Translating Math Into R Worksheet

STATS20 Introduction to Statistical Programming

Selina Cheng & Thomas Maierhofer

2024-12-04

1 Introduction

Mathematical notation can be intimidating, but fear less! If you're struggling with understanding how to decode mathematical functions, this ungraded worksheet will break down various functions and give additional practice for translating it into R code.

2 Exercises

The solutions for these ungraded exercises can be found at the end of this worksheet.

Given the following data:

$$x = \{8, 3, 4, 1, 9\}$$

$$y = \{2, 7, 5, 8, 3\}$$

Compute the numeric values of the following statistics by plugging in the given values of x and y into each formula, first by hand and then with R code.

2.1 Sample Mean

The sample mean is the average value of a sample of numbers from a larger population, given by the following formula:

$$\bar{x} = \frac{1}{n} \sum x_i$$

where n represents the number of observations in the vector x, x_i represents $x_1, x_2, \ldots, x_n$, which are the first observation of x, the second observation of x, up until x_n which is the last observation of x.

Compute $\bar{x}$, the sample mean of x, and $\bar{y}$, the sample mean of y:

Write your answer here:

2.2 Harmonic Mean

The harmonic mean is a type of average most appropriately used for ratios and rates, given by the following formula:

$$H = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

The numerator n represents the number of observations in x.

In the denominator, $\sum_{i=1}^{n} \frac{1}{x_i}$ represents the sum of each observations's reciprocal.

$\sum_{i=1}^{n}$ is sum notation, starting from the first observation to the last observation of x.

$\frac{1}{x_i}$ is the reciprocal of each observation of x.

Compute H_x, the harmonic mean of x, and H_y the harmonic mean of y:

Write your answer here:

2.3 Sample Standard Deviation

The sample standard deviation measures the amount of variability of the values of a sample about its sample mean. To compute this you will have to compute the sample mean (see Section 2.1) first.

$$s_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$x_i - \bar{x}$ is the difference between the observed values and the sample mean, $\bar{x}$.

$\sum_{i=1}^{n} (x_i - \bar{x})^2$ is the sum of the aforementioned differences squared.

$\frac{1}{n-1}$ is the reciprocal of the sample size minus 1.

Compute s_x, the sample standard deviation of x, and s_y, the sample standard deviation of y:

Write your answer here:

2.4 Z-Score

The z-score, also known as the standard score, is the number of standard deviations by which the value of a raw score is above or below the mean value of the sample.

$$z_{x_i} = \frac{x_i - \bar{x}}{s_x}$$

x_i represents the ith entry of x.

$\bar{x}$ represents the sample mean of x, using the aforementioned formula (Section 2.1) in this worksheet.

s_x represents the sample standard deviation, also using the aforementioned formula (Section 2.3 in this worksheet.

Compute z_{x_1}, the z-score of the first element of x, and z_{y_1}, the z-score of the first element of y:

Write your answer here:

2.5 Correlation Coefficient

The correlation coefficient is a measure of linear correlation, implying a statistical relationship between two variables.

This following formula is Pearson's correlation coefficient, written as the mean of the products of the z-scores.

$$r_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} z_{x_i} \cdot z_{y_i}$$

z_{x_i} represents the z-score of the ith entry of x.

Similarly, z_{y_i} represents the z-score of the ith entry of y.

n is the sample size, implying that x and y should be of the same length.

Essentially, multiply the z-score of each entry of each data set with one another (e.g. multiply z-score of 1st entry of x with the z-score of 1st entry of y, then multiply z-score of 2nd entry...) and add all of the products together.

Alternatively, it can also be written as:

$$r_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s_x} \right) \cdot \left(\frac{y_i - \bar{y}}{s_y} \right)$$

This version of the formula replaces z_{x_i} and z_{y_i} with the formula for standardizing values, as mentioned in the Z-Score section.

Compute the correlation coefficient of x and y:

Write your answer here:

2.6 Correlation Coefficient (2)

The following formula is an alternative way to calculate the Pearson correlation coefficient for a sample:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

This formula is similar to the previous form of the correlation coefficient without directly utilizing the formula for z-score.

Compute the correlation coefficient of x and y:

Write your answer here:

2.7 Regression Parameters

Regression parameters are the variables that describe the relationship between a predictor variable (x) and a response variable (y) in a regression model formatted as $y = \beta_1 + \beta_0 x$.

$$\beta_1 = r_{xy} \left(\frac{s_y}{s_x} \right)$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

r_{xy} represents the correlation coefficient between the data sets x and y.

$\frac{s_y}{s_x}$ represents the sample standard deviation of y divided by the sample standard deviation of x.

$\bar{y}$ and $\bar{x}$ represent their respective sample means.

Compute β_1 and β_0 for the regression model $y = \beta_1 + \beta_0 x$:

Write your answer here:

2.8 Gini Coefficient

The Gini coefficient measures the inequality among the values of a frequency distribution, commonly used for levels of income. A Gini coefficient of 0 means all values of the distribution are the same, whereas a Gini coefficient of 1 reflects maximum inequality across the distribution, where one value's frequency is the frequency of the whole distribution.

$$G(x) = \frac{2\sum_{i=1}^{n} i \cdot x_i}{n\sum_{i=1}^{n} x_i} - \frac{n+1}{n}$$

x_i represents an observed value of x.

n represents the number of values in x.

