Table of Contents

[Least Squares Regression 1](#_Toc30057879)

[Slope and Intercept 1](#_Toc30057880)

[Assumptions of Least Squares Regression 1](#_Toc30057881)

[Analysis of the Variance (ANOVA) 5](#_Toc30057882)

[Degrees of Freedom 6](#_Toc30057883)

[Coefficient of Determination 10](#_Toc30057884)

[Adjusted Coefficient of Determination 11](#_Toc30057885)

[Train and Test Data 11](#_Toc30057886)

[Target and Predictor Variables 11](#_Toc30057887)

[Preventing Randomization During Test-Train Split 13](#_Toc30057888)

**Note:**

You are encouraged to work with others and you can discuss your solutions over-the-shoulder as well but DO NOT SHARE COPIES of YOUR WORK ELECTONICALLY OR IN ANY OTHER FORM. Keep good karma with your career network and protect your reputation – do not plagiarize.

## Least Squares Regression

Least squares regression offers a method to generate a best fit line through a scatter of data. The best fit line can then be used to make predictions about a target variable given a set of predictor variables. You do not need to prove or memorize the equations used to calculate and analyze least squares regression but it is important to understand how to use them.

### Slope and Intercept

The least squares equation is described with the help of **m**, the slope, and **b**,the intercept. These values are calculated with the following equations:

Equation : Slope

Equation : Y intercept

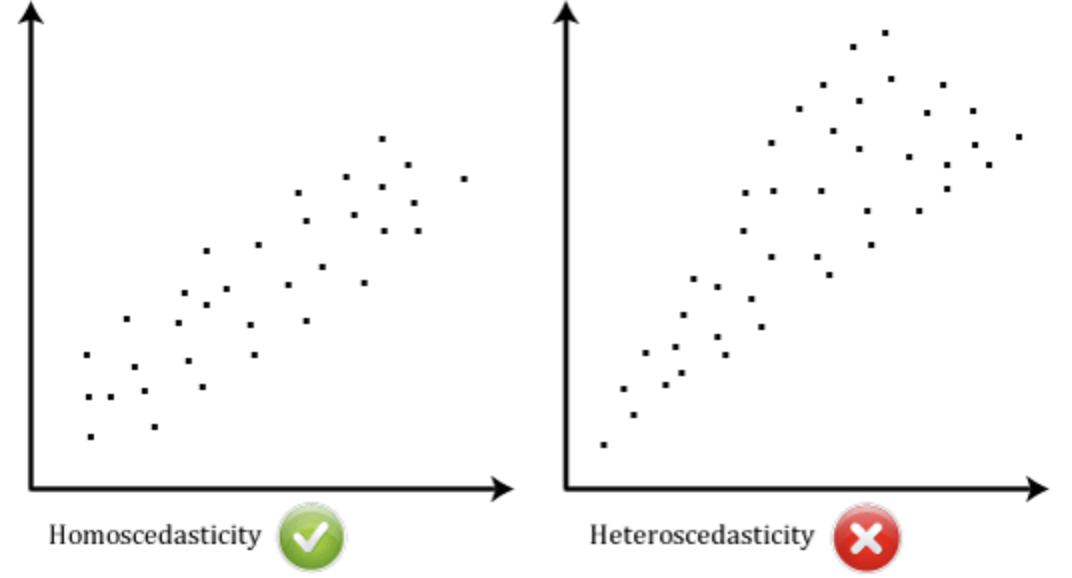
### Assumptions of Least Squares Regression

Proper least squares regression requires the residuals (error) and response variables to be:

1. **Homoscedastic**

Homoscedasticity refers to an even spread whereas heteroscedasticity is a broadening spread (see Figure 1). The diagram on the left is **homoscedastic** since the relationship between x and y is evenly spread. The diagram on the right demonstrates **heteroscedasticity** since the relationship between x and y disperses outward.

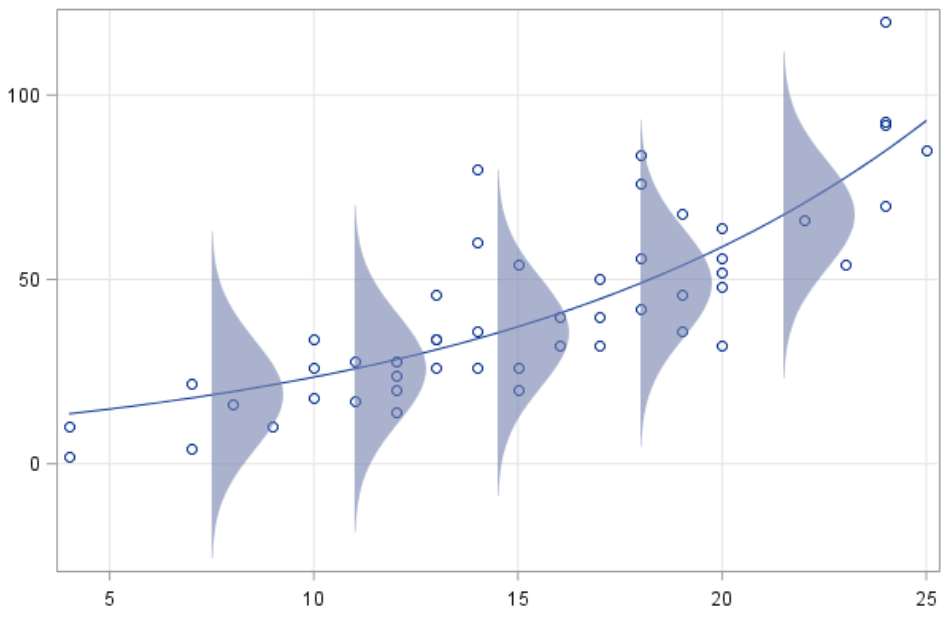
Figure : Homoscedasticity versus. Heteroscedasticity



