

# Intro to Analysis

*Ben Chu*

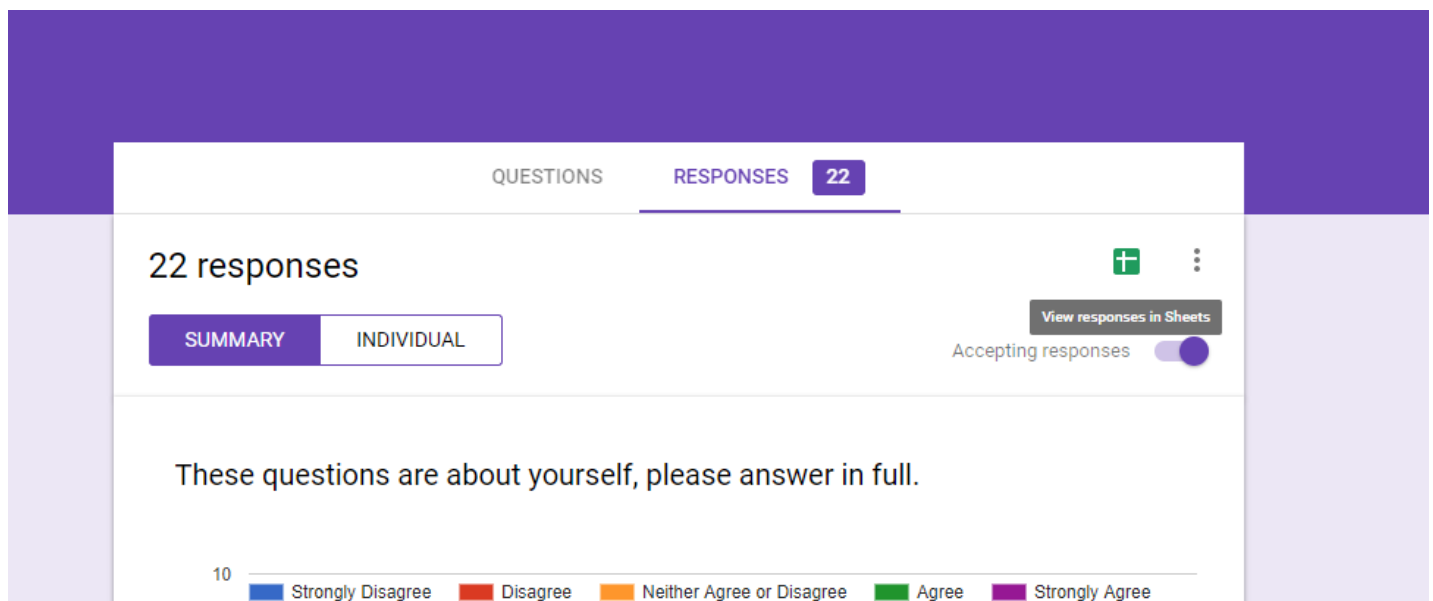
*March 4, 2018*

## This is a brief introduction into data analysis.

In this small handout, I will explain how to do basic data cleaning and analysis. This instruction is meant to show you how data scientists approach turning data into usable information. I want to further preface by saying there are many ways to clean data but this way is fairly fast and clean.

## First things to do is to find the data itself.

If you are using Google forms, please open the survey and switch over to the response section (next to questions) and click on the green icon next to the vertical dots.



Great!

It's your data! #Next thing we want to do is to make the variables we are using less complicated. We want to turn items into less confusing names because

"These.questions.are.about.yourself..please.answer.in.full...I.cried.many.times.this.year." is a really long unnecessary item. THANKS GOOGLE

This is usually row number 1, which is considered the headers for many data programs and sheets.

File Edit View Insert Format Data Tools Form Add-ons Help Last edit was sec				
100% \$ % .0 .00 123 Arial 10 B				
fx	These questions are about yourself, please answer in full. [I cried many times this year]			
	A	B	C	D
1	Timestamp	These questions are about	These questions are about	These questions are ab
2	2/26/2018 16:28:36	Strongly Disagree	Strongly Disagree	Strongly Disagree
3	2/26/2018 16:28:52	Strongly Disagree	Disagree	Strongly Disagree
4	2/26/2018 16:29:14	Disagree	Disagree	Disagree
5	2/26/2018 19:54:31	Agree	Agree	Neither Agree or Disagr
6	3/4/2018 18:11:02	Strongly Disagree	Strongly Disagree	Strongly Disagree

Go ahead and just replace it with a shortened version of your variable name. And keep doing so for all your variables.