For this problem only, use the following data:

$$x = \{1000, 200, 500, 5000, 10000\}$$

$$y = \{70000, 1110, 3500, 11400, 2000\}$$

Compute $G(x)$ for x and $G(y)$ for y:

Write your answer here:

2.9 Cronbach's Alpha

Cronbach's alpha is a reliability coefficient used to measure the internal consistency of tests (ideally with normally distributed, linear data), i.e., it measures whether a series of questions measure "the same thing". You do not need to know what Cronbach's Alpha is, what it does, and how it works. Just plug the numbers into the formula.

Given a collection of vectors z, containing multiple vectors $x, y, ...,$

$$\alpha(x) = \frac{k}{k-1}\left(1 - \frac{\sum_{i=x,y,\ldots}\sigma^2_{z_i}}{\sigma^2_z}\right)$$

k represents the total number of vectors (e.g. tests) in z.

z_i for $i = x, y, \ldots$ represents each individual vector in the overall collection z.

$\sigma^2_{z_i}$ represents the variance associated with each individual vector in the collection.

σ^2_z represents the variance associated with the entire collection.

For the numerator $\sum_{i=x,y,\ldots}\sigma^2_{x_i}$, compute the value by adding the variance of vector x and the variance of vector y.

For the denominator σ^2_z, compute the value by first adding vector x and vector y, resulting in vector z, then taking the variance of z (the total).

Compute α for the collection of vectors $z = \{x, y\}$:

Write your answer here:

2.10 Coefficient of Skewness

The coefficient of skewness measures how much the distribution of a sample differs from symmetry. A perfectly symmetric distribution will have a skewness of 0. If the skewness coefficient is significantly greater than 0, then the distribution is right-skewed. If the skewness coefficient is significantly less than 0, then the distribution is left-skewed.

$$\text{Skew}(x) = \frac{\sqrt{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{[\sum_{i=1}^{n}(x_i - \bar{x})^2]^{3/2}}$$

where $\bar{x} = \sum_{i=1}^{n} x_i$ is the sample mean.

A one-pass formula for the coefficient of skewness is

$$\text{Skew}(x) = \frac{\sqrt{n}\left(\sum_{i=1}^{n} x_i^3 - 3\bar{x}\sum_{i=1}^{n} x_i^2 + 2n\bar{x}^3\right)}{\left(\sum_{i=1}^{n} x_i^2 - n\bar{x}^2\right)^{3/2}}$$

Be sure to break apart each component and follow the order of operations. Use parentheses to help organize your work.

Compute the skew for x using both formulas:

Both formulas should result in the same value of skew.

Write your answer here:

3 Exercises In R

3.1 Compute using R

Now that you've practiced computing these statistics by hand, you have a greater understanding of how to break down the components of a formula into smaller, manageable parts. Use those skills to recompute each statistical measure using R with the previously given data sets of x and y. You can use your R code to check what you computed by hand and vice-versa.

4 Functionalize your Code

Once you have R code that computes a statistic for the data sets x and y, wrap it up nicely in a function. Writing a function should be the last thing you do, make sure you test your function on the data sets x and y.

5 Check your Functions on New Data

Use your R functions to compute each statistic for the following new data sets.

```r
set.seed(42)
a <- sample(1:100, size = 500, replace = TRUE)
b <- a + sample(1:500, size = 500, replace = TRUE)
```

1. Sample mean
2. Harmonic mean
3. Sample standard deviation
4. Z-score
5. Correlation coefficient
6. Correlation coefficient (alternative formula)
7. Regression parameters
8. Gini's coefficient
9. Cronbach's alpha
10. Coefficient of skewness

6 Solutions for Exercises

6.1 Sample Mean

$$\bar{x} = \frac{(8+3+4+1+9)}{5} = 5$$

$$\bar{y} = \frac{(2+7+5+8+3)}{5} = 5$$

6.2 Harmonic Mean

$$H_x = \frac{5}{\frac{1}{8}+\frac{1}{3}+\frac{1}{4}+\frac{1}{1}+\frac{1}{9}} = 2.748$$

$$H_y = \frac{5}{\frac{1}{2}+\frac{1}{7}+\frac{1}{5}+\frac{1}{8}+\frac{1}{3}} = 3.843$$

6.3 Sample Standard Deviation

$$s_x = \sqrt{\frac{1}{5-1}\left[(8-5)^2+(3-5)^2+(4-5)^2+(1-5)^2+(9-5)^2\right]} = 3.391$$

$$s_y = \sqrt{\frac{1}{5-1}\left[(2-5)^2+(7-5)^2+(5-5)^2+(8-5)^2+(3-5)^2\right]} = 2.550$$

6.4 Z-Score

$$z_{x_1} = \frac{8-5}{3.391} = 0.885$$

$$z_{y_1} = \frac{2-5}{2.550} = -1.177$$

6.5 Correlation Coefficient

$$r_{xy} = \frac{1}{5-1}\left[\left(\frac{8-5}{3.391}\cdot\frac{2-5}{2.550}\right)+\left(\frac{3-5}{3.391}\cdot\frac{7-5}{2.550}\right)+\left(\frac{4-5}{3.391}\cdot\frac{5-5}{2.550}\right)+\left(\frac{1-5}{3.391}\cdot\frac{8-5}{2.550}\right)+\left(\frac{9-5}{3.391}\cdot\frac{3-5}{2.550}\right)\right] = -0.954$$

