Lab 8 Correlation (Part 2)

B. Sosnovski

03/28/2022

Correlation (Part 2)

In this lab, we will continue to discuss the correlation between two variables. You will learn to make scatter plots and regression lines in R.

General Goals

The goals for this lab are the following:

1. To make the scatter plot using R
2. To discuss the possible meaning of correlations observed
3. To plot the regression line

Important Info

We continue to use data from the World Happiness Report (http://worldhappiness.report) for 2018 and 2022.

Scatter plots

Let's again generate two distributions, *x* and *y*, and plot the data in a scatter plot using ggplot2. Let's also return the correlation and print it on the scatter plot.

```
x <- c(4,15,5,14,9,8,17,10,6,7)
y <- c(2,8,17,4,9,14,3,2,11,12)
x
```

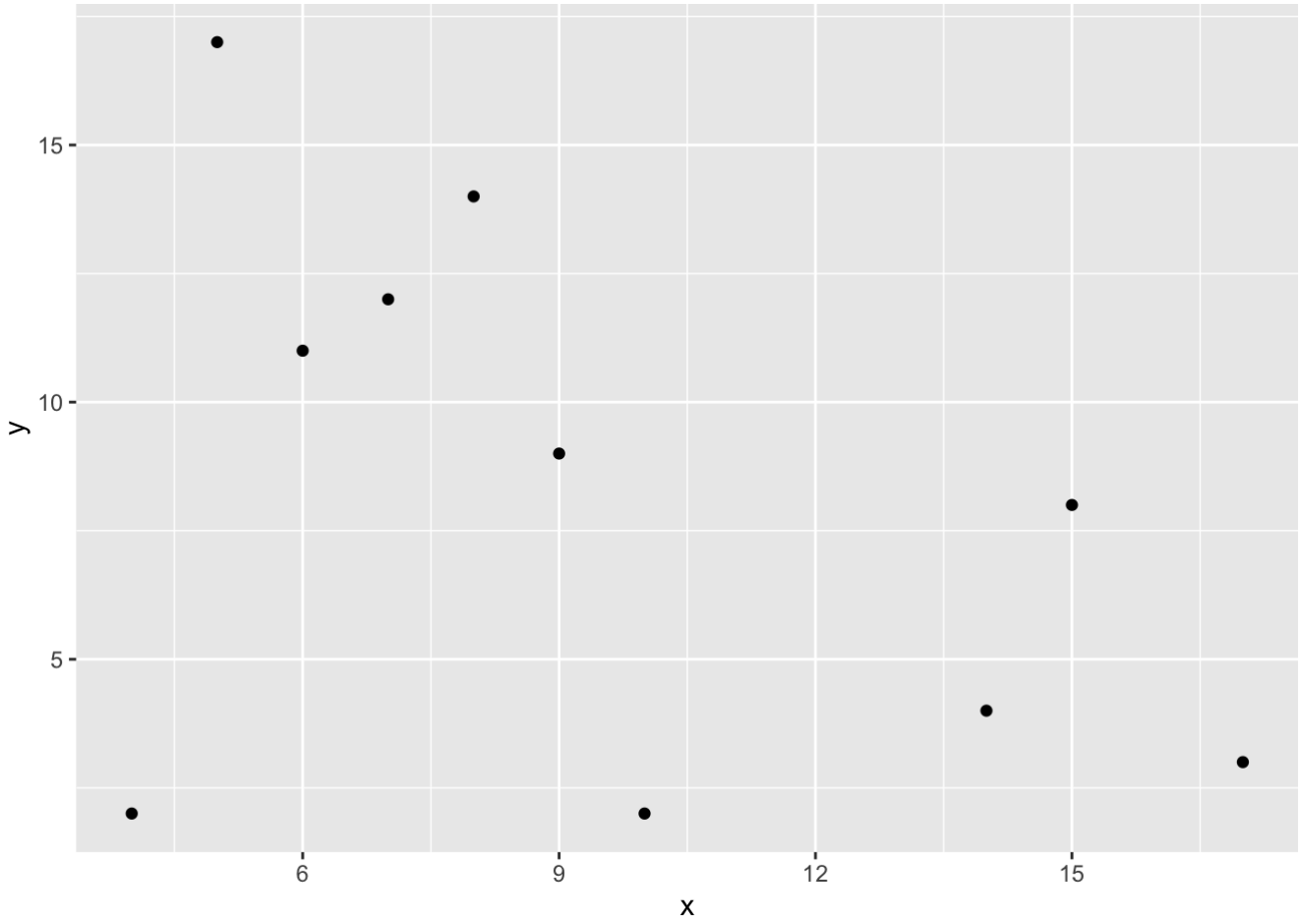
```
##  [1]  4 15  5 14  9  8 17 10  6  7
```

```
y
```

```
##  [1]  2  8 17  4  9 14  3  2 11 12
```

