Lab 7 Correlation (part 1)

B. Sosnovski

03/28/2022

Correlation (Part 1)

The idea of correlation was discussed in the lecture and the textbook. This is the idea that two things that we measure can be somehow related to one another.

Some of the relationships we can measure are meaningful and might reflect a causal relationship where one thing causes a change in another. Some of the relationships are spurious and do not reflect a causal relationship.

In this lab, you will learn how to compute correlations between two variables in software and then ask questions about the correlations you observe.

General Goals

The goals for this lab are the following:

1. To compute Pearson's r (correlation coefficient) between two variables using software
2. To discuss the possible meaning of correlations that you observe

Important Info

We use data from the World Happiness Report (http://worldhappiness.report). They just released the 10th Anniversary Edition of their report for data for 2022.

The 2018 and 2022 data sets from Happiness Report are already available to you in this lab's folder on Cocalc.com (https://cocalc.com).

cor for correlation

R has the `cor` function for computing correlation coefficient *r* (Pearson's coefficient) between any two variables. This same function computes other correlation versions, but we'll skip those here. To use the function, you need two variables with numbers like these:

```
x  <- c(5,3,2,5,4,6,4,8,7)
y  <- c(6,5,6,7,8,9,8,10,12)
cor(x,y)
```

```
## [1] 0.8022222
```

Example

Create two lists of 5 numbers and ask R to calculate the correlation. Use variables *w* and *z* to represent your lists.

```
Enter your code below:
```

Generating a uniform distribution

You can use the `runif()` function to sample random numbers from a uniform distribution between a minimum and a maximum value.

In the example below, sample 15 (n=15) random numbers between the range 8 (min = 8) and 30 (max=30). Every time you run this code, the 15 values in x will be re-sampled, and 15 new random numbers.

```
x <- runif(n=15, min=8, max=30)
x
```

```
##  [1] 11.30572 20.02493 19.67985 10.28725 27.07067 27.45102 23.40553 17.73924
##  [9] 24.71006 25.60447 17.05964 13.34574 14.60198 14.86911 28.70322
```

You can compute the correlation between two sets of random numbers by first sampling random numbers into each variable and then running the `cor()` function.

```
x <- runif(n=15, min=8, max=30)
y <- runif(n=15, min=8, max=30)
cor(x,y)
```

```
## [1] 0.1561992
```

Running the above code will give different values for the correlation each time because the numbers in x and y are always randomly different.

As you learned from the textbook, we can find correlations by chance alone, even when there is no actual correlation between the variables. For example, if we sampled randomly into x and then sampled some numbers randomly into y. We know they aren't related because we randomly sampled the numbers. However, doing this creates some correlations by chance.

Chance correlations

Let's explore the idea that correlations between two measures can arise by chance alone by generating a sample of 50 correlations and then estimate the mean of these correlations generated by chance. We should expect that the mean of these correlations should be zero because they rise from randomly selected x and y distributions.

One way to estimate the mean of the correlations is to repeat the above code many times. For example, if you run the above code 50 times, you could save the correlations each time, then compute the mean. This would estimate the mean of correlations that can be produced by chance. How can you repeat the above code many times to solve this problem?

We can do this using a `for` loop. The code below shows how to repeat everything inside the for loop 50 times. The variable `i` is an index that goes from 1 to 50. The `saved_value` variable starts as an empty variable, and then we put a value into it (at index position i, from 1 to 50). In this code, we put the correlations of x and y into the `saved_value` variable. At the end of the simulation, the `save_value` variable contains 50 numbers. The `mean()` function is used to find the mean value for the 50 simulations.

Doing this will allow you to run the simulation 50 times and find the mean correlation that arises by chance.

```
saved_value <- c() #make an empty variable
for (i in 1:50){
  x <- runif(n=15, min=8, max=30)
  y <- runif(n=15, min=8, max=30)
  saved_value[i] <- cor(x,y)
}
saved_value
```

```
##  [1]  0.197354529  0.314166022 -0.237419317  0.193519479  0.110485387
##  [6]  0.099368396 -0.224797280 -0.727794282  0.022174253  0.364544551
## [11]  0.176770579  0.063885830  0.467487926  0.261369272  0.128532368
## [16]  0.269863577  0.234850039  0.167752832 -0.281779922 -0.242647863
## [21]  0.058138588  0.249817708  0.370316476  0.088809782 -0.324128431
## [26] -0.120780235  0.096036104 -0.348387605 -0.356161642 -0.166270355
## [31] -0.127472605  0.376108854  0.435273519 -0.122272093  0.432920882
## [36] -0.004847941 -0.003145139 -0.528588245  0.252776704  0.261889997
## [41] -0.015062336  0.223625040  0.099466188  0.056224865  0.273408441
## [46] -0.154305963 -0.481217924  0.378886779  0.357404275  0.195712990
```

```
mean(saved_value)
```

```
## [1] 0.05623726
```

You can see that the mean is not always zero. Sometimes it looks like there is a correlation when we know there shouldn't be. You can keep re-doing these simulations by re-building (re-knitting) your R Markdown document. You simulate the outcomes as often as you press the "Build" button.

