby's Weight


Non-premature baby's Weight based on how gestation weeks
Mean = 7.393
Median = 7.44
Frequency
0
5
10
15
20
25
30
4
5
6
7
8
9
10
Baby's Weight

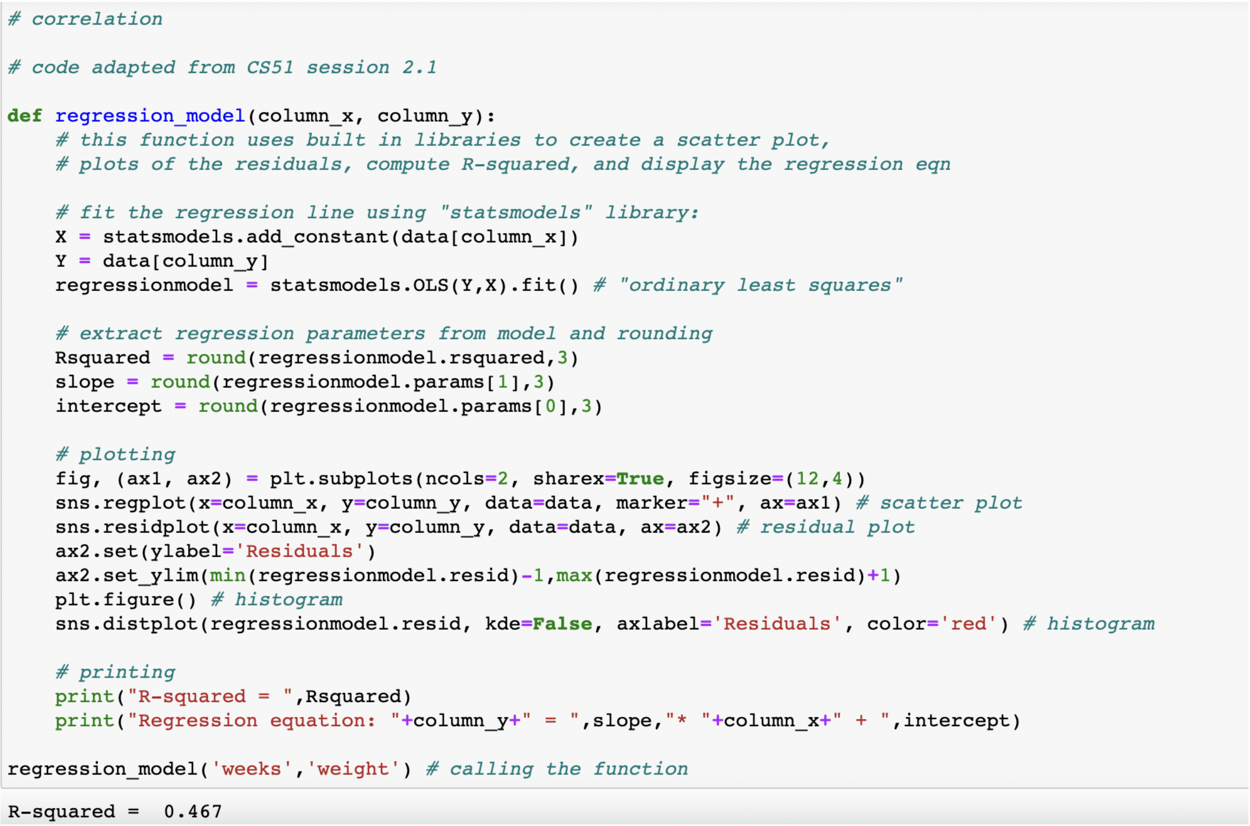
Appendix B



```
# adapted from my pre-class work for CS51 session 1.1

# function to calculate the formula's summation
def summation(data_setX, data_setY, meanX, meanY, x_stdev, y_stdev):
    total = 0
    for i in range(len(data_setX)):
        total += (((data_setX[i] - meanX)/x_stdev) * ((data_setY[i] - meanY)/y_stdev))
    return total


# dataset values
data_setX = data['weeks'].tolist()
data_setY = data['weight'].tolist()

# means previously calculated (we could use the mean function here)
meanX = stat.mean(data_setX)
meanY = stat.mean(data_setY)

# len of the dataset (data points)
n = len(data_setX)

# stdev calculation
x_stdev = stat.stdev(data_setX)
y_stdev = stat.stdev(data_setY)


# call summation function
sum_data = summation(data_setX, data_setY, meanX, meanY, x_stdev, y_stdev)

# r computing
r = (1/(n-1)) * sum_data

# print of r
print('r =', r)
```

```
r = 0.68366015856943
```