6.6 Correlation Coefficient (2)

$$r_{xy} = \frac{[(8-5)\cdot(2-5)]+[(3-5)\cdot(7-5)]+[(4-5)\cdot(5-5)]+[(1-5)\cdot(8-5)]+[(9-5)\cdot(3-5)]}{\sqrt{(8-3)^2+(3-5)^2+(4-5)^2+(1-5)^2+(9-5)^2}\cdot\sqrt{(2-5)^2+(7-5)^2+(5-5)^2+(8-5)^2+(3-5)^2}} = -0.954$$

6.7 Regression Parameters

$$\beta_0 = -0.954\left(\frac{2.550}{3.391}\right) = -0.717$$

$$\beta_1 = 5 - \left[\left(-0.954\cdot\frac{2.550}{3.391}\right)\cdot 5\right] = 8.587$$

6.8 Gini's Coefficient

$$G_x = \frac{2[(1 \cdot 1000) + (2 \cdot 200) + (3 \cdot 500) + (4 \cdot 5000) + (5 \cdot 10000)]}{5 \cdot (1000 + 200 + 500 + 5000 + 10000)} - \frac{5+1}{5} = 0.546$$

$$G_x = \frac{2[(1 \cdot 70000) + (2 \cdot 1110) + (3 \cdot 3500) + (4 \cdot 11400) + (5 \cdot 2000)]}{5 \cdot (70000 + 1110 + 3500 + 11400 + 2000)} - \frac{5+1}{5} = -0.571$$

6.9 Cronbach's Alpha

$$\frac{2}{2-1}\left(1 - \frac{11.5 + 6.5}{1.5}\right) = -22$$

6.10 Coefficient of Skewness

$$\text{Skew}_x = \frac{\sqrt{5}[(8-5)^3 + (3-5)^3 + (4-5)^3 + (1-5)^3 + (9-5)^3]}{[(8-5)^2 + (3-5)^2 + (4-5)^2 + (1-5)^2 + (9-5)^2]^{3/2}} = 0.129$$

$$\text{Skew}_y = \frac{\sqrt{5}[(2-5)^3 + (7-5)^3 + (5-5)^3 + (8-5)^3 + (3-5)^3]}{[(2-5)^2 + (7-5)^2 + (5-5)^2 + (8-5)^2 + (3-5)^2]^{3/2}} = 0$$

One-Pass Versions:

$$\text{Skew}_x = \frac{\sqrt{5}\left[(8^3 + 3^3 + 4^3 + 1^3 + 9^3) - 3(5)[8^2 + 3^2 + 4^2 + 1^2 + 9^2] + 2(5)(5^3)\right]}{[(8^2 + 3^2 + 4^2 + 1^2 + 9^2) - 5(5^2)]^{3/2}} = 0.129$$

$$\text{Skew}_y = \frac{\sqrt{5}\left[(2^3 + 7^3 + 5^3 + 8^3 + 3^3) - 3(5)[2^2 + 7^2 + 5^2 + 8^2 + 3^2] + 2(5)(5^3)\right]}{[(2^2 + 7^2 + 5^2 + 8^2 + 3^2) - 5(5^2)]^{3/2}} = 0$$

7 Solutions in R

```
x <- c(8, 3, 4, 1, 9)
y <- c(2, 7, 5, 8, 3)
```

7.1 Sample Mean

```
sum(x)/length(x)
```

```
## [1] 5
```

```
sum(y)/length(y)
```

```
## [1] 5
```

```
# built-in function: mean(x)
```

7.1.1 Function

```
sample_mean <- function(x) {
  sum(x) / length(x)
}
sample_mean(x)
```

```
## [1] 5
```

```r
sample_mean(y)
```

```
## [1] 5
```

```r
sample_mean(a)
```

```
## [1] 50.182
```

```r
sample_mean(b)
```

```
## [1] 299.326
```

7.2 Harmonic Mean

```r
length(x) / sum(1/x)
```

```
## [1] 2.748092
```

```r
length(y) / sum(1/y)
```

```
## [1] 3.842635
```

7.2.1 Function

```r
harmonic_mean <- function(x) {
  length(x) / sum(1/x)
}
harmonic_mean(x)
```

```
## [1] 2.748092
```

```r
harmonic_mean(y)
```

```
## [1] 3.842635
```

```r
harmonic_mean(a)
```

```
## [1] 19.08902
```

```r
harmonic_mean(b)
```

```
## [1] 193.7974
```

7.3 Sample Standard Deviation

```r
sqrt((1/(length(x) - 1)) * sum((x - mean(x))^2))
```

```
## [1] 3.391165
```

```r
sqrt((1/(length(y) - 1)) * sum((y - mean(y))^2))
```

```
## [1] 2.54951
```

```r
# built-in function: sd(x)
```