Example

Create two lists of 100 numbers and use R to calculate the standard deviation of the correlations.

```
Enter your code below:
```

World Happiness Report

Let's take a look at some correlations in real data. We are going to look at responses to a questionnaire about happiness that was sent around the world from the world happiness report (http://worldhappiness.report)

Load the data

We load the data into a data frame. Reminder, the following assumes that you have downloaded the data.

```
library(data.table)
whr18_data <- fread('WHR2018.csv')
```

If the Internet is enabled in your project, you can also load the data using the following URL

```
library(data.table)
whr_data <- fread("https://raw.githubusercontent.com/CrumpLab/statisticsLab/master/data/WHR2018.csv")
```

Look at the data

```
dim(whr18_data)
```

```
## [1] 1562   19
```

```
colnames(whr18_data)
```

```
##  [1] "country"
##  [2] "year"
##  [3] "Life Ladder"
##  [4] "Log GDP per capita"
##  [5] "Social support"
##  [6] "Healthy life expectancy at birth"
##  [7] "Freedom to make life choices"
##  [8] "Generosity"
##  [9] "Perceptions of corruption"
## [10] "Positive affect"
## [11] "Negative affect"
## [12] "Confidence in national government"
## [13] "Democratic Quality"
## [14] "Delivery Quality"
## [15] "Standard deviation of ladder by country-year"
## [16] "Standard deviation/Mean of ladder by country-year"
## [17] "GINI index (World Bank estimate)"
## [18] "GINI index (World Bank estimate), average 2000-15"
## [19] "gini of household income reported in Gallup, by wp5-year"
```

```
head(whr18_data)
```

```
##        country year Life Ladder Log GDP per capita Social support
## 1: Afghanistan 2008    3.723590           7.168690      0.4506623
## 2: Afghanistan 2009    4.401778           7.333790      0.5523084
## 3: Afghanistan 2010    4.758381           7.386629      0.5390752
## 4: Afghanistan 2011    3.831719           7.415019      0.5211036
## 5: Afghanistan 2012    3.782938           7.517126      0.5206367
## 6: Afghanistan 2013    3.572100           7.503376      0.4835519
##    Healthy life expectancy at birth Freedom to make life choices Generosity
## 1:                         49.20966                    0.7181143 0.18181947
## 2:                         49.62443                    0.6788964 0.20361446
## 3:                         50.00896                    0.6001272 0.13763019
## 4:                         50.36730                    0.4959014 0.17532922
## 5:                         50.70926                    0.5309350 0.24715924
## 6:                         51.04298                    0.5779554 0.07473493
##    Perceptions of corruption Positive affect Negative affect
## 1:                 0.8816863       0.5176372       0.2581955
## 2:                 0.8500354       0.5839256       0.2370924
## 3:                 0.7067661       0.6182654       0.2753238
## 4:                 0.7311085       0.6113873       0.2671747
## 5:                 0.7756198       0.7103847       0.2679191
## 6:                 0.8232041       0.6205848       0.2733281
##    Confidence in national government Democratic Quality Delivery Quality
## 1:                         0.6120721          -1.929690        -1.655084
## 2:                         0.6115452          -2.044093        -1.635025
## 3:                         0.2993574          -1.991810        -1.617176
## 4:                         0.3073857          -1.919018        -1.616221
## 5:                         0.4354402          -1.842996        -1.404078
## 6:                         0.4828473          -1.879709        -1.403036
##    Standard deviation of ladder by country-year
## 1:                                     1.774662
## 2:                                     1.722688
## 3:                                     1.878622
## 4:                                     1.785360
## 5:                                     1.798283
## 6:                                     1.223690
##    Standard deviation/Mean of ladder by country-year
## 1:                                         0.4765997
## 2:                                         0.3913617
## 3:                                         0.3948027
## 4:                                         0.4659422
## 5:                                         0.4753669
## 6:                                         0.3425687
##    GINI index (World Bank estimate)
## 1:                               NA
## 2:                               NA
## 3:                               NA
## 4:                               NA
## 5:                               NA
## 6:                               NA
##    GINI index (World Bank estimate), average 2000-15
## 1:                                                NA
## 2:                                                NA
```

```
## 3:                                                    NA
## 4:                                                    NA
## 5:                                                    NA
## 6:                                                    NA
##    gini of household income reported in Gallup, by wp5-year
## 1:                                                       NA
## 2:                                                0.4419058
## 3:                                                0.3273182
## 4:                                                0.3367642
## 5:                                                0.3445396
## 6:                                                0.3043685
```

