|  |  |  |  |
| --- | --- | --- | --- |
| Date: | **30-07-2020** | Name: | **Varun G Shetty** |
| Course: | **Coursera** | USN: | **4AL17EC093** |
| Topic: | Basic Statestics | Semester & Section: | **6th & ‘B’** |
| GitHub Repository: | **Varunshetty4** |  |  |

**Report-**

R Language:

Z-Score: We can calculate the z-score for a given value (X) as (X - mean) / standard deviation. In R you can do this with a whole variable at once by putting the variable name in the place of X.

Scatterplot: Let's have a look at the relationship between height and weight through a scatterplot, using the R function plot(). The first argument of plot() is the x-axis coordinates, and the second argument is the y-axis coordinates. Use main = \*title here\* inside plot() to add the title "Heights and Weights"

Contingency Table: We can make a contingency table of this data using the table() function. While previously you may have used this with one variable, this time you will use it with two. The first variable used with table() will appear in the rows, while the second variable will appear in the columns.

Round(): if you want a quick way to do this through R you can use the round() function. The first argument of round() is the value that you want to round (this can be in the form of a raw number, or an equation), and the second argument is digits =, where you specify the number of decimal places you want the number rounded to. For instance,

 round(12.6734, digits = 2) would return the value 12.67.

Correlation: We can calculate the correlation in R using the function cor(), which takes your two variables as it's first argument.

Regression Line: When we draw a line through our data, we measure error as the sum of the difference between the observation and the line. We usually square this so that positive and negative residuals don't cancel each other out. The line that gives us the least error is our regression line.

To do this you should use the sum() function, which returns the sum of all vectors provided between brackets. You can also put ^2 inside the brackets with your vectors in order to square the differences. For example, sum((vector1 - vector2) ^ 2).

Regression Expression: The regression equation is Y = a + bx, where a is the intercept and b is the slope of the line.

Regression Coefficient: We can find the regression coefficients for our data using the lm() function, which takes our model as the first argument: first the y variable, followed by a ~ symbol, then the x variable.

After you have created your scatterplot, you can add a line using the function abline(). abline() takes the intercept of the line as its first argument, and the slope of the line as its second argument. This makes it a pretty good candidate for storing your lm() output as an object, and putting it straight into abline

 One line shows the mean, and one shows the regression line. Clearly, there is less error when we use the regression line compared to the mean line. This reduction in error from using the regression line compared to the mean line tells us how well the independent variable (money) predicts the dependent variable (prosocial behaviour). Conveniently, the R squared is equivalent to squaring the Pearson R correlation coefficient.