7.3.1 Function

```r
sample_stdev <- function(x) {
  sqrt((1/(length(x) - 1)) * sum((x - mean(x))^2))
```

```r
}
sample_stdev(x)
```

```
## [1] 3.391165
```

```r
sample_stdev(y)
```

```
## [1] 2.54951
```

```r
sample_stdev(a)
```

```
## [1] 29.48334
```

```r
sample_stdev(b)
```

```
## [1] 148.9416
```

7.4 Z-Score

```r
# z score for individual observations
(x[1] - mean(x))/sd(x)
```

```
## [1] 0.8846517
```

```r
(y[1] - mean(y))/sd(y)
```

```
## [1] -1.176697
```

```r
# alternatively, you can return all standardized observations
(x - mean(x))/sd(x)
```

```
## [1]  0.8846517 -0.5897678 -0.2948839 -1.1795356  1.1795356
```

```r
(y - mean(y))/sd(y)
```

```
## [1] -1.1766968  0.7844645  0.0000000  1.1766968 -0.7844645
```

7.4.1 Function

```r
# Calculate the z-score of a single observation in a vector
# Let x be the vector, i be the index of the observation being standardized

z_score <- function(x, i) {
  (x[i] - mean(x))/sd(x)
}
z_score(x, 1)
```

```
## [1] 0.8846517
```

```r
z_score(y, 1)
```

```
## [1] -1.176697
```

```r
z_score(a, 1)
```

```
## [1] -0.04009044
```

```r
z_score(b, 1)
```

```
## [1] -0.2506083
```

```r
# alternatively, you can return all standardized observations
z_score <- function(x) {
  (x - mean(x))/sd(x)
}
z_score(x)
```

```
## [1]  0.8846517 -0.5897678 -0.2948839 -1.1795356  1.1795356
```

```r
z_score(y)
```

```
## [1] -1.1766968  0.7844645  0.0000000  1.1766968 -0.7844645
```

```r
z_score(a)
```

```
##   [1] -0.040090436  0.502588899 -0.854109437  0.807846024  1.689699942
##   [6] -1.091531646 -0.040090436 -0.107925353 -0.888026896  0.706093649
##  [11]  1.689699942  1.316607900 -0.447099937 -1.023696729 -0.820191979
##  [16] -1.600293522 -0.311430103  1.316607900 -0.786274521 -0.481017395
##  [21]  1.520112650 -1.532458605  1.147020608 -0.548852312  1.418360275
##  [26] -1.600293522  0.265166690  1.587947567 -0.277512645 -0.888026896
##  [31] -0.684522145 -0.243595186 -1.193284021 -0.955861813  0.265166690
##  [36] -1.430706230 -0.481017395  0.604341274  1.214855525 -1.091531646
##  [41]  1.418360275  0.638258732 -1.566376064  1.621865026 -0.006172977
##  [46]  1.655782484  1.282690442  1.248772983 -0.040090436 -0.820191979
##  [51] -1.498541147 -1.498541147 -1.634210980 -1.600293522 -0.989779271
##  [56] -1.634210980  0.265166690 -1.362871313 -0.345347561 -1.532458605
##  [61] -0.582769770 -0.040090436  1.689699942  0.773928566 -0.718439604
##  [66]  0.875680941  1.147020608 -1.396788772 -0.514934853  1.452277734
##  [71] -1.159366563  1.418360275  0.638258732  1.418360275 -1.634210980
##  [76]  1.079185691 -0.888026896 -1.091531646  0.638258732  0.163414315
##  [81] -0.345347561 -0.989779271  1.689699942  0.231249231  1.689699942
##  [86] -0.277512645 -1.091531646  1.384442817 -1.261118938  0.095579398
##  [91]  0.129496856  1.113103150 -0.616687229  1.011350774  0.333001607
##  [96] -0.718439604  1.045268233  0.773928566  1.180938066 -0.243595186
## [101]  0.265166690  0.740011107 -0.718439604  0.163414315 -0.413182478
## [106] -1.668128439 -1.261118938  0.943515858 -1.532458605  0.773928566
## [111]  0.163414315 -1.159366563  1.350525358 -0.243595186 -0.277512645
## [116] -0.277512645  1.587947567  0.231249231 -0.718439604 -0.854109437
## [121]  0.434753982 -0.616687229  1.045268233 -1.227201480 -1.498541147
## [126] -0.107925353 -0.243595186  0.400836523 -0.447099937  1.011350774
## [131] -0.650604687 -0.548852312  1.486195192  1.554030109 -1.498541147
## [136]  1.214855525 -0.413182478  1.520112650 -1.498541147  1.147020608
## [141] -1.193284021 -0.548852312 -0.277512645  1.248772983  0.333001607
## [146] -1.295036397 -0.820191979 -0.311430103  0.502588899  0.536506357
## [151]  0.197331773 -0.888026896  1.621865026 -0.854109437  1.418360275
## [156]  0.366919065  0.400836523 -1.227201480 -0.548852312  0.536506357
## [161]  1.520112650 -0.616687229 -0.786274521 -1.362871313  0.231249231
## [166] -0.752357062 -0.447099937 -1.362871313 -1.532458605 -0.514934853
## [171]  0.943515858 -1.227201480 -0.752357062  1.587947567  1.655782484
## [176]  0.129496856  1.350525358 -0.650604687 -0.243595186  0.061661939
## [181]  1.554030109  1.045268233  0.299084148 -0.786274521 -0.684522145
## [186]  0.061661939  1.689699942  1.147020608  0.841763482  1.045268233
## [191]  0.773928566 -1.125449105  0.400836523 -1.261118938  1.520112650
## [196]  0.434753982 -0.040090436  0.366919065 -1.668128439 -1.125449105
## [201] -0.582769770 -0.752357062 -1.634210980 -0.650604687 -1.430706230
```