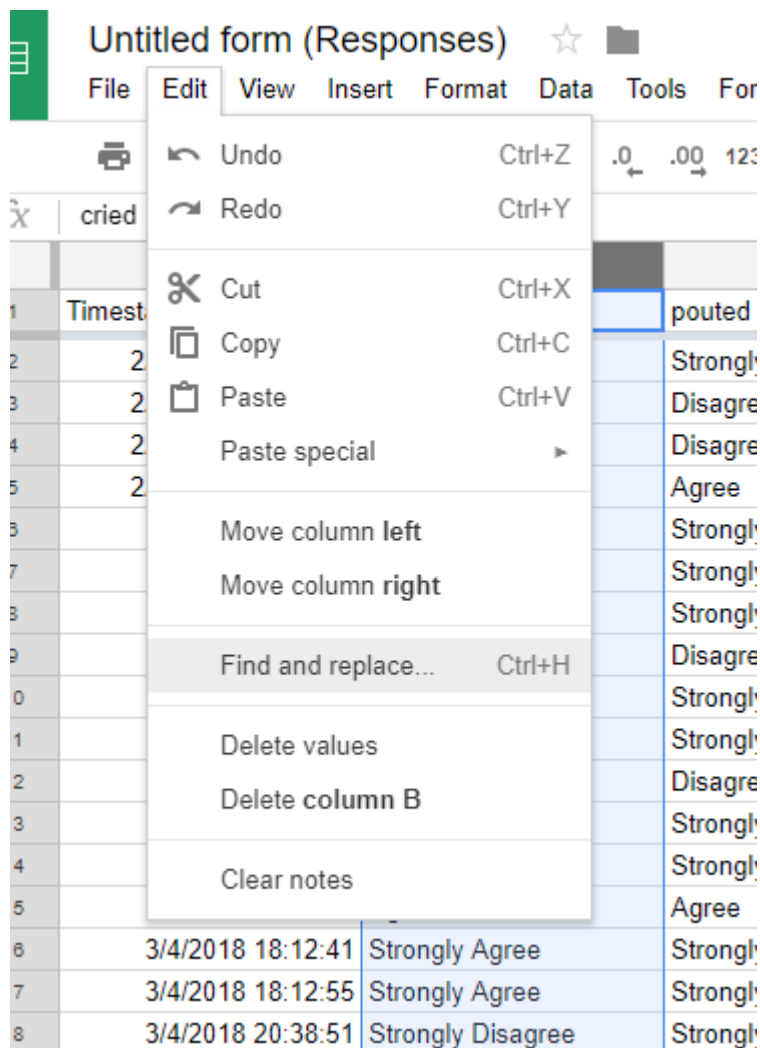
Untitled form (Responses) ☆ Last edit was 2 minutes ago						
File Edit View Insert Format Data Tools Form Add-ons Help Last edit was 2 minutes ago						
100% \$ % .0 .00 123 Arial 10 B I S A						
fx	cried					
	A	B	C	D	E	
1	Timestamp	cried				These
2	2/26/2018 16:28:36	Strongly Disagree	Strongly Disagree	Strongly Disagree	Strongly Agree	Strong
3	2/26/2018 16:28:52	Strongly Disagree	Disagree	Strongly Disagree	Agree	Disagr
4	2/26/2018 16:29:14	Disagree	Disagree	Disagree	Agree	Strong
5	2/26/2018 19:54:31	Agree	Agree	Neither Agree or Disagree	Agree	Disagr
6	3/4/2018 18:11:02	Strongly Disagree	Strongly Disagree	Strongly Disagree	Strongly Agree	Strong
7	3/4/2018 18:11:13	Strongly Disagree	Strongly Disagree	Strongly Disagree		Disagr
8	3/4/2018 18:11:13	Strongly Disagree	Strongly Disagree	Strongly Disagree	Strongly Disagree	Strong

Great ##

## Let's turn some characters into numerics.

Now that we have understandable names, we need to create understandable and analyzable responses. This is because character codes can't be compared to each other.

Find the Edit dropdown and click "find and replace" or Ctrl+H



Great, it should pop up a box with information. Please first select “Match entire cell contents”.

We want sheets to find information that says “strongly disagree” and replace it with “5”

	Disagree	Strongly Agree	Strongly Agree	A super fuck
Strongly Disagree	Strongly Disagree	Strongly Disagree	Strongly Disagree	Santa took t
Agree	Disagree	Agree	Strongly Agree	A super fuck
Strongly Disagree				Little
Strongly Disagree				Santa took t
Disagree				Santa took t
Disagree				Little
Strongly Agree				A super fuck
Disagree				Little
Strongly Agree				A super fuck
Strongly Agree				Alot
Neither Agree				Santa took t
Neither Agree				Little
Neither Agree				or Disagree
Strongly Agree				Average
Disagree				Quite a bit
Strongly Disagree				Alot
Disagree				Average
				Little
Neither Agree or Disagree	Neither Agree or Disagree	Neither Agree or Disagree	Neither Agree or Disagree	Quite a bit
Agree	Agree	Agree	Agree	Average
Strongly Agree	Strongly Agree	Strongly Agree	Strongly Agree	Little

Find and replace

Find

Strongly Disagree

Replace with

5

Search

Specific range

'Form Responses 1'!

