Personality Psychology Lab Handbook

Barnard College
Department of Psychology
PSYC BC2124

Fall 2023

Table of contents

Syllabus

> Human personalities are rather like fractals. It is not just that what we do in the large-scale narratives of our lives—love, career, friendships—tends to be somewhat consistent over time, with us often repeating the same kinds of triumph or mistakes. Rather, what we do in tiny interactions like the way we shop, dress or talk to a stranger on a train or decorate our houses, shows the same kinds of patterns as can be observed from examining a whole life.
>
> – Daniel Nettle, Personality: What Makes You the Way You Are

Pre-requisites: PSYC BC1001 Introduction to Psychology; PSYC BC1101 Statistics; PSYC BC 1024 Research Methods
Co-requisite: PSYC BC2125 Personality Psychology

Time, venue & instructor

Section 001: Monday 10:10-1PM, Milbank 410
Section 002: Monday 1:10-4:00PM, Milbank 410

Instructor: Dr. Rob Brotherton (rbrother@barnard.edu)
Office hour: TBC

Course overview

This lab will usually be taken concurrently with BC2125 Personality lecture. It will expand upon some of the theoretical, methodological and analytic issues introduced there, as well as giving you the opportunity to explore topics of your choosing from within (or beyond) those covered in the lectures in greater depth though hands-on experience with research design and data analysis.

The semester is broken into 3 projects. The first will involve planning and executing a correlational analysis of existing data. The second will involve performing a multiple regression analysis on existing data. The third project will involve designing and pilot-testing a novel scale to measure an aspect of personality. Through these projects you will gain experience in formulating psychological research questions pertaining to personality; collecting, analyzing and visualizing data; and interpreting and communicating your findings. Projects will be undertaken in groups; group members will collaborate on design and analysis. Each project will culminate in an in-class group presentation and submission of a brief individual write up of your project.

Class format & participation

Labs are substantially more interactive and discussion-based than the traditional lecture format, and depend on everyone's active participation in class discussions and activity as well as group work focused around the projects. Your active participation across the semester will therefore contribute a substantial portion of your grade.

If you have questions, thoughts, or ideas you want to share, feel free to do so at any time (while keeping within the bounds of polite conversation, obviously–don't interrupt or talk over other people! But do feel free to respond to others without having to raise your hand or wait to be called on). Everyone will get the most out of this lab when the discussion can develop organically and everyone feels free to be part of the conversation if & when they have something to add.

Being part of the in-person discussion is one obvious way to participate, but it's not the only way. Different people have different styles of participation, and the lab is designed to try and accommodate and encourage different approaches. Your level of engagement with your project partner(s), your TA and Prof. Brotherton as you work through your projects is also an important form of participation. You can also participate by coming to office hours.

At a minimum (i.e. for a passing grade), I'll be looking for some form of participation (loosely defined) from you every week. Higher participation grades will be earned through regular, enthusiastic, productive participation (note that quality is more important than quantity).

Workload

As a general rule for the amount of time students should expect to commit to classes, the college suggests three hours per week in or outside of class per credit. Since this class is worth 1.5 credits, that corresponds to 4.5 hours per week, split between time in the classroom and time spent completing the associated assignments.

Final Grades

Your numeric score for the course is a product of your scores for each assignment, weighted as follows:

	Weight (%)
Participation (over the course of the semester)	10
Presentations	30
Reports	60

Final grades are determined according to the following boundaries:

```
Letter grade:  A+ A  A- B+ B  B- C+ C  C- D  F
Numeric score: 97 93 90 87 83 80 77 73 70 60 <60
```

Course policies

Attendance & timeliness

In-person attendance of every lab session is expected, and you should expect to stay for the full duration of the lab. Normally it is departmental policy to remove students who miss more than two lab sessions from the course; however, given ongoing revisions to college-wide health-related policies, exceptions may be made. If you are feeling unwell, you should not come to class and notify me of nonattendance before class if possible.

When you are attending, please arrive on time for class. Frequent lateness will impact your participation grade.

Assignment deadlines & late policy

Assignments are listed in the class schedule next to the class in which details about completing the assignment will be provided. The assignment must be completed and submitted before the following class, i.e. requirements for the first presentation slide will be explained in class on 10/2, and the slide must be submitted before the next class on 10/9.