```
## [206]  1.011350774 -1.600293522 -1.261118938 -1.295036397  0.502588899
## [211] -0.684522145  0.027744481  0.333001607  1.520112650 -0.447099937
## [216] -0.107925353  0.197331773  0.672176191 -1.159366563 -1.362871313
## [221]  0.706093649  0.773928566 -0.854109437 -1.600293522  1.079185691
## [226]  0.095579398 -0.752357062  0.977433316  1.452277734  0.231249231
## [231]  1.147020608  1.316607900  1.011350774  1.282690442  1.587947567
## [236]  0.366919065 -0.243595186 -0.955861813 -0.820191979  0.129496856
## [241]  0.129496856  1.147020608  0.163414315  0.129496856  0.265166690
## [246] -0.175760269 -0.447099937 -0.243595186  0.977433316  1.180938066
## [251] -0.548852312 -0.854109437  0.502588899 -1.227201480 -0.040090436
## [256] -1.328953855 -0.752357062 -1.091531646  0.163414315  1.621865026
## [261] -0.277512645 -0.481017395 -1.023696729  0.977433316  0.366919065
## [266]  0.706093649 -0.786274521  1.486195192 -1.532458605  1.214855525
## [271] -1.023696729 -0.311430103 -1.328953855  1.418360275 -0.413182478
## [276]  0.773928566 -0.921944354  1.350525358 -1.328953855 -1.057614188
## [281]  1.418360275  1.180938066 -0.481017395 -0.277512645  1.214855525
## [286]  1.316607900  0.129496856  0.299084148 -0.684522145  0.299084148
## [291]  1.180938066 -0.989779271  0.672176191 -0.107925353  0.807846024
## [296]  1.282690442  0.163414315 -1.464623689 -1.227201480 -1.295036397
## [301]  0.095579398  0.027744481 -1.362871313  1.316607900 -0.447099937
## [306]  0.536506357 -0.514934853 -0.548852312 -0.786274521 -0.141842811
## [311] -0.650604687  1.384442817  1.011350774  1.689699942  0.095579398
## [316] -0.888026896 -0.413182478  1.180938066 -1.023696729  0.841763482
## [321]  1.587947567  0.943515858  1.418360275 -0.582769770  1.452277734
## [326] -0.481017395 -0.175760269  1.554030109 -0.955861813 -0.209677728
## [331]  0.468671440 -1.464623689 -1.532458605 -0.650604687 -0.616687229
## [336]  1.079185691 -0.379265020 -1.261118938 -0.684522145  0.129496856
## [341] -1.634210980 -0.752357062 -1.600293522 -0.582769770  1.452277734
## [346]  1.147020608 -1.396788772  0.129496856 -1.362871313  0.706093649
## [351]  1.418360275  1.079185691 -0.854109437  0.807846024  0.773928566
## [356] -1.057614188  0.977433316 -0.141842811 -1.600293522 -0.955861813
## [361]  0.366919065 -1.023696729 -0.277512645  1.621865026  1.248772983
## [366] -1.125449105  1.079185691  1.113103150 -0.141842811 -1.125449105
## [371]  1.486195192  0.807846024 -0.447099937 -0.175760269 -0.006172977
## [376]  0.027744481  0.807846024 -0.074007894 -0.548852312  0.265166690
## [381]  0.604341274  1.316607900  1.350525358  0.366919065 -0.582769770
## [386]  1.214855525 -1.057614188  0.672176191  1.316607900 -0.684522145
## [391]  1.316607900 -1.634210980 -0.413182478 -1.396788772  0.197331773
## [396]  0.841763482 -1.498541147 -0.650604687  0.638258732 -1.227201480
## [401]  0.468671440 -0.107925353  0.502588899 -0.311430103 -1.362871313
## [406] -0.040090436  0.875680941 -0.955861813 -1.668128439 -1.566376064
## [411]  0.570423815  0.299084148  0.095579398 -0.413182478  0.299084148
## [416]  0.638258732 -0.277512645  0.400836523 -1.023696729 -0.548852312
## [421]  1.248772983  1.689699942  0.231249231 -1.023696729 -1.396788772
## [426] -0.786274521 -0.141842811 -0.141842811 -1.600293522  1.520112650
## [431]  0.434753982 -1.091531646 -1.362871313  1.587947567 -0.345347561
## [436]  1.147020608  0.095579398  0.672176191 -0.718439604  0.604341274
## [441]  1.384442817 -1.396788772 -0.345347561  1.248772983  1.350525358
## [446]  0.875680941  0.807846024  1.248772983  0.129496856  0.231249231
## [451] -0.786274521  1.621865026  1.011350774  1.316607900 -0.752357062
## [456] -1.193284021 -0.921944354  0.366919065  1.113103150 -1.362871313
## [461] -1.261118938  0.400836523 -1.668128439 -1.328953855 -0.820191979
## [466]  0.807846024  0.977433316  1.045268233 -1.430706230  1.689699942
## [471]  0.672176191  0.807846024  1.384442817  1.147020608 -0.854109437
```