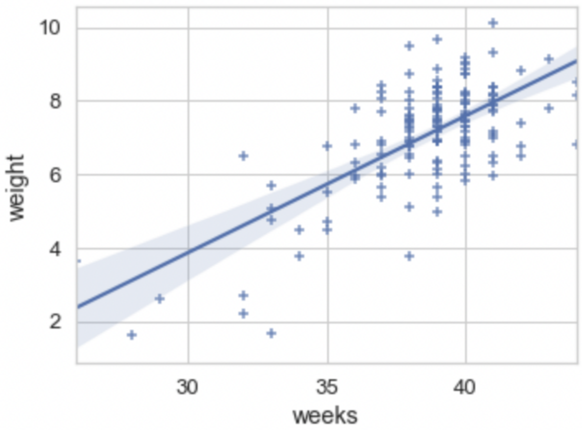
Appendix C

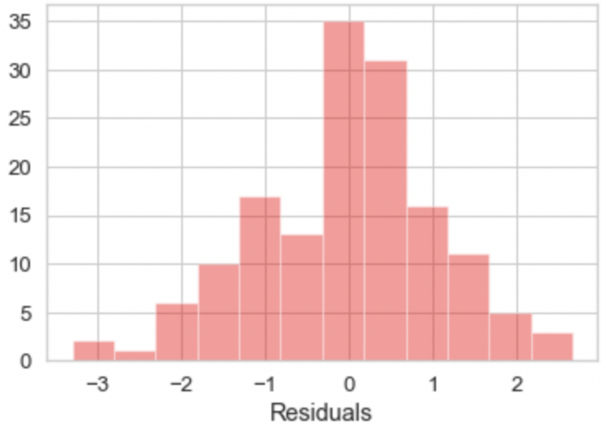


```
# correlation

# code adapted from CS51 session 2.1

def regression_model(column_x, column_y):
    # this function uses built in libraries to create a scatter plot,
    # plots of the residuals, compute R-squared, and display the regression eqn

    # fit the regression line using "statsmodels" library:
    X = statsmodels.add_constant(data[column_x])
    Y = data[column_y]
    regressionmodel = statsmodels.OLS(Y,X).fit() # "ordinary least squares"

    # extract regression parameters from model and rounding
    Rsquared = round(regressionmodel.rsquared,3)
    slope = round(regressionmodel.params[1],3)
    intercept = round(regressionmodel.params[0],3)

    # plotting
    fig, (ax1, ax2) = plt.subplots(ncols=2, sharex=True, figsize=(12,4))
    sns.regplot(x=column_x, y=column_y, data=data, marker="+", ax=ax1) # scatter plot
    sns.residplot(x=column_x, y=column_y, data=data, ax=ax2) # residual plot
    ax2.set(ylabel='Residuals')
    ax2.set_ylim(min(regressionmodel.resid)-1,max(regressionmodel.resid)+1)
    plt.figure() # histogram
    sns.distplot(regressionmodel.resid, kde=False, axlabel='Residuals', color='red') # histogram

    # printing
    print("R-squared = ",Rsquared)
    print("Regression equation: "+column_y+" = ",slope,"* "+column_x+" + ",intercept)

regression_model('weeks','weight') # calling the function
```