For the written project reports, a grade penalty of 5 points will be applied for each day (or part thereof) that an assignment is late (up to a maximum of 6 days; work not submitted before the next lab will receive a score of zero). For example, if your work is A+ quality but is submitted a day-and-a-half late, you will only receive a B+. This policy is intended to incentivize timely submission while easing the stress of genuine emergencies. When things come up that prevent timely submission you can prioritize accordingly, knowing that a small penalty on one assignment for this lab will not tank your final grade.

Late submission for the presentation slides will not be possible; failure to submit a link to your slides in advance of the presentation will obviously limit your presentation grade.

Academic integrity

Students are expected to follow the Barnard Honor Code, available at https://barnard.edu/honor-code.

Note that while you will collaborate with group members on the design, analysis, and presentation of research projects, you may not collaborate on the written report: each group member must write their own individual reports.

Academic accommodations and general wellness

It is always important to recognize the different pressures, burdens, and stressors you may be facing, whether personal, emotional, physical, financial, mental, or academic. The faculty and administration recognize this, and are prepared to provide assistance to students in need. I encourage you to seek advice from your advisor, Dean, the Center for Accessibility Resources & Disability Services (CARDS), or Barnard Health & Wellness as needed. Please let me know of any issues you wish to share with me that you feel are impacting your ability to complete the course to the best of your ability. Though it isn't always easy, it is better to proactively seek help rather than letting problems build up.

Class schedule

Date	Topic	Assignment
9/11	Course overview	
Project 1: Correlation		
9/18	Project planning	
9/25	Data cleaning	
10/2	Analysis	Presentation slide
10/9	Presentations	Project 1 report
Project 2: Multiple regression		
10/16	Project planning	
10/23	Data cleaning & analysis	
10/30	Visualization & reporting	Presentation slide
11/6	Presentations	Project 2 report
Project 3: Scale design		
11/13	Scale planning	
11/20	Questionnaire design	
11/27	*No class (Thanksgiving)*	
12/4	Analysis & reporting	Presentation slide
12/11	Presentations	Project 3 report

* Assignments due by the following class.

Project 1: Correlation

Lab 2: Project Planning

In this session we will begin the first project of the course: performing a correlational analysis using the ANES 2016 dataset. By the end of the session you will have a plan for your analysis.

Goals

- Examine the ANES 2016 dataset
- Identify variables for your correlation analysis
- Search the literature to find relevant research
- Formulate a brief research proposal

Project overview

A *correlation* refers an association between two things. It is a statement of a statistical relationship–a general tendency, rather than a rigid law. To say that some aspect of personality is correlated with something else–for example, neuroticism is correlated with lower wellbeing or openness is correlated with greater cognitive ability–is to say that those things *tend* to go together. Not everyone who scores high on neuroticism will have lower wellbeing than anyone low on neuroticism, but there is some tendency for the two to go together on the whole.

Of course, these kind of correlations aren't just facts found lying around in nature; they are empirical findings produced by researchers. All the findings you learn about in the personality psychology lecture (and beyond) are the product of research procedures. Researchers decide what psychological constructs they want to investigate; how to operationally define those constructs; what statistical analyses are appropriate; and what conclusions may be drawn.

With this project, you will examine a correlation between a personality trait and some other construct of your choosing by analyzing existing data.

Step 1. Examine the data

The dataset we will use is from the American National Election Studies (ANES), academic surveys of voters in the United States conducted before and after every presidential election, going back to the 1940s. Specifically, for this project we will use data collected around the 2016 election. The reason for using this (instead of more recent data) is that the 2016 survey included a personality scale: the Ten-item Personality Inventory (TIPI: Gosling et al., 2003). This scale is a short measure of the Big 5 personality traits of Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism. For your project, you will pick one of these traits and investigate its correlation with one of the other constructs recorded in the survey.

To see what variables were recorded, you will skim the Codebook. This is a pdf document that details the survey methodology exhaustively, including listing every question that was asked. It's not fun reading, but it is your guide to picking out .

To give you an idea of what you will find: many of the variables have to do with people's political preferences and actions (voting preferences, approval of various political actors, etc).