```
## [476] -0.888026896  1.011350774 -1.159366563 -1.600293522  0.197331773
## [481]  1.554030109 -0.989779271  0.570423815 -1.362871313  1.180938066
## [486] -1.464623689 -1.057614188 -0.718439604 -0.175760269 -0.277512645
## [491] -0.141842811 -0.888026896  1.452277734 -0.684522145 -0.175760269
## [496]  1.655782484 -0.277512645  1.621865026 -0.379265020  0.061661939
```

```r
z_score(b)
```

```
##   [1] -0.250608277  0.501364262  1.025059424 -0.364746966  0.367083452
##   [6] -1.714269112 -1.029436978 -0.908584248  1.145912153 -0.633308587
##  [11]  0.769925884  0.461080019  1.098913869  0.978061140 -0.143183628
##  [16]  1.313763166  1.246622761 -0.250608277  1.300335085 -0.163325750
##  [21] -0.801159600 -1.754553355  1.206338518 -1.083149302 -0.230466155
##  [26]  0.682643357 -0.747447276  1.904598733  0.293229006 -0.129755547
##  [31] -0.962296573 -1.485991734 -0.861585965 -1.559846180 -0.653450708
##  [36]  1.394331653  1.347333369  0.038095466  1.448043977 -0.975724654
##  [41]  0.279800925  0.152234155  1.327191247 -0.666878789 -0.989152735
##  [46]  1.716605598  1.119055991 -0.787731519  1.145912153  1.327191247
##  [51] -0.250608277 -0.405031209 -1.929118409  0.555076587 -0.606452425
##  [56]  0.226088601  1.568896706  0.790068005 -0.660164749 -0.539312019
##  [61]  0.031381425 -1.224144153  0.783353965 -0.828015762  1.481614179
##  [66] -0.572882222 -0.102899385 -0.452029493  1.421187815  0.998203261
##  [71] -0.740733235 -0.828015762 -0.317748682 -0.586310303 -0.814587681
##  [76] -1.405423248  0.514792343  0.709499519 -0.841443843 -0.458743533
##  [81] -1.237572234 -0.922012330 -0.646736668 -1.506133855  1.072057707
##  [86]  0.561790627 -0.995866775 -1.190573951  0.172376277 -1.519561937
##  [91] -0.237180196 -0.760875357 -0.176753831 -0.163325750  0.004525263
##  [96] -0.042473020  0.380511533 -0.754161316  0.346941330 -0.331176763
## [101]  0.508078303  0.273086884 -0.720591114 -0.431887371 -1.002580816
## [106] -1.788123558 -0.653450708 -1.123433545  1.313763166  1.286907004
## [111]  1.636037112 -1.036151019  1.689749436 -1.673984869 -0.364746966
## [116] -1.653842747 -0.841443843  1.374189531 -1.136861626 -0.324462723
## [121] -0.626594546 -1.546418099  1.206338518  1.340619328 -0.250608277
## [126]  0.628931032 -0.532597979  1.172768315 -0.351318885  0.642359113
## [131] -1.009294856  0.051523547  1.535326504 -1.103291424  1.340619328
## [136]  1.374189531 -0.384889087  0.118663952 -0.875014046 -0.452029493
## [141]  1.273478923  1.394331653 -1.673984869  0.427509817 -0.559454141
## [146]  1.018345383  0.622216992 -1.083149302 -0.337890804 -1.257714356
## [151]  0.655787195 -0.183467871  1.991881260 -0.606452425 -0.734019195
## [156]  0.534934465 -1.130147586  0.944490937  0.017953344  0.172376277
## [161]  1.166054275 -1.673984869 -0.203609993 -0.747447276  1.266764883
## [166]  1.192910437  0.870636491 -1.170431829  1.300335085 -1.573274261
## [171] -0.492313736  0.145520114 -0.734019195  1.649465193  0.514792343
## [176] -0.438601412 -0.848157884 -0.962296573 -0.143183628  1.662893274
## [181] -0.814587681 -1.197287991  0.266372844  1.031773464 -0.754161316
## [186]  1.145912153  0.528220425  1.159340234 -0.062615142  1.226480639
## [191]  0.064951628  0.622216992 -0.599738384 -0.720591114  0.608788911
## [196] -1.526275977  0.461080019 -0.324462723 -1.875406085 -1.754553355
## [201]  1.508470342  0.615502951  1.004917302  1.501756301  1.166054275
## [206]  0.904206694  0.508078303  0.702785478 -0.740733235  0.602074870
## [211]  1.072057707 -0.895156167  0.548362546 -0.754161316  0.501364262
## [216]  1.125770031 -0.391603128  1.266764883 -0.861585965  1.260050842
## [221] -0.499027776 -1.136861626 -0.411745249 -0.431887371  0.494650222
## [226] -0.405031209 -1.734411233 -0.821301722  0.145520114  0.367083452
## [231]  1.421187815  0.924348816  1.172768315  0.944490937 -0.619880506
```