Note: To use ggplot2 to graph the scatterplot, we need the data to be in a data frame, so we first put our x and y variables in a data frame.

```
library(ggplot2)
# create a data frame for plotting
my_df <- data.frame(x,y)
# plot it
ggplot(my_df, aes(x=x,y=y))+
  geom_point()
```



Example

Create your data set with 8 pairs for variables *w* and *z* and plot the scatter plot.

```
Enter your code below:
```

Lots of scatter plots

Before we move on to real data, let's look at artificial data. Often we will have many measures of X and Y, split between a few different conditions, for example, A, B, C, and D. Let's make some data for X and Y, for each condition A, B, C, and D, and then use facet_wrapping to look at four scatter plots all at once.

1. Let's create the data sets

```
x<-rnorm(40,0,1)
y<-rnorm(40,0,1)
conditions<-rep(c("A", "B", "C","D"), each=10)
x
```

```
##  [1]  1.00513865  0.51715034 -0.13389259 -1.74683869  0.30595008 -1.63630141
##  [7]  1.37754925  1.76645955  1.15730440 -0.97831793 -1.24575025  0.12302010
## [13]  0.48991446  0.12867864 -0.16337073  1.07023166 -0.20743445 -0.56939507
## [19] -0.21957496  2.16491542 -2.62140334  0.40255606  0.45287969 -0.94703186
## [25]  0.02765056  2.01132678  2.29427902 -0.42894897 -1.01533787  1.12385548
## [31]  0.58346104 -0.93235609  0.87437994  0.46518096  0.39275287  0.83466825
## [37]  0.88370280  1.25726815  0.40054475  0.62436035
```

```
y
```

```
##  [1]  0.31488657  0.62214380 -0.35565284 -0.02565289 -1.19352334  1.44847116
##  [7]  1.71145421 -0.01559008  2.19643665 -0.88664316  0.41013246  0.75134810
## [13] -0.88801003 -1.26868172 -0.91785323  1.89902092 -0.48792219 -0.16294776
## [19] -0.24468192 -0.65749936 -1.17083028  0.61847074 -0.18249204  0.44901111
## [25]  1.29604297 -2.24155469 -0.58491156 -0.17959315 -0.22683510 -0.41136311
## [31]  1.48547183 -0.79364111  1.57279365 -1.38577774 -0.11157884 -1.46549434
## [37]  0.83630093  0.19649094  0.31929265  1.00426224
```

```
conditions
```

```
##  [1] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "B" "B" "B" "B" "B" "B" "B" "B" "B"
## [20] "B" "C" "C" "C" "C" "C" "C" "C" "C" "C" "C" "D" "D" "D" "D" "D" "D" "D" "D"
## [39] "D" "D"
```

2. Now we create the data frame (we need this to use ggplot2)

```
all_df <- data.frame(conditions, x, y)
# the following displays it nicely
knitr::kable(all_df)
```