1. **Normally distributed**

With the classic normal distribution samples are evenly spread symmetrically about the mean. The highest sample concentration is at the mean (see Figure 2). Residual errors are also distributed normally about the mean.

Figure : Normal spread of target variables and residual errors.



Example : Simple Least Squares Regression

Figure 3 shows five bacteria samples with growth (y) in grams per day (x). This example demonstrates how you can use least squares regression to generate the best fitting line equation to predict levels of bacteria growth (y) over time.

**Note: Usually a much larger sample set is needed but only five samples have been included here to make it easier to explain the calculations manually.**

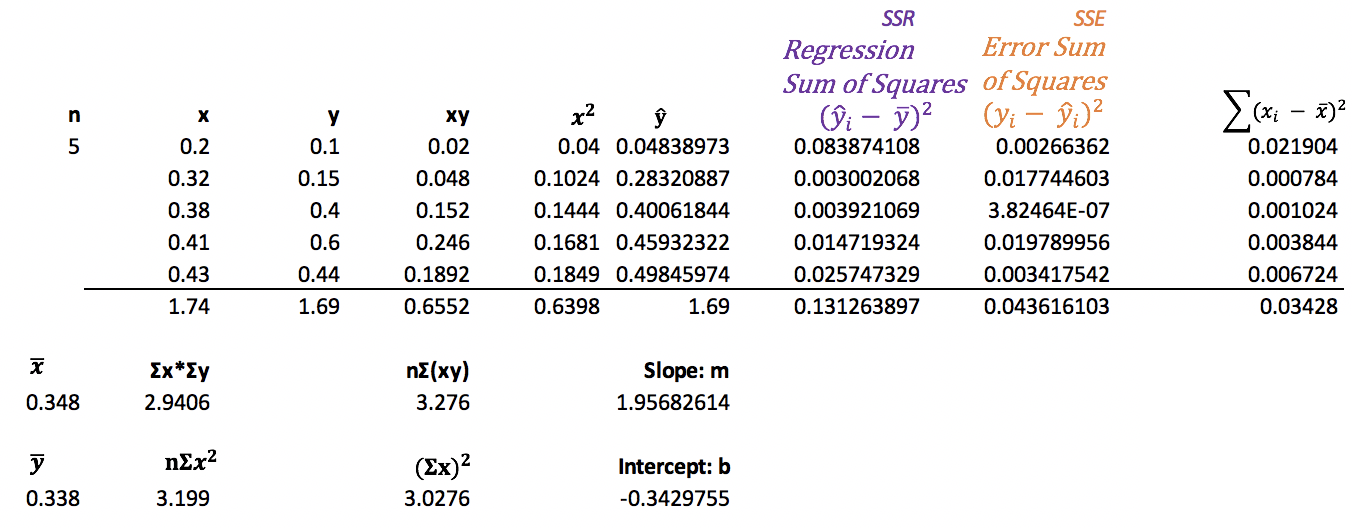
Figure : Sample Data points

|  |  |  |
| --- | --- | --- |
|  | **x (days)**  0.20  0.32  0.38  0.41  0.43 | **y**  **(bacteria)**  0.10  0.15  0.40  0.60  0.44 |

`

If we plug the values of x and y into Equation 1 we get the slope and Equation 2 gives the y-intercept when x = 0 (see Figure 4). Some other calculations have been included in Figure 4 for later.

Figure : Calculating the slope and intercept for x and y



Average x 0.348

Average y 0.338

Now that we have calculated values for ***m*** and ***b*** we can say the equation generated is:

*Predicted bacteria in grams*

Explaining the equation in simpler terms, we could say bacteria appears to grow 1.96 times each day. The intercept of -0.3430 does not make as much sense since it is not possible to have a negative growth amount.

If we plug in different values for days (x) into the straight-line equation that is generated through least squares regression we can obtain predicted levels () of bacteria (see Figure 5). The blue line is our prediction line.