☐ Match case
 ☒ Match entire cell contents
 ☐ Search using regular expressions [Help](#)
☐ Also search within formulas

Find

Replace

Replace all

Done

If you are using a Likert type scale, you would want

Strongly Disagree = 1 Disagree = 2 Neither Agree nor Disagree = 3 Agree = 4 Strongly Agree = 5

*note* If you have negatively worded items, you might want to reverse code them at this point. This means if an item is worded in this fashion I do not think I am great

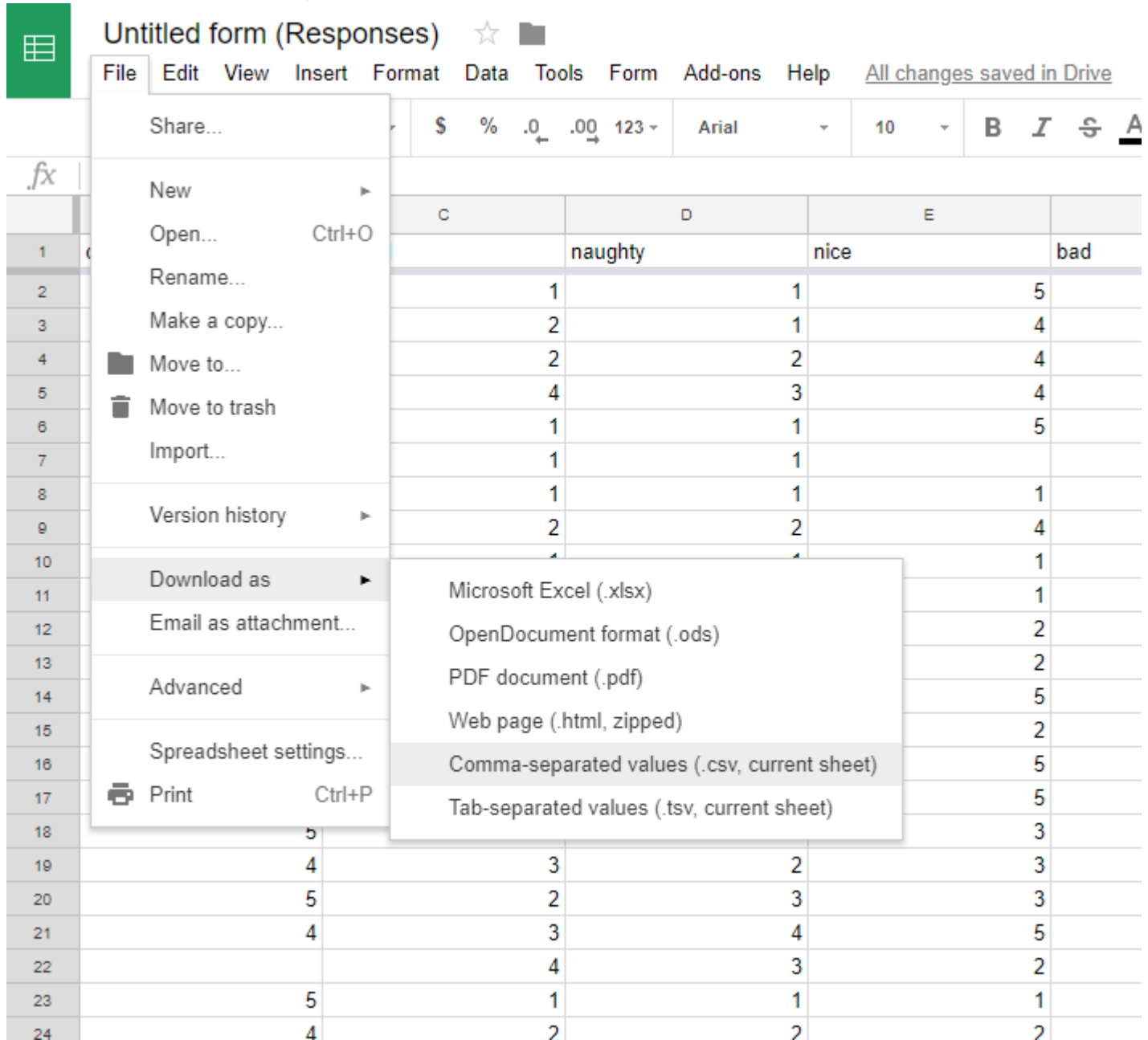
The scale would be the opposite. Strongly Disagree = 5 Disagree = 4 Neither Agree nor Disagree = 3 Agree = 2 Strongly Agree = 1

Untitled form (Responses)												
File Edit View Insert Format Data Tools Form Add-ons Help All changes saved in Drive												
100% \$ % .0 123 Arial 10 B I A												
	B	C	D	E	F	G	H	I	J	K	L	
1	cried	pouted	naughty	nice	bad	good	goodness4	horns	drums	stockings	gifts	
2		5	1	1	5	5	5	5	6	6	6	€
3		5	2		4	4	5	4	5	6	5	€
4		4	2	2	4	5	4	5	5	4	5	4
5		2	4	3	4	4	4	5	6	6	6	€
6		5	1	1	5	5	5	5	6	6	6	€
7		5	1	1		4	5	5	6	6	6	€
8		5	1	1	1	5	1	1	1	1	1	1
9		4	2	2	4	4	4	5	6	6	6	€
10		5	1	1	1	5	1	1	2	2	2	2
11		1	5	5	1	5	1	1	1	1	1	1
12		4	2	2	2	4	2	2	1	1	1	1
13		1	5	5	2	4	2	2	2	2	2	2
14		1	5	5	5	5	5	5	6	6	6	€
15		2	4	4	2	4	1	1	2	2	1	2
16		1	5	5	5	5	5	5	6	6	6	€
17		1	5	5	5	5	5		5	5	5	€
18		5	1	1	3	4	4	5	1	1	1	1

Whew, that's alot of copy and pasting.