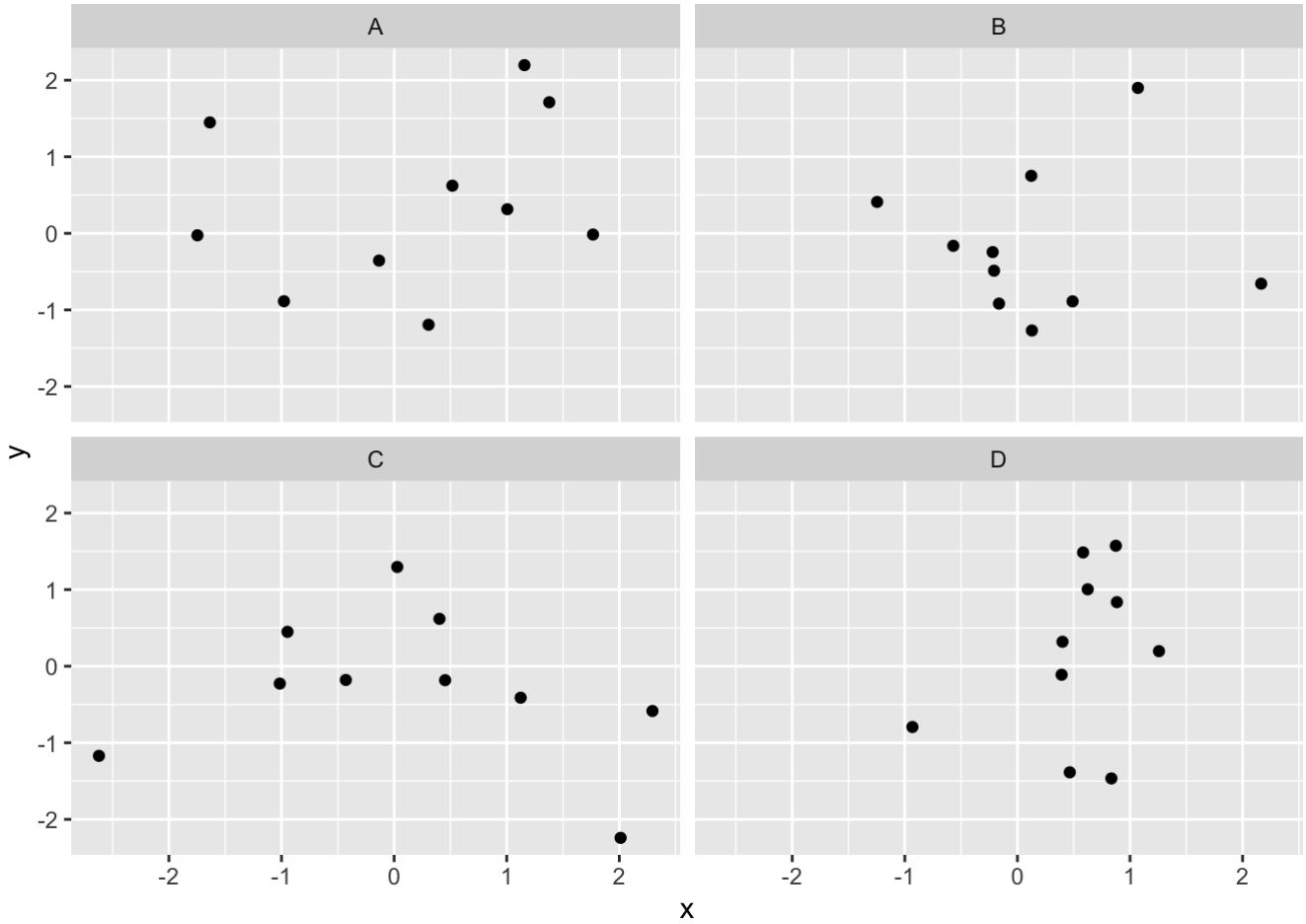
conditions	x	y
A	1.0051387	0.3148866
A	0.5171503	0.6221438
A	-0.1338926	-0.3556528
A	-1.7468387	-0.0256529
A	0.3059501	-1.1935233
A	-1.6363014	1.4484712
A	1.3775493	1.7114542

conditions	x	y
A	1.7664595	-0.0155901
A	1.1573044	2.1964366
A	-0.9783179	-0.8866432
B	-1.2457502	0.4101325
B	0.1230201	0.7513481
B	0.4899145	-0.8880100
B	0.1286786	-1.2686817
B	-0.1633707	-0.9178532
B	1.0702317	1.8990209
B	-0.2074344	-0.4879222
B	-0.5693951	-0.1629478
B	-0.2195750	-0.2446819
B	2.1649154	-0.6574994
C	-2.6214033	-1.1708303
C	0.4025561	0.6184707
C	0.4528797	-0.1824920
C	-0.9470319	0.4490111
C	0.0276506	1.2960430
C	2.0113268	-2.2415547
C	2.2942790	-0.5849116
C	-0.4289490	-0.1795931
C	-1.0153379	-0.2268351
C	1.1238555	-0.4113631
D	0.5834610	1.4854718
D	-0.9323561	-0.7936411
D	0.8743799	1.5727937
D	0.4651810	-1.3857777
D	0.3927529	-0.1115788
D	0.8346682	-1.4654943
D	0.8837028	0.8363009

conditions	x	y
D	1.2572682	0.1964909
D	0.4005447	0.3192926
D	0.6243604	1.0042622

3. Finally, we make the scatterplot for each condition (we covered the facet_wrap function in a previous lab)

```
ggplot(all_df, aes(x=x,y=y))+
  geom_point()+
  facet_wrap(~conditions)
```