Figure : Predicting y using the straight-line equation.

|  |  |
| --- | --- |
|  |  |

Here is the code used to draw the graph in Figure 5.

|  |
| --- |
| **import** matplotlib.pyplot **as** plt  *# Plot scatter of actual values.* daysX = [0.2, 0.32, 0.38, 0.41, 0.43] bacteriaY = [0.1, 0.15, 0.4, 0.6, 0.44] plt.scatter(daysX, bacteriaY, color=**'green'**, label=**'Sample Data'**)  *# Plot prediction line.* daysX2 = [0, 0.1, 0.2, 0.3, 0.4, 0.5] bacteriaY2 = [-0.3430, -0.1473, 0.0484, 0.2441, 0.4398, 0.6354] plt.plot(daysX2, bacteriaY2, color=**'blue'**, label=**'y=1.9568\*x - 0.3430'**)  *# Show average* x3 = [0, 0.5] y3 = [0.338, 0.338] plt.plot(x3, y3, **'--'**, color=**'Black'**, label=**'Average Bacteria Level'**)  *# Add a legend, axis labels, and title.* plt.legend() plt.xlabel(**"Days"**) plt.ylabel(**"Bacteria (g)"**) plt.title(**'Bacteria Growth per Day'**)  plt.show() |

Exercise (15 marks)

Using Excel, set up a spreadsheet similar to Figure 4 to manually implement least squares regression to calculate the intercept and slope of a model that predicts the y value given the following data set.

﻿ x = [0.19, 0.28, 0.35, 0.37, 0.4, 0.18]

y = [0.13, 0.12, 0.35, 0.3, 0.37, 0.1]

Show a screenshot of your spreadsheet with the required data here: **(Keep your spreadsheet handy for the next quiz.)** (6 marks)

|  |
| --- |
|  |

Show your slope here:

|  |
| --- |
|  |

Show your intercept here:

|  |
| --- |
|  |

Show your least squares regression model here: (4 marks)

|  |
| --- |
|  |

Using your model make predictions for x = [0.13, 0.22, 0.33] (5 marks)

|  |
| --- |
|  |

### Analysis of the Variance (ANOVA)

When using different forms of regression, we can use ANOVA to \*help\* determine the reliability of the prediction model in terms of variance. A high variance is undesirable since predicted values are further away from actual values. Variance is often described with comparisons that are relative to these three measures:

y = actual value

= predicted value

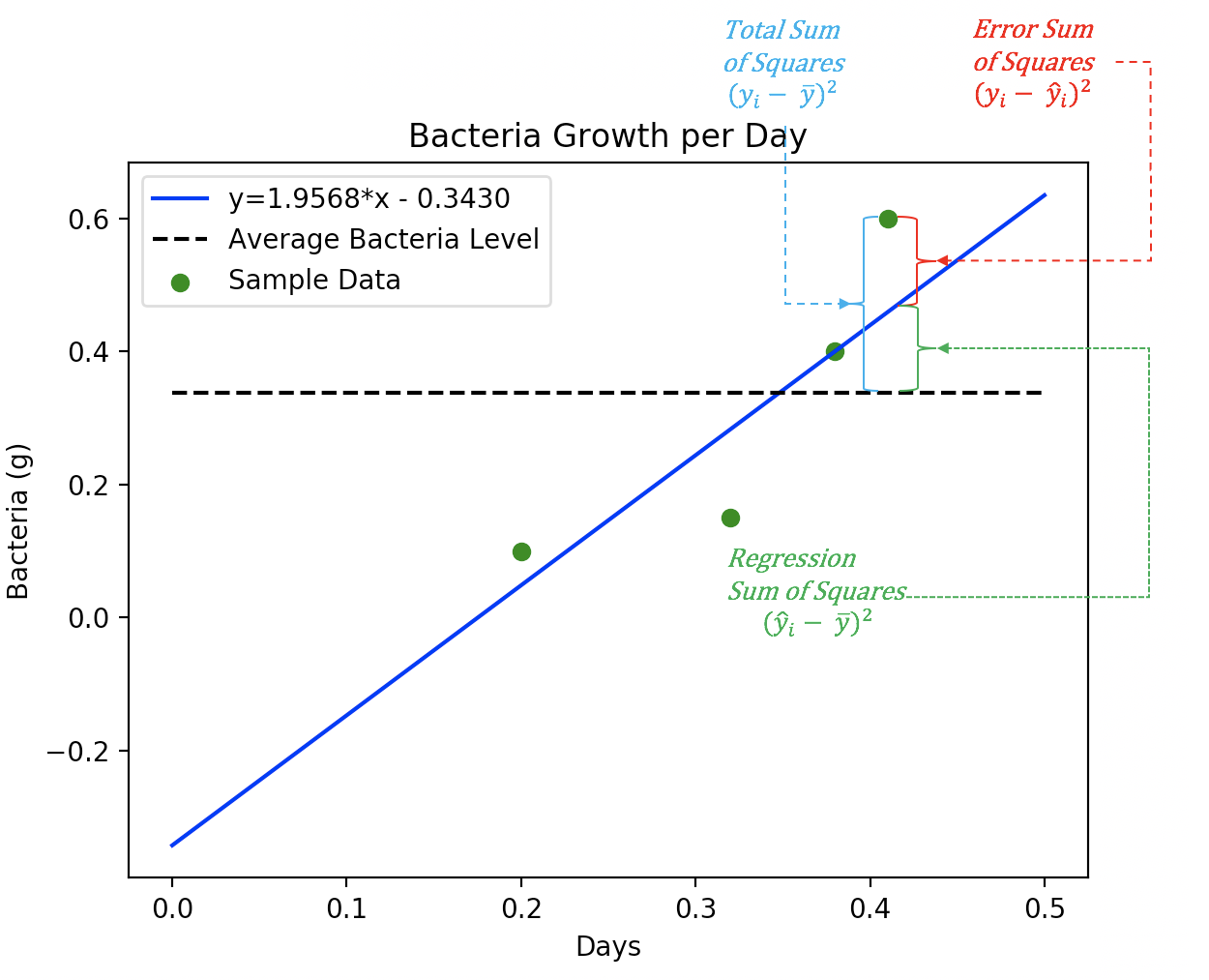
= mean

The total variance (SST) is composed of the variance explained by the predictive model (SSR) and the error variance (SSE). Variance is often summarized with these three equations.