You should see that there is data for many different countries across a few different years. There are many different kinds of measures, each given a name. I'll show you some examples of asking about correlations with this data; then, you get to ask and answer your questions.

Question

For the year 2017 only, does a country's measure for "freedom to make life choices" correlate with the country's measure for "Confidence in national government"?

Let's find out. We calculate the correlation:

```
x <- whr18_data$`Freedom to make life choices`
y <- whr18_data$`Confidence in national government`
cor(x,y)
```

```
## [1] NA
```

What happened here? The correlation was `NA` (meaning undefined).

Let's look at the first 100 values of the variable `x`.

```
head(x,100)
```

```
##   [1] 0.7181143 0.6788964 0.6001272 0.4959014 0.5309350 0.5779554 0.5085140
##   [8] 0.3889276 0.5225662 0.4270109 0.5286047 0.5252233 0.5689584 0.4874963
##  [15] 0.6015121 0.6318303 0.7346484 0.7038507 0.7298189 0.7496110 0.5926958
##  [22] 0.5295613 0.5866635        NA        NA 0.4366705 0.5837018 0.4560286
##  [29] 0.4095549 0.3745416 0.7330037 0.6528326 0.6782222 0.6366464 0.7302582
##  [36] 0.8158020 0.7474984 0.7372504 0.7450578 0.8812237 0.8477022 0.8319662
##  [43] 0.5201978 0.6054108 0.4621566 0.4414127 0.4592567 0.4645247 0.5018638
##  [50] 0.5040818 0.5064871 0.5510266 0.6109869 0.6136971 0.9349733 0.8906820
##  [57] 0.9157333 0.9320586 0.9445865 0.9351463 0.9333792 0.9229323 0.9218710
##  [64] 0.9223157 0.9105502 0.9413823 0.8790693 0.8959798 0.9393556 0.9197039
##  [71] 0.9217345 0.8850269 0.9003052 0.8885140 0.8900306 0.7715282 0.5220463
##  [78] 0.6010433 0.4981384 0.5010711 0.5374843 0.5988591 0.6719570 0.7327729
##  [85] 0.7642895 0.7125732 0.7310305 0.8959314 0.8620029 0.8698704 0.6818229
##  [92] 0.8092059        NA 0.8495212 0.8886911 0.9058585 0.6116642 0.6045383
##  [99] 0.6060122 0.6309311
```

The correlation was `NA` because there are some missing data points. We can remove all the rows with missing data first, then do the correlation.

Removing NA values

We will do this in a couple of steps. First, create our own data frame with only the numbers we want to analyze. We can select the columns we want to keep using `select`. Then we use `filter` to remove the rows with NAs.

```
library(dplyr)
smaller_df <- whr18_data %>%
               select(country, year,
                      `Freedom to make life choices`,
                      `Confidence in national government`) %>%
               filter(!is.na(`Freedom to make life choices`),
                      !is.na(`Confidence in national government`))

smaller_df <- smaller_df %>% filter(year=='2017')
sx <- smaller_df$`Freedom to make life choices`
sy <- smaller_df$`Confidence in national government`
r <- cor(sx,sy)
r
```

```
## [1] 0.4419082
```

Now we see the correlation is 0.4419082.

This shows that as `Freedom to make life choices` increases in a country, that country's confidence in its national government also increases. This is a positive correlation.

Example

Find the correlation for the newest data report (2022) from the World Happiness Report. Choose the variables of your interest.

```
Enter your code below:
```

References

The material used in this document contains excerpts and modifications from:

- Matthew J. C. Crump, Anjali Krishnan, Stephen Volz, and Alla Chavarga (2018) "Answering questions with data: Lab Manual". Last compiled on 2019-04-06. https://www.crumplab.com/statisticsLab/ (https://www.crumplab.com/statisticsLab/)