We've seen how we can make four graphs at once. `Facet_wrap` will always try to make as many graphs as there are individual conditions in the column variable. In this case, there are four, so it makes four.

Example

Create a data set with 6 conditions that are people's names, each with 12 pairs of x and y from the uniform distribution (with parameters of your choice). Then plot the scatter plot.

```
Enter your code below:
```

Computing the correlations all at once

Note that the scatter plots don't show the correlation (r) values. We will make a table of the correlations in addition to the scatter plot. For this, we again use package `dplyr` .

Chance correlations (again!)

We will find some correlations by chance alone by generating artificial data.

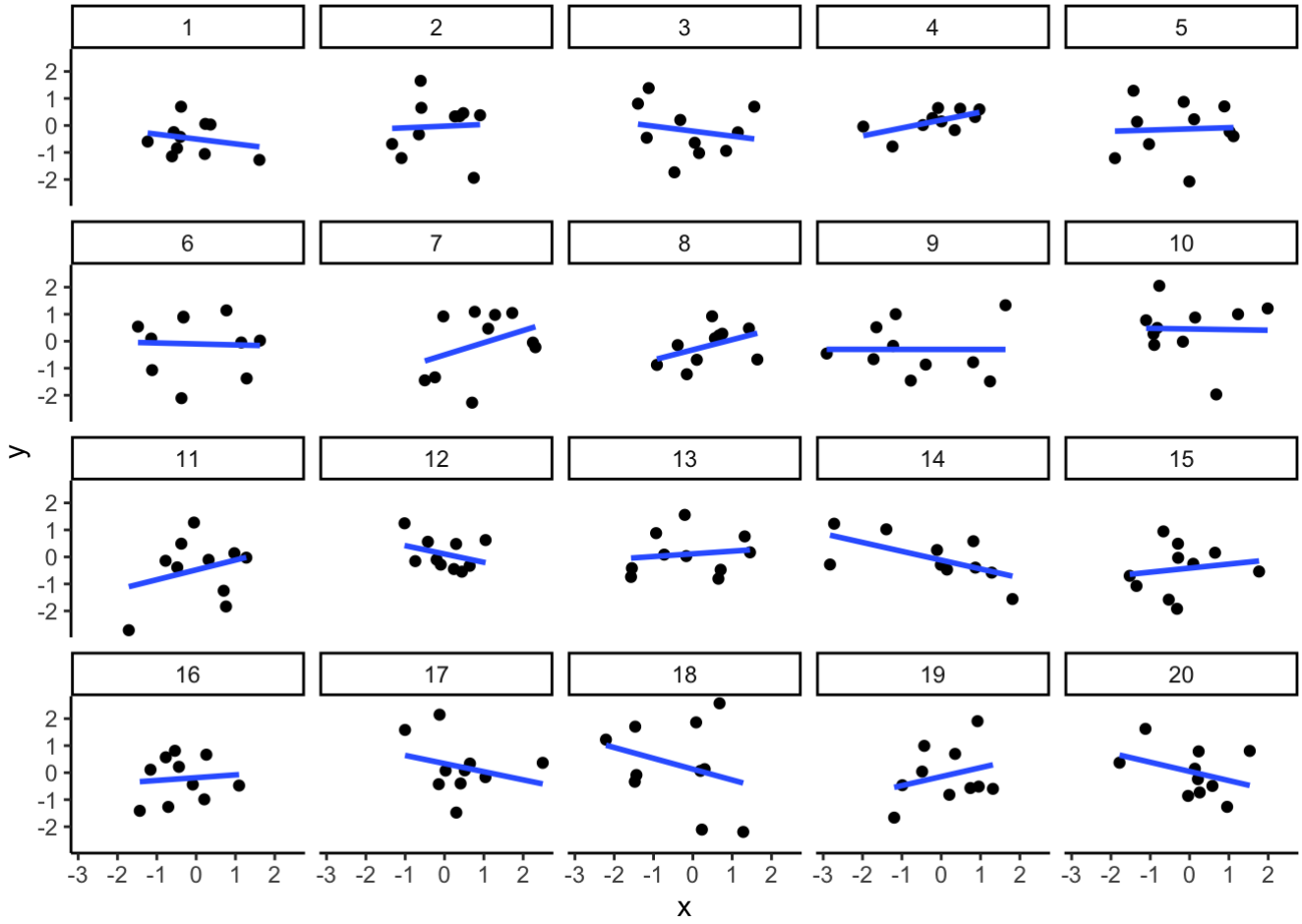
It's a repeat of what we did before, with a few more added conditions. Let's look at 20 conditions, with random numbers for *x* and *y* in each. For each, the sample size will be 10. We'll make the fake data, then make a big graph to look at all. And, even though we get to regression later in the lab, I'll put the best-fit line onto each scatter plot, so you can "see the correlations."

```
x<-rnorm(10*20,0,1)
y<-rnorm(10*20,0,1)
conditions<-rep(1:20, each=10)
all_df <- data.frame(conditions, x, y)
head(all_df)
```

```
##   conditions          x           y
## 1          1 -0.5691601 -0.24659687
## 2          1  0.2197037 -1.05690562
## 3          1 -0.4860749 -0.84263560
## 4          1  0.2330215  0.05924522
## 5          1 -0.4038984 -0.42330567
## 6          1 -0.6140673 -1.14351754
```

```
ggplot(all_df, aes(x=x,y=y))+
  geom_point()+
  geom_smooth(method=lm, se=FALSE)+
  facet_wrap(~conditions)+
  theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



You can see that the slope of the blue line is not always flat (slope zero). Sometimes there is a correlation when we know there shouldn't be. You can keep re-doing this graph by re-building (re-knitting) your R Markdown document. You simulate the outcomes as often as you press the "Build" button.

Example

Create a new data set with 6 conditions, each with 10 pairs of *x* and *y* from the uniform distributions with a min value of 5 and a max value of 10, then plot the scatter plot.