Now we wanto do download it as a .csv and upload it into Rstudio.

File>download as> comma seperated value.



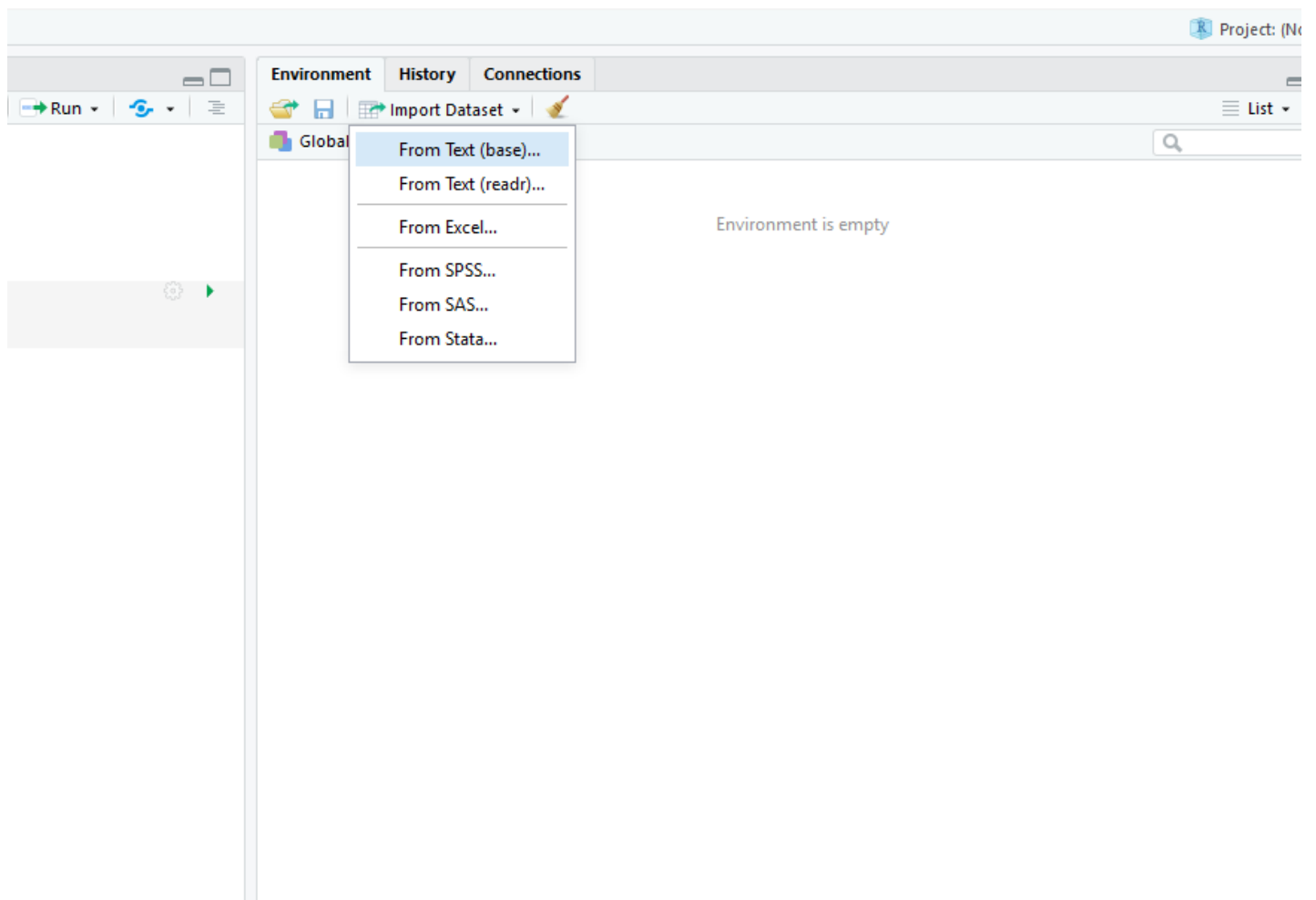
The screenshot shows a Google Sheets spreadsheet titled 'Untitled form (Responses)'. The 'File' menu is open, and the 'Download as' option is selected, which has opened a submenu. In the submenu, 'Comma-separated values (.csv, current sheet)' is highlighted. The spreadsheet data is as follows:

	C	D	E	
		naughty	nice	bad
1				
2	1	1		5
3	2	1		4
4	2	2		4
5	4	3		4
6	1	1		5
7	1	1		
8	1	1		1
9	2	2		4
10				1
11				1
12				2
13				2
14				5
15				2
16				5
17				5
18	5			3
19	4	3	2	3
20	5	2	3	3
21	4	3	4	5
22		4	3	2
23	5	1	1	1
24	4	2	2	2

Next thing to do is to import it into your environment. My file ended up being saved as `Santa.Correlation.csv`