|  |  |  |
| --- | --- | --- |
| Equation 3: SSR | Regression Sum of Squares: Variance explained by model. |  |
| Equation 4: SSE | Error Sum of Squares: Unexplained variance. |  |
| Equation 5: SST | Total Sum of Squares: Total variance of sample. | = SSR + SSE |

The total sum of squares, SST, in Figure 6 represents the variance between the actual values of the target variable and overall average in absence of a model. This total variance can then be split into SSR and SSE components. The green section in Figure 6 shows the SSR variance component which is the variance between the predicted value and the overall average. In other words, the green section represents the variance explained by the model. The red section of Figure 6 highlights the unexplained variance between the estimated and actual value. Higher amounts of green versus red are preferred.

Figure : Analysis of Variance for Least Squares Regression



### Degrees of Freedom

We use degrees of freedom to account for a reduced sample size and the number of predictor variables. Small sample sizes and higher numbers of predictor variables make the model less reliable.

#### Model DF: Total Number of Variables

Having too many variables can lead to overfitting which in turn leads to unreliable results. The model DF adds a penalty for each predictor variable used in the model. For Example 1 we have one predictor variable, *x*, so the model DF is 1.

**Model DF = # of model parameters = 1**

(Does not include intercept)

#### Error DF:

The error DF represents the variance which is not explained by the model.

Error DF = N – Model DF – 1

(Includes the intercept)

#### F Statistic

The F Statistic is the ratio mean square of the variance explained by the model divided by model degrees of freedom over the mean square of the error residuals divided by error DF.

*Equation 6:*

A higher proportion of explained variance over unexplained variance is obviously desirable so a large F statistic is preferred. Notice how the degrees of freedom can shift the placement of the F statistic.

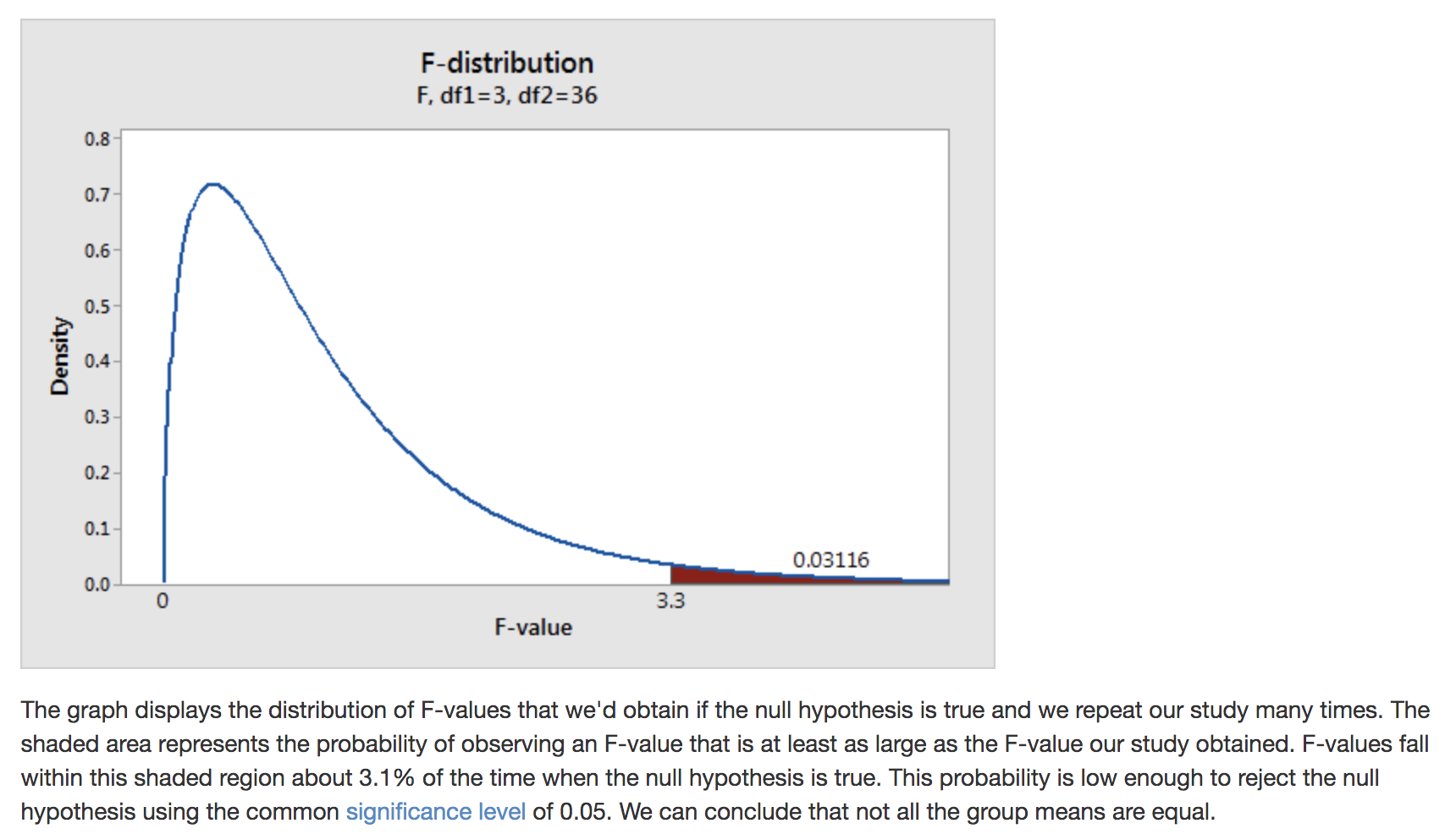
#### Null Hypothesis, , for Slope Coefficients