```
Enter your code below:
```

World Happiness Report

Let's look again at some correlations in real data.

Load the data

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

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

Question 1

We answer the same question as in the previous lab: 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"?

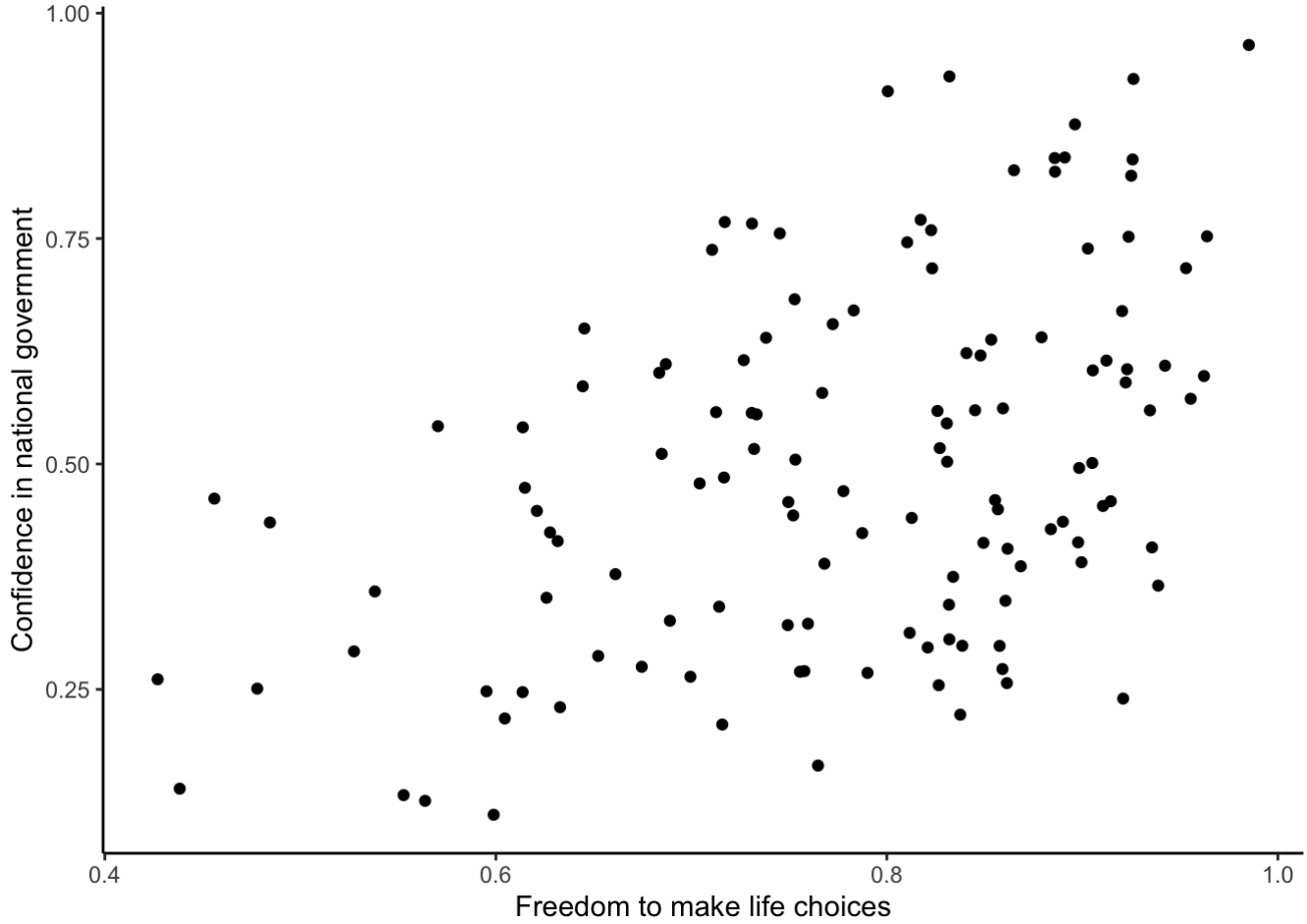
This time we make the scatter plot to answer the question.