## Rstudio and data analysis

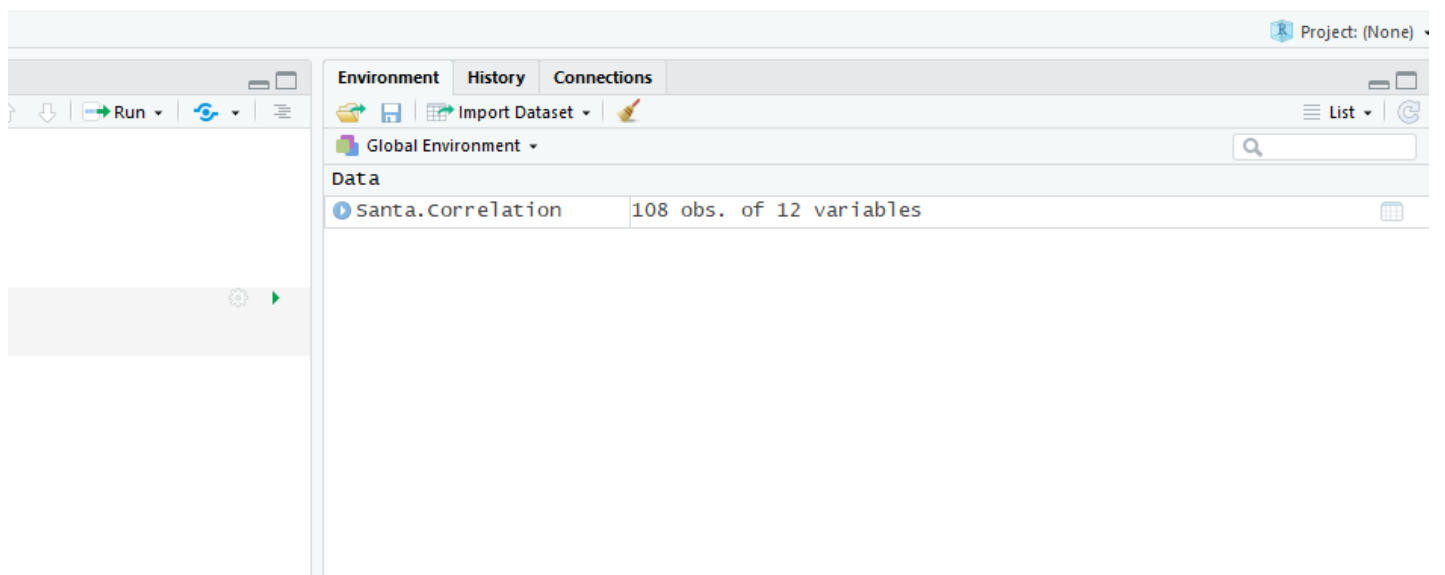
We want to import our data into Rstudio so we can run correlations and etc.



You can also do this via the command line.

```
Santa.Correlation <- read.csv("C:/Users/Branly McIanbry/Desktop/Santa.Correlation.csv", header=TRUE)
```