- Demographic variables (age, race, gender, marital status, education, income, etc)
- Knowledge/intelligence (WORDSUM: a vocab test that is a common proxy for IQ; questions about political general knowledge)
- Religiosity
- Security/anxiety about various things… risks of vaccines, financial security,
- Affect - feeling thermometers
- Life satisfaction
- Conspiracy beliefs (is Barack Obama Muslim? Did the U.S. government know about 9/11 in advance?)
- Opinionatedness (R forms opinions about everything; Important for R to hold strong opinions; It bothers R to remain neutral)
- Stereotypes, sexism
- Authoritarianism (child-rearing style), traditionalism, conservatism, social trust
- Equalitarianism
- Patriotism
- Media exposure and attention

Step 2. Find relevant research

Real research doesn't happen in a vacuum; research plans and expectations should be informed by what has come before. Therefore once you have an idea of what variables you would like to analyze, you will search the literature to see what other researchers have found of these (or related) personality traits.

What to look for

A published, scholarly journal article detailing an empirical finding relevant to your variables of interest. This might be a paper reporting one or several individual studies that the researchers conducted, or it may be a review paper or meta-analysis.[1]

Where to look

* Google Scholar

Google Scholar searches the full text of scholarly articles. It casts a wide net, searching across all disciplines, and including books and other materials in addition to journal articles, so will likely find many articles not very relevant to the topic as well as those that are relevant.

* APA PsycINFO

The link above should take you to PsycINFO, a database for scholarly psychology research (you can also serch for `psycinfo` in a CLIO quicksearch). PsycInfo gives you the ability to do much more focused searching than Google Scholar.

- You can do sequential searches and then combine them with the Boolean operators `AND`, `OR`, and `NOT` by selecting them from the dropdown boxes.
- You can select where your keywords should appear, i.e. in the title, abstract, or full text of articles. Selecting `Word in Major Subject Heading` can help narrow down your search to articles that are actually on the topic you're interested in (rather than just containing the keyword).
- The *APA Thesaurus of Psychological Index Terms* link at the top can be useful for finding technical terms used in psychology, and other related ideas.

Step 3. Articulate your hypothesis

Having chosen your variables and found some relevant research to inform your theoretical perspective about how the variables are (or aren't) associated, you should be able to articulate your *hypothesis*. This is a formal statement of your expectations about how the variables are associated, and it will be tested quantitatively by calculating a correlation statistic.

[1] A meta-analysis pools the findings of many individual studies by different researchers into a single analysis.

Your proposal

At the start of next week's class, each project team will give a short, informal presentation of their research proposal. This should outline:

- Your variables of interest (one should be one of the Big 5, the other any variable of your choosing)
- Your theoretical perspective (based on the research you found)
- Your expectations (this should follow from your theoretical perspective)

Lab 3: Data cleaning

You should begin this session with variables from the ANES 2016 dataset in mind for your analysis. In class we will introduce the R language and RStudio environment, and perform necessary data cleaning and manipulation in preparation for analysis.

Goals

- Get your R environment set up
- Read the data you need into R
- Select required variables
- Filter the data based on completeness (and any other criteria)
- Compute any required variables (scale means, number of items missing, etc)

Working with data in R

Getting R ready

In addition to containing a Big 5 personality scale, the ANES 2016 dataset is convenient for our purposes because someone went to the trouble of creating an R package which makes working with the ANES data relatively straightforward (not that you won't still run into issues!): `anesr` (github.com/jamesmartherus/anesr).

To start exploring the data in R, you first need to set up your environment. This means installing the `anesr` package from github. Since the package is hosted on GitHub (as opposed to the official R repository of packages), the easiest way to install it is by first installing the `devtools` package, which has a function for installing other packages from GitHub.

```
install.packages("devtools")

devtools::install_github("jamesmartherus/anesr")
```

We will also use some other packages for data wrangling and analysis. Developers have created a collection of packages for R called the `tidyverse` to make coding these common tasks easier. The `tidyverse` can be installed like so:

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

If you execute those lines of code the packages will be installed on your system. That step only needs to be done once, but you need to 'activate' the packages using `library()` to make their functions and data available each time to start a new R session.

```r
library(anesr)
library(tidyverse)
```

Getting data into R

```r
data(timeseries_2016)
```

When you run that line of code you won't see any output, but you should see the name `timeseries_2016` appear in your Environment pane. That is now an object in R called a data.frame. You can think of it as a spreadsheet like you're familiar with from Excel or Google Sheets; a set of columns, one for each variable in the dataset, and a row for each participant's answers.