```
# remove the NAs values
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')
smaller_df
```

```
##          country year Freedom to make life choices
##   1: Afghanistan 2017                     0.4270109
##   2:     Albania 2017                     0.7496110
##   3:   Argentina 2017                     0.8319662
##   4:     Armenia 2017                     0.6136971
##   5:   Australia 2017                     0.9105502
##  ---
## 124:     Uruguay 2017                     0.8978516
## 125:  Uzbekistan 2017                     0.9851778
## 126:       Yemen 2017                     0.5951908
## 127:      Zambia 2017                     0.8231686
## 128:    Zimbabwe 2017                     0.7528261
##      Confidence in national government
##   1:                         0.2611785
##   2:                         0.4577375
##   3:                         0.3054303
##   4:                         0.2469010
##   5:                         0.4534070
##  ---
## 124:                         0.4130321
## 125:                         0.9646904
## 126:                         0.2477870
## 127:                         0.7170041
## 128:                         0.6826467
```

```
# we make the plot
ggplot(smaller_df, aes(x=`Freedom to make life choices`,
                       y=`Confidence in national government`))+
  geom_point()+
  theme_classic()
```



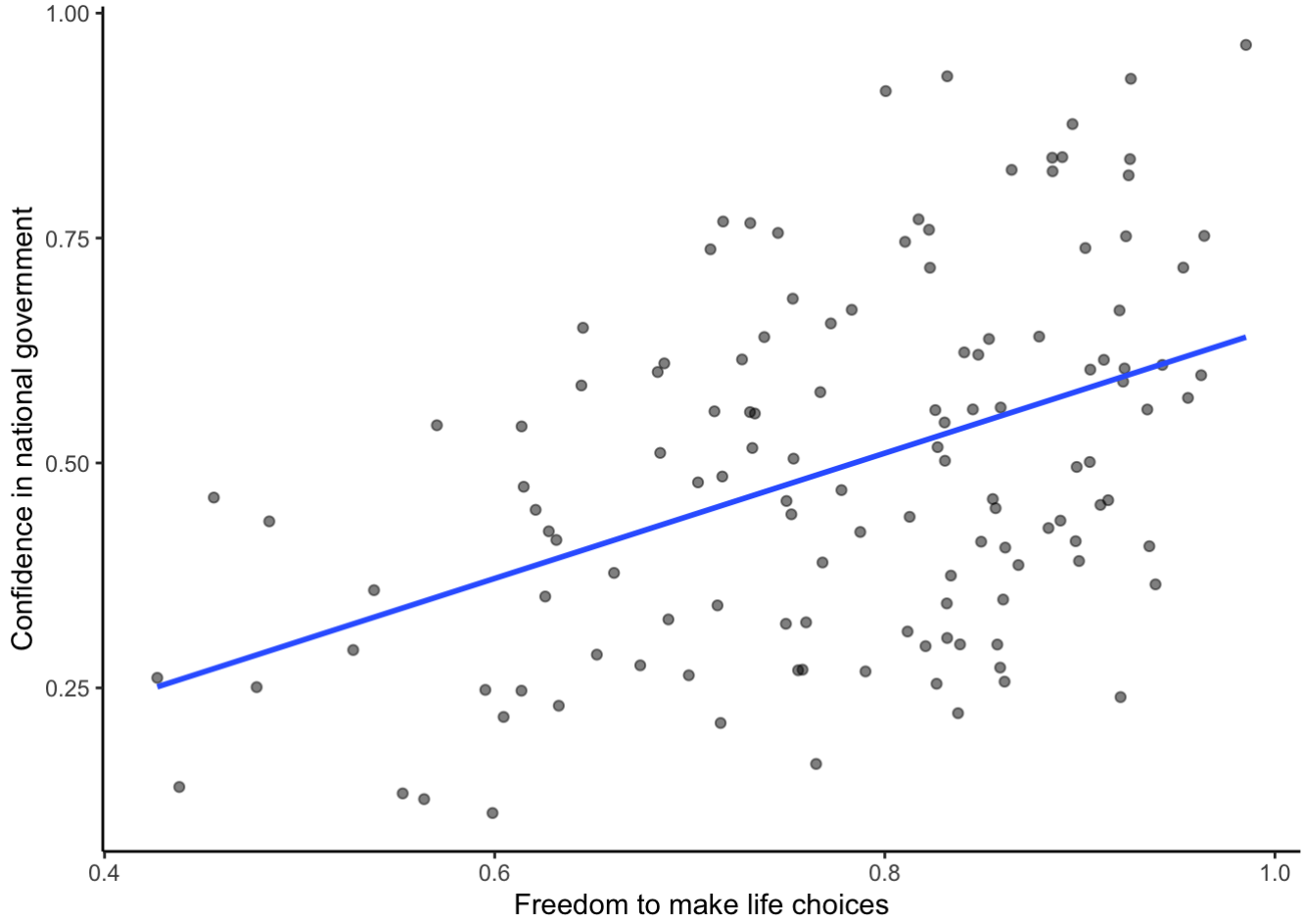
Although the scatter plot shows the dots are everywhere, it generally indicates that as `Freedom to make life choices` increases in a country, that country's confidence in its national government also increases. This is a positive correlation.

Best-fit Line and adjust the size of the points

Let's do this again and add the best-fit line so the trend is more apparent; we use `geom_smooth(method=lm, se=FALSE)` . I also changed the `alpha` value of the dots, so they blend in a bit, and you can see more of them.

```
# plot the data with the best-fit line
ggplot(smaller_df, aes(x=`Freedom to make life choices`,
                       y=`Confidence in national government`))+
  geom_point(alpha=.5)+
  geom_smooth(method=lm, se=FALSE)+
  theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Question 2