Nice! it should appear in your global environment

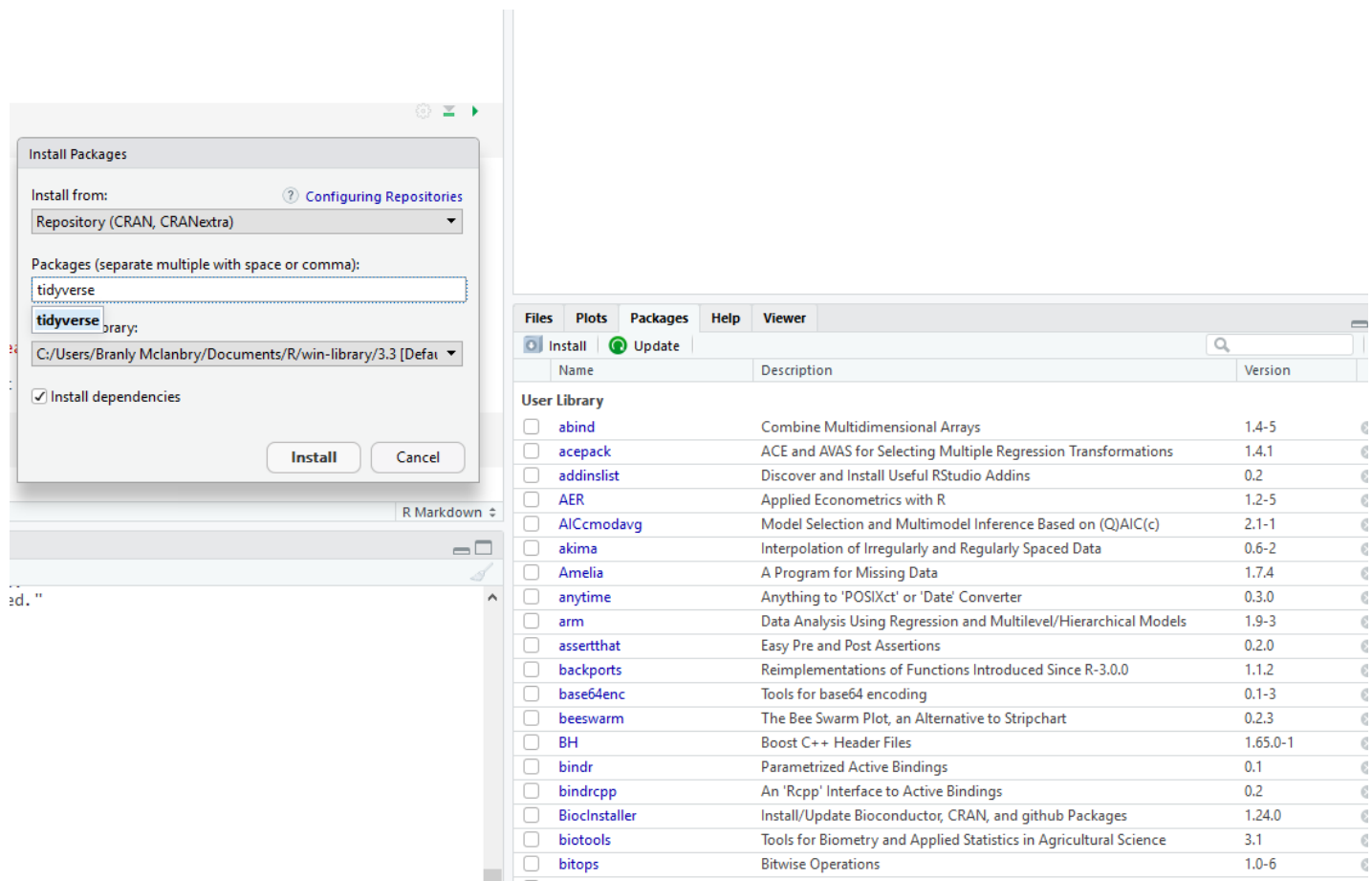


# The first thing we want to do is to install and activate some packages.

Packages are pre-built user submitted functions that makes life easier. An analogy would be like buying furniture from IKEA. They provide you with everything you need to do what you want, you simply need to follow instructions.

You can install it via the point and click install, but most people prefer to install it via the command line. The `tidyverse` package comes from a genius called Hadley Wickham. He's a `r` rockstar, a `r` rockstar because he essentially brainchilded code that makes it easier for people to understand and utilize. If I ever see him in public and I'm going to shake his hand and have him sign my forehead.

```
install.packages("tidyverse")
```



Nice! You can activate the package by using this function.

```
library(tidyverse)
```

From the `tidyverse` package, we want to use functions to create new variables based on previous variables.

The `%>%` functions basically means “and then”.

The `mutate` function tells us to create a new variable based on previous inputs.

So the function below is telling us several things. “Please create a new variable called `niceness` using the information from `Santa.Correlation`.”

# Composite scores.

We can create a composite variable that is a combination of the items together. For example, we want to know the total niceness of a child on christmas night. It's the total of nice items divided by the number of nice items.

```
Santa.Correlation <- Santa.Correlation %>%
  mutate(niceness = (cried + pouted + naughty + nice + bad + good + goodness4)/7)
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

Nice!

Now let's do it for our two scales.

```
Santa.Correlation <- Santa.Correlation %>%
  mutate(niceness = (cried + pouted + naughty + nice + bad + good + goodness4)/7,
         somanygifts = (horns + drums + stockings + gifts)/4)
```

Cool bean! Now we have composite scores.

Cool bean! Now we have composite scores.

Let's just take a look at our data in terms of means or ranges. Use the `describe` function from the `Hmisc` package and it'll provide a small summary of how our data looks. This function is important because it provides overall information during our write up. The function operates as `data$variable`. Essentially it is asking us to select our data, and from that data to choose a variable.

*note* `describe` is a super common function that is terribly titled. We need to specify which function from which package we are using. To do so, we just use the name of the package, and the function within the `::` tells use which function. `package::function`.

```
psych::describe(Santa.Correlation$niceness)
```

```
##      vars  n mean   sd median trimmed  mad  min  max range skew kurtosis
## X1      1 22 3.24 0.75   3.36    3.23 0.85 2.14 4.43  2.29 0.06   -1.32
##      se
## X1 0.16
```

```
psych::describe(Santa.Correlation$somanygifts)
```

```
##      vars  n mean   sd median trimmed  mad  min  max range skew kurtosis  se
## X1      1 26 3.62 1.96   3.5    3.64 2.78   1   6    5    0   -1.69 0.38
```

## Reliability

We want to know about the internal reliability of our items. This concept gets at the idea of “are each items measuring what they are supposed to be measuring?” For example, If I wanted to ask questions of how nice a child was, I would want to ask questions like “how nice are you” or “have you been good?”. Asking a question like “What do you think about the color purple” might not be the best construct to ask for niceness.



We will be using the package `psych` and within that, the function of `alpha` which will provide us with Cronbach's alpha.

We first need to be selecting the items we want to test using the `select` function.

Let's call our overall scale `naughtynicelist` which is selected by nice items from our `data.total`

```
naughtynicelist <- Santa.Correlation %>%  
  select(nice,good,goodness4,cried,pouted,naughty,bad)
```

Super! How about another list for our overall gifts?

```
overallgifts <- Santa.Correlation %>%  
  select(horns, drums, stockings, gifts)
```

Now let's look at the total Cronbach's Alpha, but first we need to load the package `psych`.