For least squares regression the F statistic is used to validate the null hypothesis, that the slope coefficients do not help to predict the target outcome. A rejection of is desired. The rejection of the null hypothesis suggests that the slope coefficients are not zero and are therefore helpful in predicting outcomes. For example, the prediction line:

y = mx + b is only useful if m 0.

The F statistic distribution is right skewed. **F** values in the red-shaded region is desirable since it indicates a rejection of (see Figure 7).

Figure : F-distribution



Example : Validating the Slope Coefficient When = 0.05

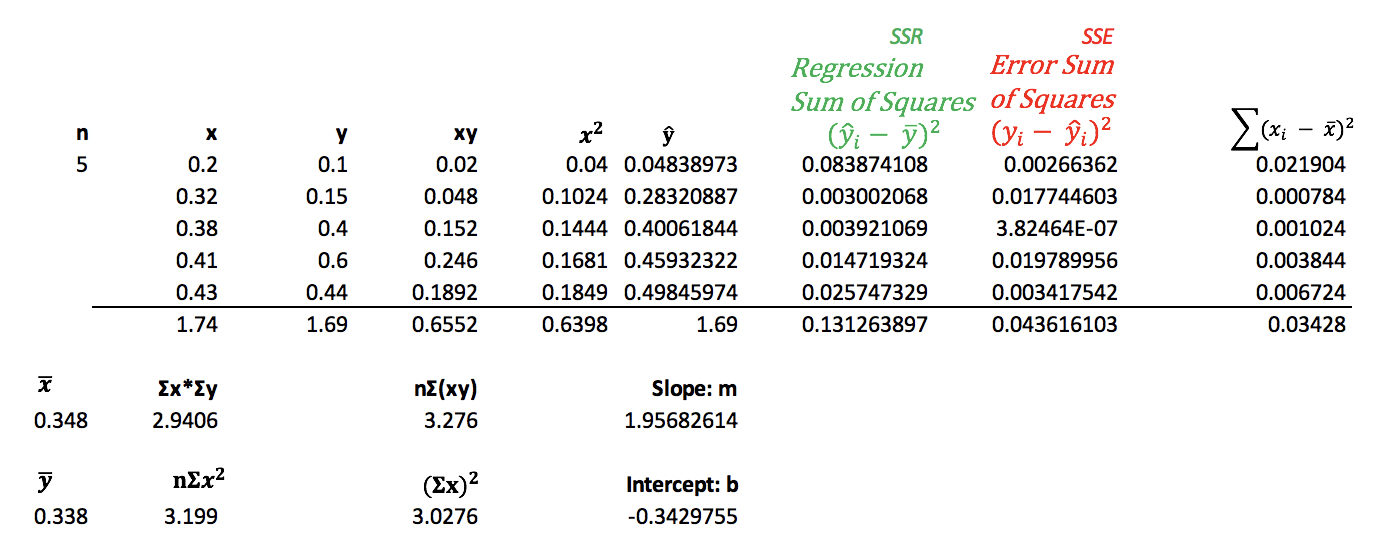
This example demonstrates how to validate the slope coefficient of our model with an region of *5*% for the F-Statistic.

Null hypothesis = = slope, *m*, is equal to zero. **(Undesirable)**

Alternate hypothesis = = slope, *m*, is not equal to zero. **(Desirable)**

To calculate the F-statistic for Example 1, we need calculations for SSR and SSE (see Figure 8) as well as the degrees of freedom components that were calculated earlier.

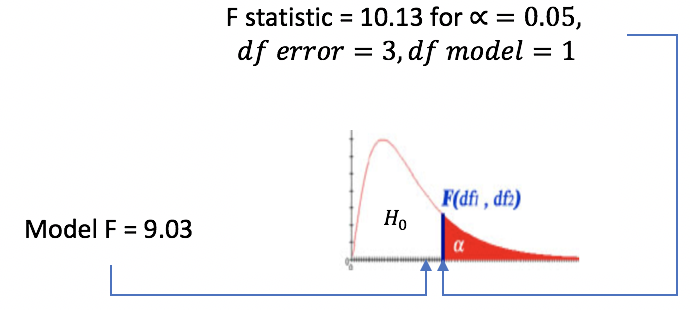
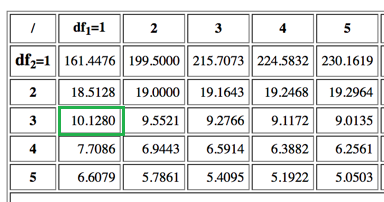
Figure : Manually Calculating SSR and SSE



We can then check the F statistic tables online for the confidence interval to determine if there is a chance that is accepted. When looking up the F statistic, the **error df** is used for the **row** of the F statistic table and the **model df** is used for the **column** of the F Statistic table.