```
## [236] -1.418851329  0.649073154 -0.143183628 -1.398709207 -1.123433545
## [241]  0.924348816  0.192518398 -1.251000315 -0.002188777  1.602466909
## [246] -1.519561937  1.535326504  0.212660520  0.944490937 -0.378175047
## [251]  0.709499519 -0.734019195  0.071665669  0.461080019  1.602466909
## [256]  1.307049126 -0.801159600 -0.734019195 -1.412137288  1.817316206
## [261] -0.888442127  0.138806074  0.165662236 -1.358424964  1.179482356
## [266] -0.525883938 -0.794445560 -1.049579100 -0.552740101  0.924348816
## [271]  0.541648506 -1.532990018 -1.452421531  1.528612463  1.139198113
## [276]  1.656179233 -1.465849612 -0.290892520 -1.271142437  0.273086884
## [281]  1.387617612 -0.264036358 -1.741125274  1.038487505 -1.157003748
## [286]  0.957919018  1.649465193 -0.284178479  0.044809506  1.058629626
## [291] -0.519169898 -1.251000315  1.246622761 -0.673592830  1.535326504
## [296]  1.884456611 -0.633308587  1.374189531 -0.123041507  1.045201545
## [301]  1.414473774 -0.257322317  0.494650222 -0.425173330  1.038487505
## [306]  0.870636491  0.702785478  1.555468625  0.158948195  1.380903572
## [311] -0.176753831  0.111949912  0.615502951 -0.223752115 -1.425565369
## [316] -0.250608277 -0.217038074 -1.432279410  0.474508100 -1.123433545
## [321]  0.729641640  1.213052558  0.729641640 -0.472171614  1.723319638
## [326] -1.291284559 -0.754161316 -1.297998599  1.380903572  1.340619328
## [331] -1.116719505  0.346941330 -0.861585965 -0.908584248  0.064951628
## [336] -1.036151019 -0.123041507  1.360761450 -1.573274261  0.091807790
## [341] -0.646736668  1.186196396  0.031381425 -1.177145870  0.528220425
## [346]  0.024667385  1.320477207  0.353655371 -1.828407801  1.703177517
## [351] -1.257714356  1.548754585  1.065343667  1.669607314 -1.009294856
## [356] -1.291284559  1.689749436  1.562182666 -0.928726370 -0.425173330
## [361]  1.609180950 -0.673592830  1.347333369 -0.599738384 -0.344604844
## [366] -1.183859910 -0.552740101 -1.284570518 -1.539704058 -0.183467871
## [371]  0.783353965  0.447651938  1.166054275  0.064951628  1.649465193
## [376] -0.284178479 -0.539312019 -0.438601412  1.045201545 -0.002188777
## [381] -0.579596263 -0.875014046  0.957919018 -0.391603128 -1.620272544
## [386]  1.830744287 -0.572882222 -0.586310303  1.656179233 -1.432279410
## [391] -0.257322317 -0.767589397  1.313763166  1.219766599 -0.035758980
## [396] -1.056293140 -0.868300005  0.152234155 -1.284570518  1.313763166
## [401] -0.317748682 -0.029044939  0.393939614  0.561790627  0.662501235
## [406]  0.232802641  1.689749436  0.769925884  0.011239304  0.031381425
## [411] -1.479277693 -0.868300005 -0.734019195 -1.163717789 -0.304320601
## [416] -0.666878789 -0.693734952  0.830352248 -0.156611709  0.843780329
## [421]  1.333905288  1.824030246  1.246622761 -0.284178479  0.957919018
## [426] -0.324462723 -1.170431829  0.333513249  0.722927600  1.871028530
## [431] -0.975724654 -1.586702342 -0.955582532 -0.915298289  0.984775180
## [436] -0.337890804 -0.948868492  0.158948195 -0.619880506 -1.331568802
## [441]  0.776639924 -1.398709207 -1.324854761 -0.747447276 -0.740733235
## [446]  0.501364262 -1.136861626 -0.398317168 -0.525883938 -1.532990018
## [451] -1.418851329  0.978061140 -0.975724654  1.320477207 -1.324854761
## [456]  0.810210127 -0.720591114 -0.478885655 -0.982438694  0.984775180
## [461]  0.843780329  0.179090317 -0.935440411 -1.331568802 -0.559454141
## [466]  0.367083452  0.487936181  0.642359113  0.320085168 -0.774303438
## [471] -1.465849612  0.816924167  0.219374560  0.749783762  1.112341950
## [476]  0.581932749 -0.317748682  0.608788911 -1.103291424 -0.566168182
## [481] -0.156611709 -1.633700626 -0.290892520  1.045201545 -1.002580816
## [486]  0.937776897  0.655787195 -0.062615142  1.642751152 -1.613558504
## [491] -1.412137288  1.166054275  1.783746003 -1.586702342 -1.157003748
## [496] -0.895156167 -1.264428396  0.628931032  1.300335085 -0.264036358
```