```
library(psych)  
alpha(naughtynicelist)
```

```
## Some items ( cried ) were negatively correlated with the total scale and  
## probably should be reversed.  
## To do this, run the function again with the 'check.keys=TRUE' option
```

```
##
## Reliability analysis
## Call: alpha(x = naughtylicelist)
##
##      raw_alpha std.alpha G6(smc) average_r  S/N  ase mean   sd
##      0.43      0.42      1      0.093 0.72 0.14  3.3 0.73
##
## lower alpha upper      95% confidence boundaries
## 0.15 0.43 0.71
##
## Reliability if an item is dropped:
##      raw_alpha std.alpha G6(smc) average_r  S/N alpha se
## nice      -0.020  -0.022   0.85  -0.0036 -0.022  0.260
## good      -0.048   0.041   1.00   0.0070  0.042  0.268
## goodness4  0.048   0.136   0.82   0.0255  0.157  0.239
## cried      0.740   0.712   0.90   0.2920  2.474  0.082
## pouted     0.462   0.428   0.86   0.1108  0.748  0.126
## naughty    0.453   0.415   0.92   0.1057  0.709  0.126
## bad        0.438   0.439   1.00   0.1155  0.784  0.145
##
## Item statistics
##      n raw.r std.r r.cor r.drop mean  sd
## nice    25  0.91  0.91  0.91  0.888  3.2 1.48
## good    26  0.92  0.86  0.86  0.840  3.2 1.67
## goodness4 25  0.87  0.78  0.78  0.673  3.1 1.82
## cried    25 -0.41 -0.43 -0.43 -0.622  3.4 1.66
## pouted   26  0.38  0.39  0.39  0.060  2.8 1.57
## naughty  26  0.40  0.42  0.42  0.080  2.8 1.58
## bad      25  0.18  0.37  0.37  0.069  4.4 0.58
##
## Non missing response frequency for each item
##      1  2  3  4  5 miss
## nice    0.16 0.20 0.16 0.20 0.28 0.04
## good    0.27 0.15 0.04 0.23 0.31 0.00
## goodness4 0.36 0.08 0.08 0.08 0.40 0.04
## cried    0.24 0.12 0.04 0.24 0.36 0.04
## pouted   0.27 0.23 0.12 0.15 0.23 0.00
## naughty  0.31 0.19 0.15 0.12 0.23 0.00
## bad      0.00 0.00 0.04 0.48 0.48 0.04
```

```
alpha(overallgifts)
```

```
##
## Reliability analysis
## Call: alpha(x = overallgifts)
##
##      raw_alpha std.alpha G6(smc) average_r S/N      ase mean sd
##      0.98      0.98      0.99      0.94  60 0.0058  3.6  2
##
## lower alpha upper      95% confidence boundaries
## 0.97 0.98 0.99
##
## Reliability if an item is dropped:
##      raw_alpha std.alpha G6(smc) average_r S/N alpha se
## horns      0.98      0.98      0.98      0.94  44  0.0080
## drums      0.98      0.98      0.98      0.94  46  0.0077
## stockings  0.97      0.97      0.97      0.92  36  0.0100
## gifts      0.98      0.98      0.98      0.95  57  0.0061
##
## Item statistics
##      n raw.r std.r r.cor r.drop mean  sd
## horns   26  0.98  0.98  0.97  0.96  3.7 1.9
## drums   26  0.97  0.97  0.97  0.95  3.6 2.0
## stockings 26  0.99  0.99  0.99  0.98  3.6 2.0
## gifts   26  0.97  0.97  0.96  0.94  3.6 2.1
##
## Non missing response frequency for each item
##      1  2  3  4  5  6 miss
## horns  0.15 0.23 0.12 0.08 0.15 0.27 0
## drums  0.19 0.19 0.15 0.08 0.08 0.31 0
## stockings 0.23 0.15 0.12 0.04 0.19 0.27 0
## gifts  0.23 0.19 0.08 0.08 0.08 0.35 0
```

We will be looking at the raw\_alpha number. What this is telling us is how closely tied together our items are. I received a warning telling me that some items are negatively correlated. That might be the case that I forgot to reverse code them initially. However, in this case it might have been because I randomly selected answers.

An APA write up might look like this.

Participants filled a 7-item Likert type scale which measures the naughty-niceness contingency. Items were adapted from Santa (1547) to fit in with an online population: (1) How nice were you. (2) How naughty are you. (1 = *Strongly Disagree*, 7 = *Strongly Agree*). The items of naughty-niceness were found to have a low internal consistency ( $\alpha = .43$ )

## Correlations

Let's take a look at the correlation between Santa's list and the amount of gifts received. I will be using the `cor.test` function with our two variables. The basic formula for finding correlations is the `data$variable` and the `$` requests the variable from the data.