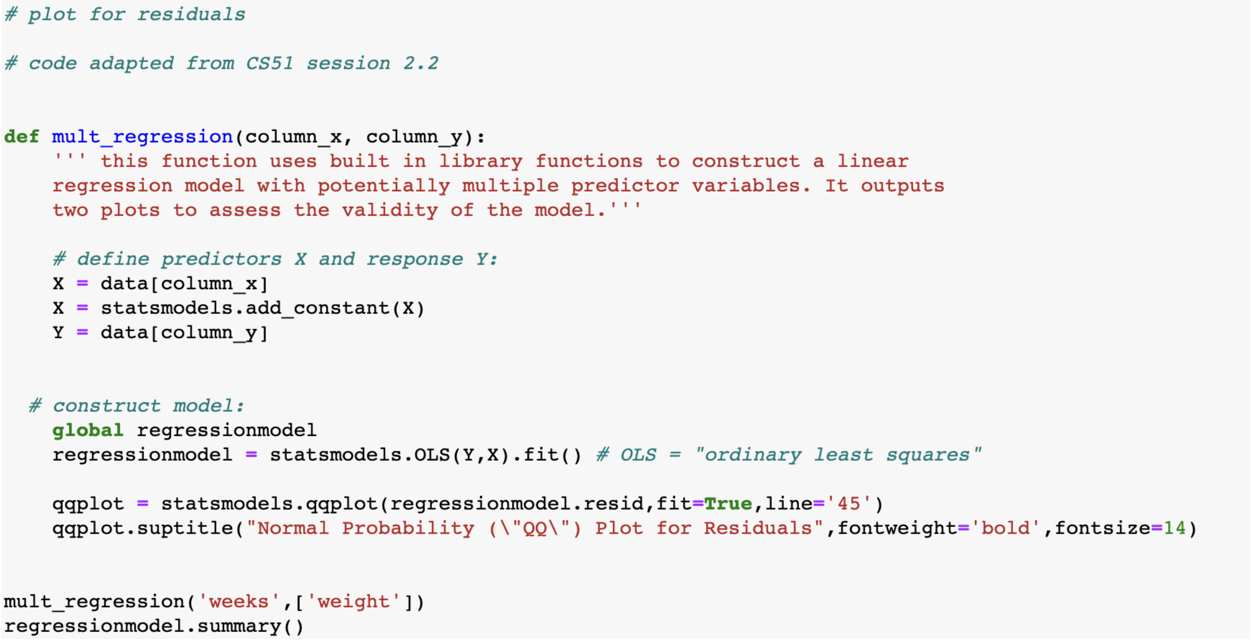
```
R-squared =  0.467
```

* The residual plot was not used in this paper

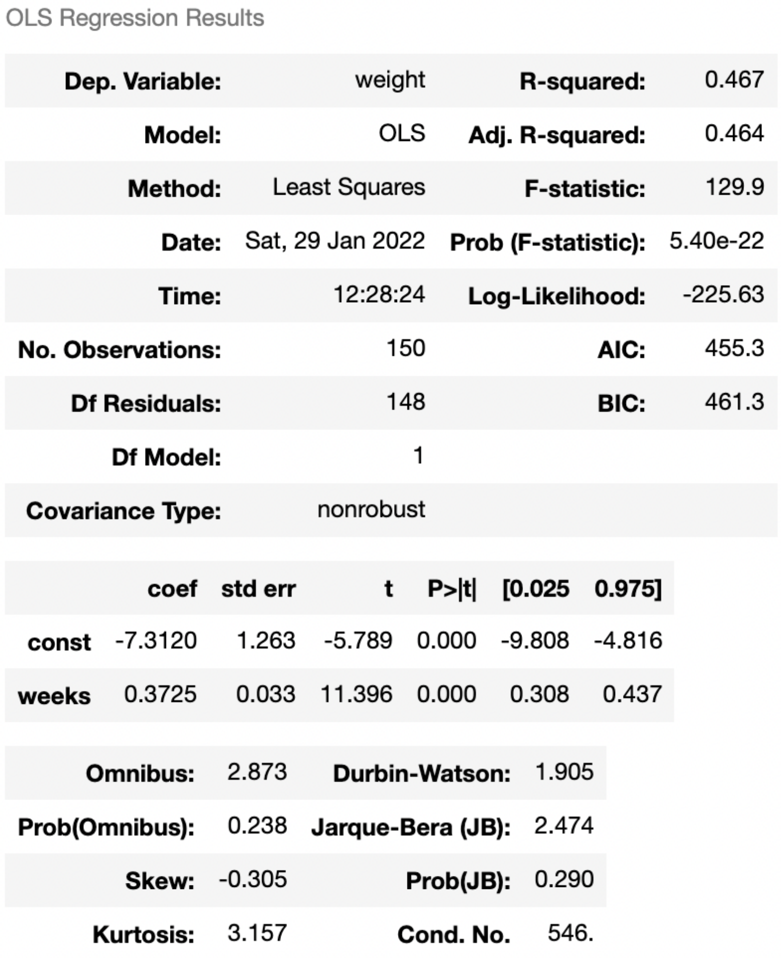

weight
10
8
6
4
2
30
35
40
weeks


35
30
25
20
15
10
5
0
−3
−2
−1
0
1
2
Residuals

Appendix D



```
# plot for residuals

# code adapted from CS51 session 2.2


def mult_regression(column_x, column_y):
    ''' this function uses built in library functions to construct a linear
    regression model with potentially multiple predictor variables. It outputs
    two plots to assess the validity of the model.'''

    # define predictors X and response Y:
    X = data[column_x]
    X = statsmodels.add_constant(X)
    Y = data[column_y]


  # construct model:
    global regressionmodel
    regressionmodel = statsmodels.OLS(Y,X).fit() # OLS = "ordinary least squares"

    qqplot = statsmodels.qqplot(regressionmodel.resid,fit=True,line='45')
    qqplot.suptitle("Normal Probability (\"QQ\") Plot for Residuals",fontweight='bold',fontsize=14)


mult_regression('weeks',['weight'])
regressionmodel.summary()
```