The F Statistic table lookup for with the model df = 1 and error df = 3 suggests that F should be 10.128. Since our calculated value of F is 9.0286 we cannot reject the null hypothesis. It appears that there is more than a 5*%* chance that our model will not enable better predictions than the average.

Figure : F Statistic vs. Model F



**Conclusion:**

We cannot reject the null hypothesis when = 0.05. Therefore, we cannot prove that *m* is not zero. In other words, we have failed to confidently establish a relationship between the x and y within this range.

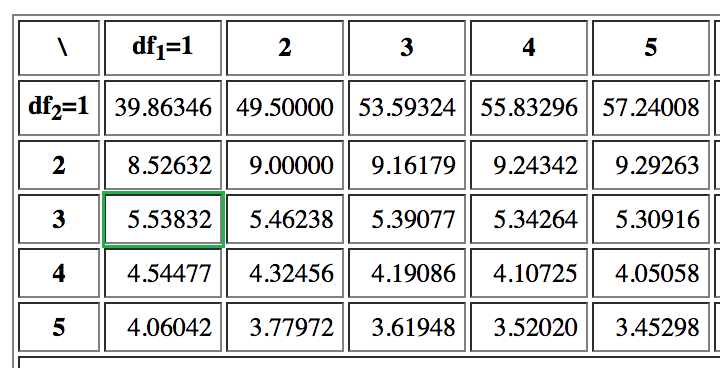
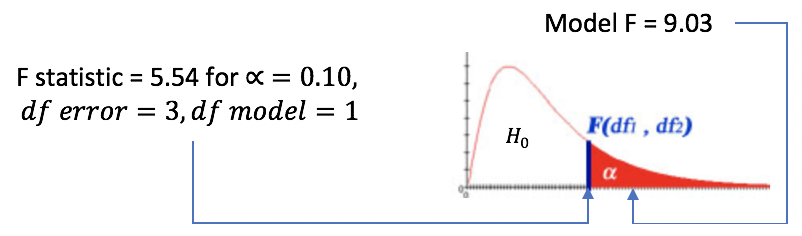
Example : Validating the Slope Coefficient When = 0.10

Continuing on from Example 2, we will test the slope coefficient where = 10%.

Null hypothesis = = slope, *m*, is equal to zero. **(Undesirable)**

Alternate hypothesis = = slope, *m*, is not equal to zero. **(Desirable)**

Figure : F Statistic vs. Model F

(Not drawn to scale)

**Conclusion:**

In F Statistic table lookup for when the model df = 1 and error df = 3, the F statistic is 5.53832. Our model generates a value for F of 9.0286 so the model’s F statistic is well into the region. We can therefore reject the null hypothesis since our F value is **statistically significant**. For a one-tailed confidence interval of 10% it appears that the slope coefficients offer a better prediction than the average target variable.

Exercise (5 marks)

Using your Excel spreadsheet from Exercise 1, calculate the regression sum of squares value. The output should be presented in a manner that is similar to the output that is presented in Figure 8. Show a screenshot of your regression sum of squares value here:

|  |
| --- |
|  |

Exercise (5 marks)

Using your Excel spreadsheet from Exercise 1, calculate the error sum of squares value. The output should be presented in a manner that is similar to the output that is presented in Figure 8. Show a screenshot of your error sum of squares value here:

|  |
| --- |
|  |

Exercise (1 mark)

What is the model degrees of freedom value for Exercise 1?

|  |
| --- |
|  |

Exercise (1 mark)

What is the error degrees of freedom value for Exercise 1?

|  |
| --- |
|  |

Exercise (3 marks)

Show the calculations for the F-statistic for Exercise 1.

|  |
| --- |
|  |

Exercise (2 marks)

Write statements for the null hypothesis and alternate hypothesis for the slope coefficient of your model here:

|  |
| --- |
|  |

Exercise (1 mark)

What is the F-value given the error and model degrees of freedom for Exercise 1 where = 0.05?

|  |
| --- |
|  |

Exercise (1 mark)

Based on the outcome in Exercise 8 do you conclude by rejecting or failing to reject the null hypothesis? Please state your conclusion clearly.

|  |
| --- |
|  |

Exercise (2 marks)

What is the F-value given the error and model degrees of freedom for Exercise 1 where = 0.1? What does this result imply?

|  |
| --- |
|  |

Exercise (2 marks)

Based on the outcome in Exercise 10Exercise 8 do you conclude by rejecting or failing to reject the null hypothesis? Please state your conclusion clearly.

|  |
| --- |
|  |

## Coefficient of Determination

The coefficient of determination, , is the proportion of variability accounted for by the model. Ideally, we want this number as high as possible (close to 1). A value of 1 suggests perfect positive or negative correlation. However, we need to be careful about overfitting the model to the sample (more on this soon).