```
cor.test(Santa.Correlation$niceness,Santa.Correlation$somanygifts)
```

```
##  
## Pearson's product-moment correlation  
##  
## data: Santa.Correlation$niceness and Santa.Correlation$somanygifts  
## t = 4.0086, df = 20, p-value = 0.0006896  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.3421273 0.8499001  
## sample estimates:  
## cor  
## 0.6674589
```

Fantastic!

This output tells us several things!

The df = the number of participants.

P-value = the significance value 95% confidence interval = the range of possible correlations that exists

cor = the correlation coefficient.

In this case, we find a significant correlation between how nice you are and the amount of gifts received.

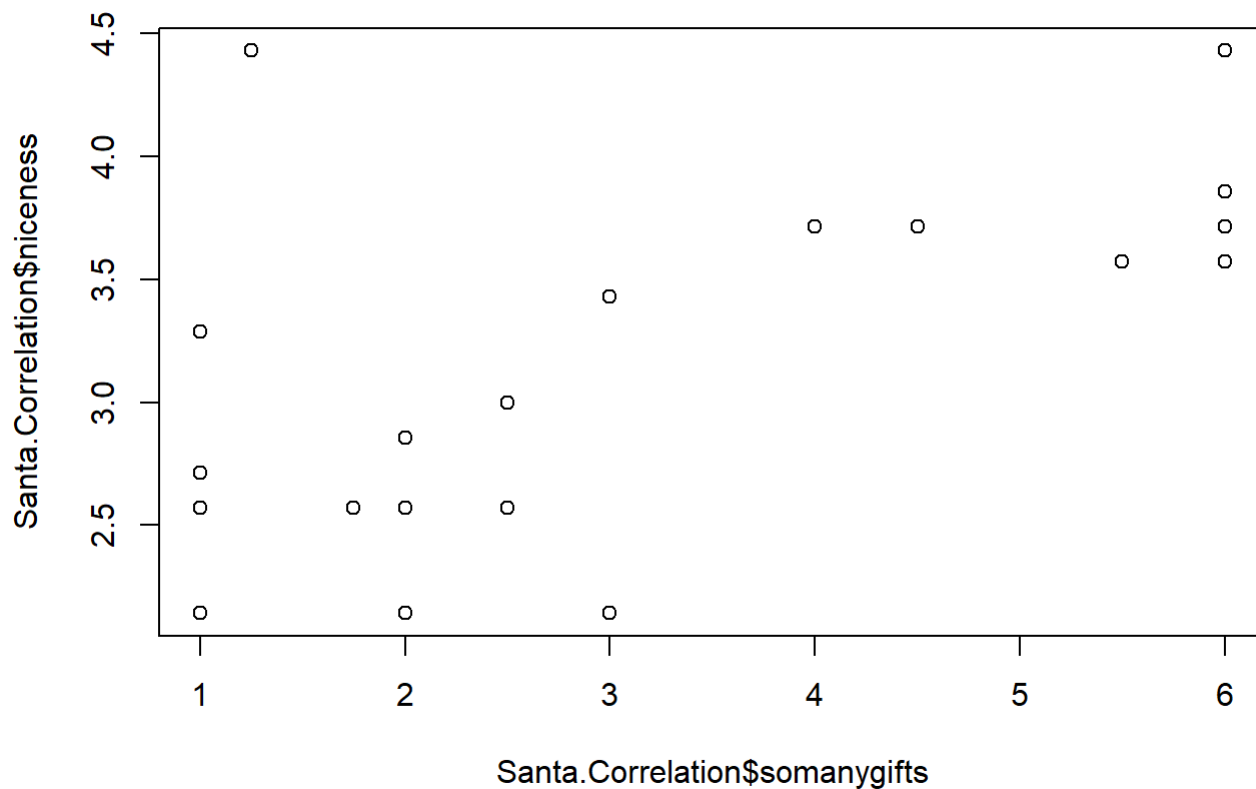
## APA write up.

There is a positive correlation between an increased amount of perceived self-niceness and frequency of gifts received.  $r = .66$ , 95% CI [0.34,0.84],  $p < .001$ . As niceness increases reported amount of gifts also increase.

## Plots

It might be hard to conceptualize the data, so instead of using our thinking brains, let's use our eyeballs. I am using the `plot` function which is a simple plot of data points.

```
plot(Santa.Correlation$somanygifts, Santa.Correlation$niceness)
```



## Regression

So now that we find a correlation between the two. Let's ask the question: Does niceness predict overall gifts received?

We do so with the `lm` function which stands for linear model. The format is as follows.

`independent variable ~ dependent variable, data`. Let's try it in action. In this particular function the `~` means "predicted by"

```
mymodel <- lm(somanygifts~niceness,Santa.Correlation)
```

sweeeeet let's find out more about our model by running a `summary` call on it.

```
summary(mymodel)
```

```
##
## Call:
## lm(formula = somanygifts ~ niceness, data = Santa.Correlation)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.2378 -0.6176  0.0605  1.2998  2.0444
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.4284     1.4814  -1.639  0.11679
## niceness      1.7875     0.4459   4.009  0.00069 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.533 on 20 degrees of freedom
## (4 observations deleted due to missingness)
## Multiple R-squared:  0.4455, Adjusted R-squared:  0.4178
## F-statistic: 16.07 on 1 and 20 DF,  p-value: 0.0006896
```

*note* important things to notice are the estimate and significance value. Essentially (this is the APA write up), we find that niceness significantly predicts gifts.  $r^2 = .64$ ,  $F = 19.97$ ,  $p < .001$ ,  $b = .84$ . As people are nicer, we expect them to receive more gifts.

Let's try that plot again, but this time add a `abline` on it. a `abline` is simply a straight line through the plot. "AB" stands for the intercept and the slope.

```
plot(Santa.Correlation$somanygifts, Santa.Correlation$niceness)
abline(lm(somanygifts~niceness,Santa.Correlation))
```