Typing the name of the data.frame and running that line of code will show the first few columns and rows.

```r
timeseries_2016
```

```
# A tibble: 4,270 x 1,842
   version       V160001 V160001_orig V160101 V160101f V160101w V160102 V160102f
   <chr>           <dbl>        <dbl>   <dbl>    <dbl>    <dbl>   <dbl>    <dbl>
 1 ANES2016Time~       1       300001   0.827    0.888        0   0.842    0.927
 2 ANES2016Time~       2       300002   1.08     1.16         0   1.01     1.08
 3 ANES2016Time~       3       300003   0.388    0.416        0   0.367    0.398
 4 ANES2016Time~       4       300004   0.360    0.385        0   0.366    0.418
 5 ANES2016Time~       5       300006   0.647    0.693        0   0.646    0.726
 6 ANES2016Time~       6       300007   0.706    0.759        0   0.688    0.725
 7 ANES2016Time~       7       300008   3.96     4.25         0   4.62     4.79
 8 ANES2016Time~       8       300012   0.962    1.03         0   0.943    1.04
 9 ANES2016Time~       9       300018   0.976    1.05         0   1.01     1.07
10 ANES2016Time~      10       300020   0.618    0.664        0   0.600    0.638
# i 4,260 more rows
# i 1,834 more variables: V160102w <dbl>, V160201 <dbl>, V160201f <dbl>,
#   V160201w <dbl>, V160202 <dbl>, V160202f <dbl>, V160202w <dbl>,
```

```
#   V160501 <hvn_lbll>, V160502 <hvn_lbll>, V161001 <hvn_lbll>,
#   V161002 <hvn_lbll>, V161003 <hvn_lbll>, V161004 <hvn_lbll>,
#   V161005 <hvn_lbll>, V161006 <hvn_lbll>, V161007 <hvn_lbll>,
#   V161008 <hvn_lbll>, V161009 <hvn_lbll>, V161010a <hvn_lbll>, ...
```

You can also click on the name in the Environment pane to see the data like a spreadsheet in a new tab.

Select your variables

As you can see, the data.frame contains a *lot* of variables; there are 1,842 columns of data. You'll only need a few of those. So the first step is selecting just the variables you need to work with.

There are a lot of ways to do this. The simplest would be to make a note of the variable IDs from the codebook and use them within `dplyr`'s `select()` function. This allows us to simply type in variable names separated by commas. You can also give the columns new names when selecting,

For this example I'll look at the correlation between extraversion and the 'feeling thermometer' for the Democratic Party. Extraversion has two TIPI items; their IDs (from the codebook) are `V162333` and `V162338`. The ID for the Democratic Party feeling thermometer is `V161095`.

```r
my_data <- timeseries_2016 |>
  select(extraversion1 = "V162333",
         extraversion2 = "V162338", ,
         feeling_thermometer = "V161095")
```

Let's see what this new data.frame looks like:

```r
my_data
```

```
# A tibble: 4,270 x 3
   extraversion1 extraversion2 feeling_thermometer
   <hvn_lbll>    <hvn_lbll>    <hvn_lbll>
 1 6             5              0
 2 6             6             15
 3 6             2             50
 4 6             4             30
 5 5             7             70
 6 5             6             15
 7 5             1             85
```

```
 8 6              3               0
 9 7              1              15
10 5              5              50
# i 4,260 more rows
```

It all looks good so far. But if you inspect the data more extensively (click the name in your Environment and scroll down a bit) you'll notice that there are some negative numbers in the data. That's from survey codes which record missing data. If you try to calculate an average score with those included it'll mess up the sums, so we need to do some data cleaning to handle things like that.