Equation : Coefficient of Determination = =

For Example 1, =

Exercise (4 marks)

Manually show the calculations for the coefficient of determination for Exercise 1.

|  |
| --- |
|  |

Is the correlation relatively high or low? Explain.

|  |
| --- |
|  |

### Adjusted Coefficient of Determination

Too many predictor variables can lead to a model that is overfit (heavily biased) based on the sample. An overfitted model may perform well with the current data but it will on average perform more poorly with a different data set.

The adjusted coefficient of determination, , is often used to describe the variability of a model instead of for because is lowered when more prediction variables are included in the model. In other words, helps to reduce bias by adding a penalty for an increased number of prediction variables.

To calculate the adjusted coefficient of determination we use the following variables:

= The coefficient of determination.

p = Number of predictors.

N = Sample size.

Equation : = 1 -

For Example 1,

= 1 - = 1 - = 0.6674

Exercise (2 marks)

Manually show the calculations for based on your result for Exercise 12.

|  |
| --- |
|  |

## Train and Test Data

Predictive models are developed with existing data. The model development step is called ***training***. Part of the validation process though involves checking the model’s predictions with known results. To help avoid adding model bias from the current data set (***overfitting***), a portion of existing data is kept out of the training data set so the model’s performance can be evaluated with it later. This reserve of data for model validation is called ***test data***. Several techniques exist for allocating test and training data. We will use a simple split technique to get started but we will examine other splitting algorithms later.

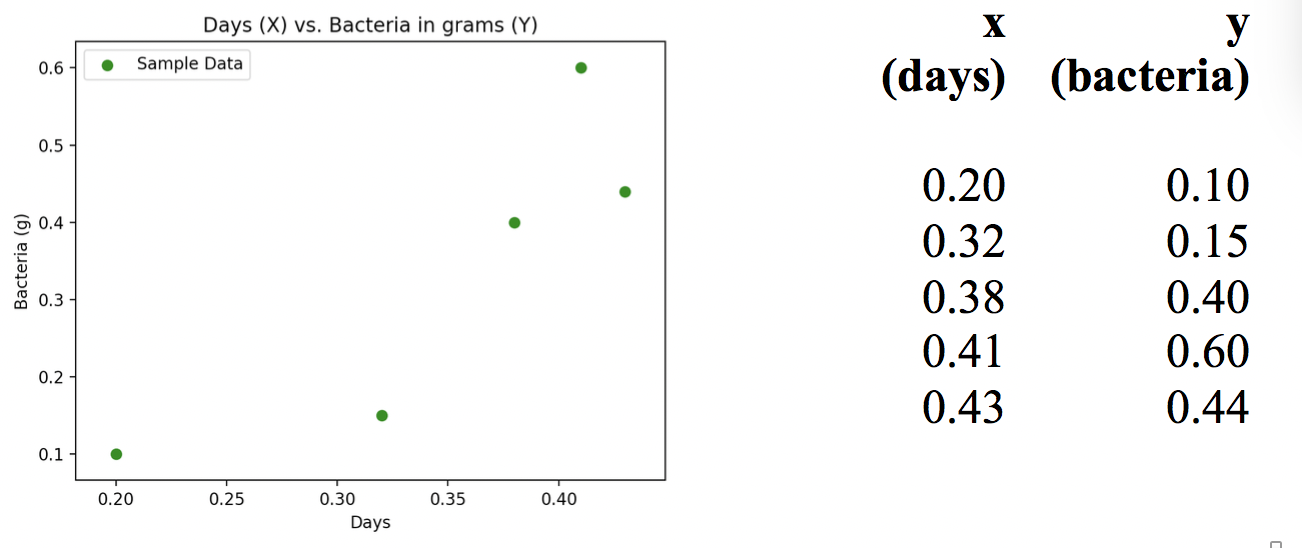
### Target and Predictor Variables

Predictive models are developed to predict outcomes. The variable being predicted is called a ***target*** variable. The target variable is represented with the name ***y***. Variables that are used to estimate the target variable are called ***predictor*** variables and they are often represented with the name ***x***.

Example : Generating Test and Training Data

This example will demonstrate how to randomly split the data set into 60% training data and 40% test data. Say we had the following data set.

Figure : Scatter Plot of Five Separate Bacteria Samples



To randomly split the data into 60% training data and 40% test data we can use the following code. The existing data is stored in the variables that are highlighted in yellow. The variables that are generated and populated with data after the *train\_test\_split()* function runs are highlighted in green.

|  |
| --- |
| import pandas as pd  from sklearn.model\_selection import train\_test\_split  # Create DataFrame.  dataSet = {'days': [0.2, 0.32, 0.38, 0.41, 0.43],  'growth': [0.1, 0.15, 0.4, 0.6, 0.44] }  df = pd.DataFrame(dataSet, columns= ['days', 'growth'])  # Store x and y values.  X = df['days']  target = df['growth']  # Create training set with 60% of data and test set with 40% of data.  X\_train, X\_test, y\_train, y\_test = train\_test\_split(  X, target, train\_size = 0.6  ) |