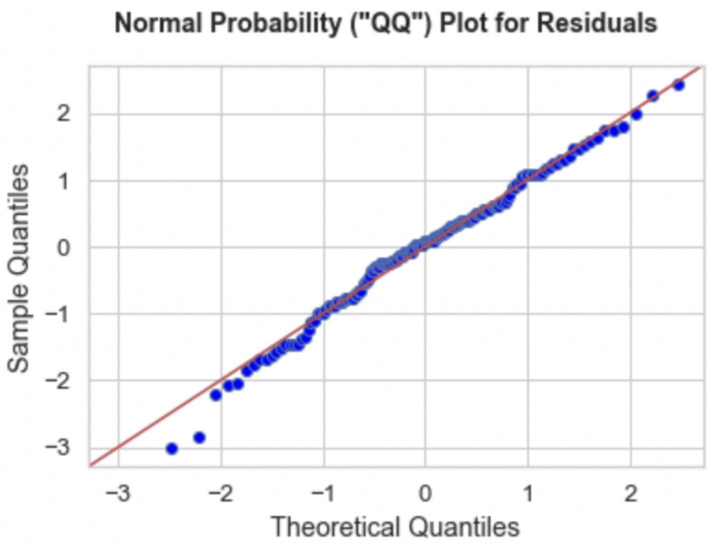


OLS Regression Results

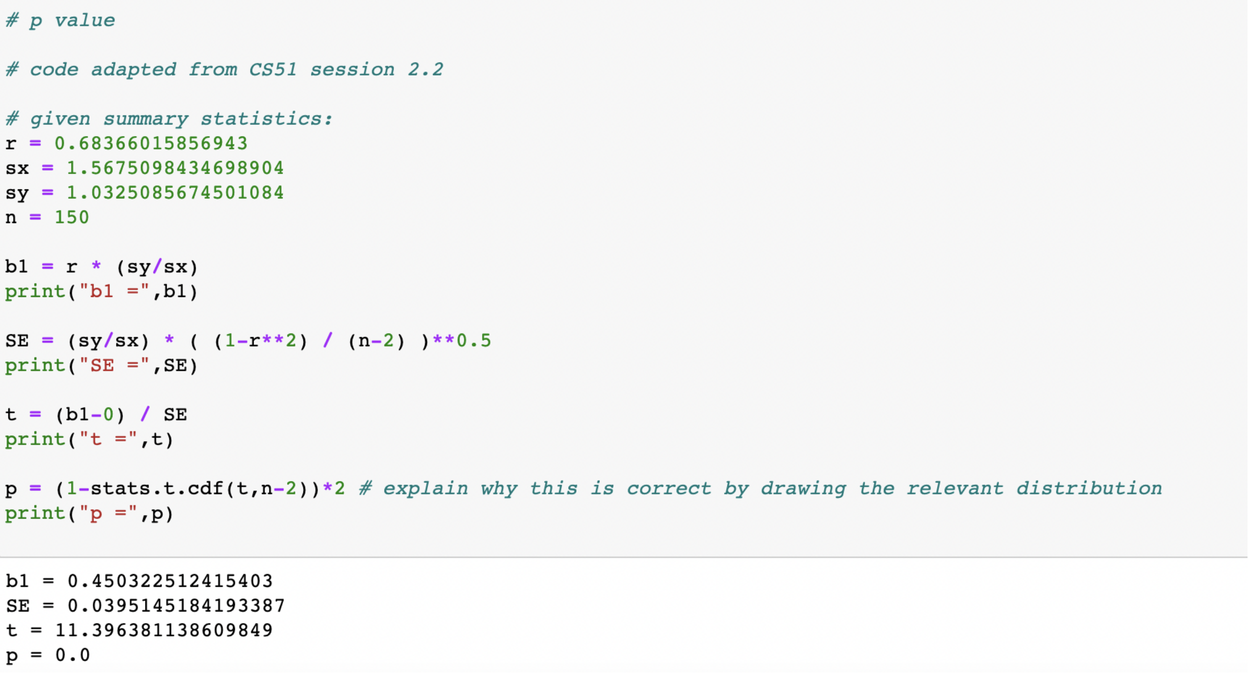
Dep. Variable:	weight	**R-squared:**	0.467
Model:	OLS	**Adj. R-squared:**	0.464
Method:	Least Squares	**F-statistic:**	129.9
Date:	Sat, 29 Jan 2022	**Prob (F-statistic):**	5.40e-22
Time:	12:28:24	**Log-Likelihood:**	-225.63
No. Observations:	150	**AIC:**	455.3
Df Residuals:	148	**BIC:**	461.3
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-7.3120	1.263	-5.789	0.000	-9.808	-4.816
weeks	0.3725	0.033	11.396	0.000	0.308	0.437

Omnibus:	2.873	**Durbin-Watson:**	1.905
Prob(Omnibus):	0.238	**Jarque-Bera (JB):**	2.474
Skew:	-0.305	**Prob(JB):**	0.290
Kurtosis:	3.157	**Cond. No.**	546.



Appendix E



```
# p value

# code adapted from CS51 session 2.2

# given summary statistics:
r = 0.68366015856943
sx = 1.5675098434698904
sy = 1.0325085674501084
n = 150

b1 = r * (sy/sx)
print("b1 =",b1)

SE = (sy/sx) * ( (1-r**2) / (n-2) )**0.5
print("SE =",SE)

t = (b1-0) / SE
print("t =",t)

p = (1-stats.t.cdf(t,n-2))*2 # explain why this is correct by drawing the relevant distribution
print("p =",p)
```

```
b1 = 0.450322512415403
SE = 0.0395145184193387
t = 11.396381138609849
p = 0.0
```