7.5 Correlation Coefficient

```
(1/(length(x) - 1)) * sum(((x - mean(x))/(sd(x))) * ((y - mean(y))/(sd(y))))
```

```
## [1] -0.9542196
```

```
# built-in function: cor()
```

7.5.1 Function

```
corr_coeff <- function(x, y) {
  (1/(length(x) - 1)) * sum(((x - mean(x))/(sd(x))) * ((y - mean(y))/(sd(y))))
}
corr_coeff(x, y)
```

```
## [1] -0.9542196
```

```
corr_coeff(a, b)
```

```
## [1] 0.1113098
```

7.6 Correlation Coefficient (2)

```
sum((x - mean(x)) * (y - mean(y)))/( sqrt(sum((x - mean(x))^2)) * sqrt(sum((y - mean(y))^2)))
```

```
## [1] -0.9542196
```

7.6.1 Function

```
corr_coeff_2 <- function(x, y) {
  sum((x - mean(x)) * (y - mean(y)))/( sqrt(sum((x - mean(x))^2)) * sqrt(sum((y - mean(y))^2)))
}
corr_coeff_2(x, y)
```

```
## [1] -0.9542196
```

```
corr_coeff_2(a, b)
```

```
## [1] 0.1113098
```

7.7 Regression Parameters

$$\beta_1 = r_{xy}\left(\frac{s_y}{s_x}\right)$$

$$\beta_0 = \bar{y} - \beta_1\bar{x}$$

```
cor(x, y) * (sd(y)/sd(x))
```

```
## [1] -0.7173913
```

```
mean(y) - ((cor(x, y) * (sd(y)/sd(x))) * mean(x))
```

```
## [1] 8.586957
```

```
# built-in function: lm(y ~ x)$coefficients
```

7.7.1 Function

```r
beta_0 <- function(x, y) {
  mean(y) - ((cor(x, y) * (sd(y)/sd(x))) * mean(x))
}

beta_1 <- function(x, y) {
  cor(x, y) * (sd(y)/sd(x))
}

beta_0(x, y)
```

```
## [1] 8.586957
```

```r
beta_1(x, y)
```

```
## [1] -0.7173913
```

```r
beta_0(a, b)
```

```
## [1] 271.1084
```

```r
beta_1(a, b)
```

```
## [1] 0.562306
```

7.8 Gini Coefficient

```r
x <- c(1000, 200, 500, 5000, 10000)
y <- c(70000, 1110, 3500, 11400, 2000)
```

```r
(2 * sum((seq_along(x) * x)))/(length(x) * sum(x)) - ((length(x)+1)/length(x))
```

```
## [1] 0.5461078
```

```r
(2 * sum((seq_along(y) * y)))/(length(y) * sum(y)) - ((length(y)+1)/length(y))
```

```
## [1] -0.5713442
```

7.8.1 Function

```r
gini <- function(x) {
  (2 * sum((seq_along(x) * x)))/(length(x) * sum(x)) - ((length(x)+1)/length(x))
}

gini(x)
```

```
## [1] 0.5461078
```

```r
gini(y)
```

```
## [1] -0.5713442
```

```r
gini(a)
```

```
## [1] 0.007479256
```

```r
gini(b)
```

```
## [1] -0.004710008
```

7.9 Cronbach's Alpha

```r
x <- c(8, 3, 4, 1, 9)
y <- c(2, 7, 5, 8, 3)
```

```r
(2)/(2-1) * (1 - ((var(x) + var(y))/(var(x+y))))
```

```
## [1] -22
```

7.9.1 Function

```r
# Calculates Cronbach's alpha given 2 data columns
cronbach <- function(x, y) {
  (2)/(2-1) * (1 - ((var(x) + var(y))/(var(x+y))))
}

cronbach(a, b)
```

```
## [1] 0.08136245
```

7.10 Coefficient of Skewness

```r
sqrt(length(x)) * sum((x - mean(x))^3) / (sum((x - mean(x))^2)^(3 / 2))
```

```
## [1] 0.1290092
```

```r
sqrt(length(y)) * sum((y - mean(y))^3) / (sum((y - mean(y))^2)^(3 / 2))
```

```
## [1] 0
```

```r
# One-pass version
sqrt(length(x)) * (sum(x^3) - 3 * mean(x) * sum(x^2) + 2 * length(x) * mean(x)^3) / (sum(x^2) - length(
```

```
## [1] 0.1290092
```

```r
sqrt(length(y)) * (sum(y^3) - 3 * mean(y) * sum(y^2) + 2 * length(y) * mean(y)^3) / (sum(y^2) - length(
```

```
## [1] 0
```

7.10.1 Function

```r
skew <- function(x) {
  deviations <- x - mean(x)
  sqrt(length(x)) * sum(deviations^3) / (sum(deviations^2)^(3 / 2))
}

skew(x)
```

```
## [1] 0.1290092
```

```r
skew(y)
```

```
## [1] 0
```

```r
skew(a)
```

```
## [1] 0.06613233
```

```r
skew(b)
```

```
## [1] 0.09890234
```

```r
skew_one_pass <- function(x)
{
  n <- length(x)
  mu_x <- mean(x)
  sqrt(n) * (sum(x^3) - 3 * mu_x * sum(x^2) + 2 * n * mu_x^3) / (sum(x^2) - n * mu_x^2)^(3 / 2)
}

skew_one_pass(x)
```

```
## [1] 0.1290092
```

```r
skew_one_pass(y)
```

```
## [1] 0
```

```r
skew_one_pass(a)
```

```
## [1] 0.06613233
```

```r
skew_one_pass(b)
```

```
## [1] 0.09890234
```