The output after the split shows how the data from the original sample set is divided into test and training data sets.

|  |  |  |  |
| --- | --- | --- | --- |
| **X\_train** | **y\_train** | **X\_test** | **y\_test** |
| ﻿﻿0.38  0.43  0.32 | ﻿ 0.40  0.44  0.15 | ﻿﻿ 0.20  0.41 | ﻿ 0.1  0.6 |

Exercise (2 marks)

Run the code in again but this time use a 80% train / 20% test split. Print out the values for X\_train, X\_test, y\_train and y\_test. Show a screenshot of your output here:

|  |
| --- |
|  |

How does your output in this exercise differ when compared to the output from ?

|  |
| --- |
|  |

### Preventing Randomization During Test-Train Split

Sometimes when documenting a model after it is developed you will want to ensure the test and train data is consistent during multiple runs. You can prevent randomization with by setting the **random\_state parameter to zero** in the train\_test\_split() function:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,

random\_state=0)

Example : Simple Linear Regression

Now that we have discussed training and test data in , this current example shows how the train and test can be used to create a model and to validate it. For this case, a simple best fit line using least squares regression will be generated to fit through the scattering of samples that are shown in .

##### Calculating the Root Mean Square Error (RMSE) – Average Deviation from the Actual Value

While also demonstrating test and train data, we are going to also calculate the root mean square error RMSE which describes the average deviation between the actual and predicted target value. Root MSE is of residuals.

Equation : Root Mean Square Error

The formula for RMSE is:

=

Do not be discouraged by the math. We really are just focusing on how the train and test data are used. Also, the RMSE measure of variance to help understand the effectiveness of our model. A lower RMSE is preferred because it means our prediction is closer to the actual result.

Example : Automating Regression with Test and Train Data

This code sample introduces automated regression with Python. This example is quite incomplete but we will expand on this next day.

|  |
| --- |
| import pandas as pd  from sklearn.model\_selection import train\_test\_split  from statsmodels.formula.api import ols  from sklearn import metrics  import math  def performSimpleRegression():  # Initialize collection of X & Y pairs like those used in example 5.  data = [[0.2,0.1],[0.32,0.15],[0.38,0.4],[0.41,0.6],[0.43,0.44]]    # Create data frame.  dfSample = pd.DataFrame(data, columns = ['X', 'target'])    # Create training set with 60% of data and test set with 40% of data.  X\_train, X\_test, y\_train, y\_test = train\_test\_split(  dfSample['X'], dfSample['target'], train\_size = 0.6  )    # Create DataFrame with test data.  dataTrain = {"X":X\_train, "target":y\_train}  dfTrain = pd.DataFrame(dataTrain, columns = ['X', 'target'])    # Generate model to predict target using X.  model = ols('target ~ X', data=dfTrain).fit()  y\_prediction = model.predict(X\_test)    # Present X\_test, y\_test, y\_predict and error sum of squares.  data = {"X\_test":X\_test, "y\_test":y\_test, "y\_prediction":y\_prediction}  dfResult = pd.DataFrame(data, columns = ['X\_test', 'y\_test', 'y\_prediction'])  dfResult['y\_test - y\_pred'] =(dfResult['y\_test']-dfResult['y\_prediction'])  dfResult['(y\_test - y\_pred)^2']=(dfResult['y\_test']-dfResult['y\_prediction'])\*\*2  # Present X\_test, y\_test, y\_predict and error sum of squares.  print(dfResult)  # Manually calculate the deviation between actual and predicted values.  rmse = math.sqrt(dfResult['(y\_test - y\_pred)^2'].sum()/len(dfResult))  print("RMSE is average deviation between actual and predicted values: "  + str(rmse))  # Show faster way to calculate deviation between actual and predicted values.  rmse2 = math.sqrt(metrics.mean\_squared\_error(y\_test, y\_prediction))  print("The automated root mean square error calculation is: " + str(rmse2))  performSimpleRegression() |

The output shows known values from our test set beside the predicted y value. Results from the root mean square error calculation are also shown. A lower deviation between the actual and predicted value is desirable.

When we inspect our output, we can see the first predicted y value is close to the actual y\_test value. However, the second predicted y value is relatively far away from the actual y\_test value. This deviation may be a problem. We can look at the RMSE to understand the overall average deviation from the actual and predicted values.

|  |
| --- |
| ﻿ X\_test y\_test y\_prediction y\_test - y\_pred (y\_test - y\_pred)^2  2 0.38 0.40 0.441602 -0.041602 0.001731  1 0.32 0.15 0.330719 -0.180719 0.032659  RMSE is the deviation between actual and predicted values: 0.1311295925504966  The automated root mean square error calculation is: 0.1311295925504966 |

Exercise (2 marks)

Run the linear regression again but adjust the code so it uses an 80% train - 20% test split. Show the output here:

|  |
| --- |
|  |

Does the RMSE go up or down? Explain why you think it changed the way that it did.

|  |
| --- |
|  |

Exercise (2 marks)

Calculate the model degrees of freedom and the error (residuals) degrees of freedom for the sample in Example 6. The model generated includes an intercept. Show your results (with calculations here)

|  |
| --- |
|  |