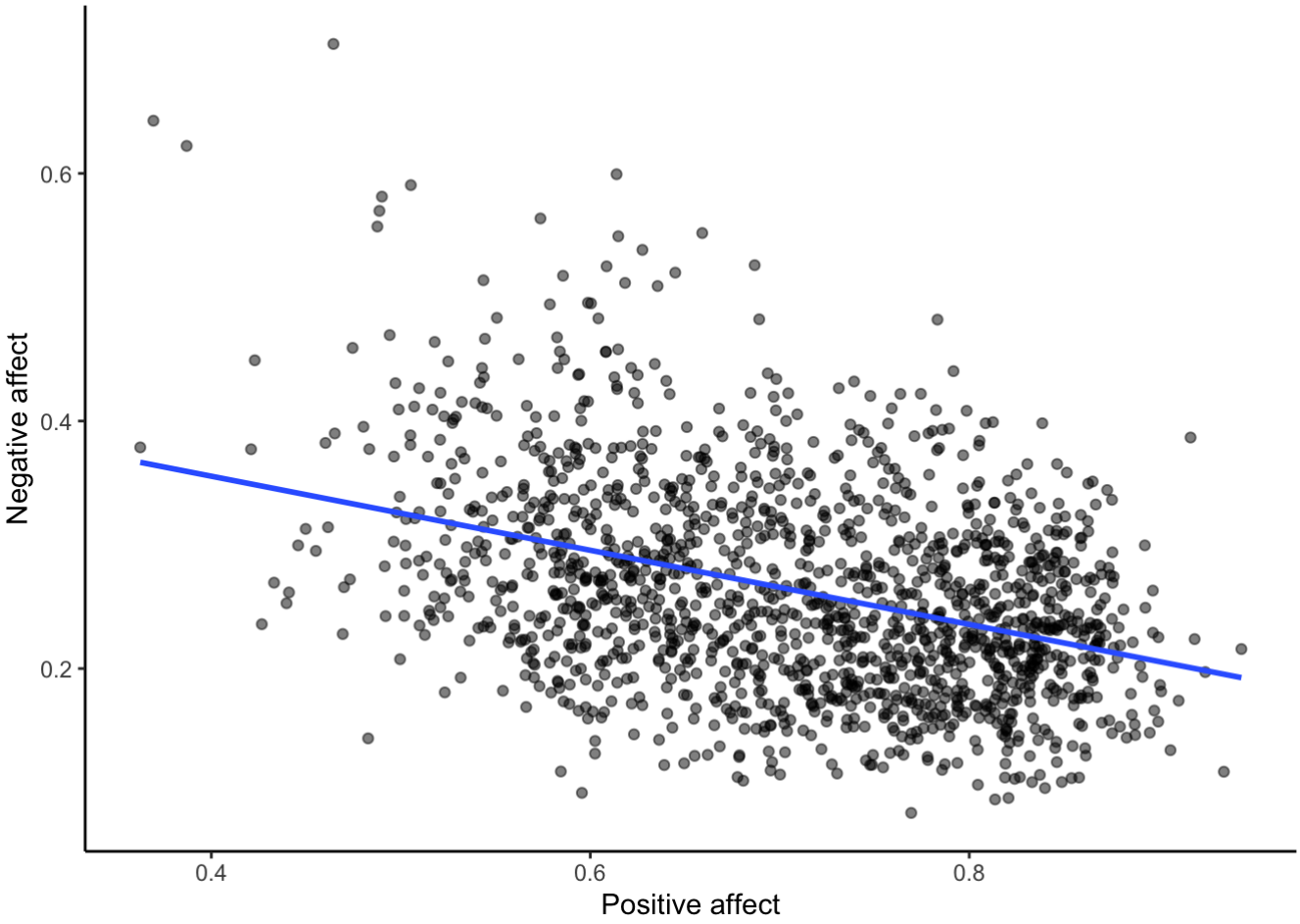
What is the relationship between positive affect in a country and negative affect in a country?

I wouldn't be surprised if there were a negative correlation here: when positive feelings generally go up, shouldn't negative feelings generally go down?

To answer this question, we copy and paste the last code block and change the variables to `Positive affect` and `Negative affect`.

```
# select DVs and filter for NAs
smaller_df <- whr18_data %>%
               select(country,
                      `Positive affect`,
                      `Negative affect`) %>%
               filter(!is.na(`Positive affect`),
                      !is.na(`Negative affect`))
# plot the data with the best-fit line
ggplot(smaller_df, aes(x=`Positive affect`,
                       y=`Negative affect`))+
  geom_point(alpha=.5)+
  geom_smooth(method=lm, se=FALSE)+
  theme_classic()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



As the positive effect goes up, the negative effect goes down. Thus, the negative correlation.

Example

Using the WHR2022.csv file, plot the scatter plot with the best-fit line between “Happiness_score” and “Explained_by_GDP_per_capita.”

```
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/)