Cleaning the data

There are a lot of different ways we could handle this. One way is to `filter()` the data, retaining only rows which meet certain conditions.

```
my_data_complete <- my_data |>
  filter(if_all(contains("extraversion"), ~ . %in% 1:7))  |>
  filter(feeling_thermometer >= 0)
```

Notice how many rows have been removed:

```
nrow(my_data)
```

```
[1] 4270
```

```
nrow(my_data_complete)
```

```
[1] 3540
```

Another way would be to `mutate()` the data; that is, change certain values based on some condition.

```
# my_data_complete <- my_data |>
#     mutate(across(contains("extraversion"), ~case_when(
#         . < 1 ~ NA,
#         TRUE ~ .
#     ))) |>
#     mutate(feeling_thermometer = case_when(feeling_thermometer < 0 ~ NA, TRUE ~ feeling_
```

Computing scale averages

Now that the data is subsetted and the missing/invalid responses are taken care of, the last thing to do is compute any new values required for analysis. As an example, if you have a scale which has multiple questions asking about a particular construct, it is often necessary to compute an average score for each participant.

The TIPI has 10 questions in total, two for each of the Big 5 personality traits. So it may be desirable to compute a mean trait score by averaging its two respective items.

Notice, however, that for each of the 5 traits, one question is positively worded and one is negatively worded. For extraversion, item `V162333` is "extraverted, enthusiastic", while item `V162338` is "reserved, quiet". The second one needs to be reverse-coded, so that higher scores on both items indicate greater extraversion. Since answers can range from 1 to 7, an easy way to recode the scores is to subtract the participant's response from 8; 1 becomes 7, 2 becomes 6, etc.

```
my_data_complete <- my_data_complete |>
    mutate(extraversion2 = 8 - extraversion2)
```

Now we can go ahead and compute the average, using `mutate()` to create a new column containing the `rowMeans()` (i.e. an average for each row) for the specified columns.

```
my_data_complete <- my_data_complete |>
    mutate(extraversion_mean = rowMeans(across(contains("extraversion"))))
```

Lab 4: Analysis

You will start this session with your cleaned data ready to use in R. By the end of the session you will have computed the correlation statistic, produced some visualizations of your data, and be ready to present and write up your findings.

Goals

- Describe and visualize your variables
- Understand what the correlation statistic quantifies
- Perform the appropriate correlational analysis on your data
- Interpret the results

Analyzing data in R

Running with my example from last week, my variables were avearge extraversion scores and the Democratic Party feeling thermometer score. I made a data.frame with just those variables, filtered the data down to complete, valid responses, recoded the negatively-worded item, and computed an extraversion mean score. To refresh your memory, here's the entire pipeline from start to finish:

```
library(tidyverse)
library(anesr)
data(timeseries_2016)

my_data_complete <- timeseries_2016 |>
  select(extraversion1 = "V162333",
         extraversion2 = "V162338", ,
         feeling_thermometer = "V161095") |>
  filter(if_all(contains("extraversion"), ~ . %in% 1:7))  |>
  filter(feeling_thermometer >= 0) |>
  mutate(extraversion2 = 8 - extraversion2,
         extraversion_mean = rowMeans(across(contains("extraversion"))))
```

Describing your data

```
my_data_complete |>
  mutate(across(everything(), as.numeric)) |>
  pivot_longer(everything()) |>
  summarize(n = n(),
            mean = mean(value),
            sd = sd(value),
            .by = name)
```

```
# A tibble: 4 x 4
  name                    n  mean    sd
  <chr>               <int> <dbl> <dbl>
1 extraversion1        3540  4.79  1.58
2 extraversion2        3540  3.65  1.76
3 feeling_thermometer  3540 48.3  30.0
4 extraversion_mean    3540  4.22  1.38
```

Visualizing the data

As with most things, there are a lot of different ways of producing graphs using R. One of the most widely used and powerful is the `ggplot2` package.[2] The name refers to the idea of the "grammar of graphics", and it is built around a layering approach. You first specify your data and aesthetics (what should data will go on the x and y axes), then geometry (do you want data to be represented by points or bars or as a histogram?), any scaling (e.g. what values should be labeled on each axis), and theme elements (how do you want the plot to look generally?). There can be a lot of complexity, but building things up layer by layer, gradually adding and refining elements, is a powerful and satisfying approach.

```
my_data_complete |>
  ggplot(aes(x = extraversion1)) +
  geom_histogram(binwidth = 1)
```

```
Don't know how to automatically pick scale for object of type <haven_labelled>.
Defaulting to continuous.
```

[2]The `ggplot2` package is part of the `tidyverse`, so because we already ran `library(tidyverse)` earlier the `ggplot2` functions are already available to us. If you needed to, you could always run `library(ggplot2)` to activate it separately.

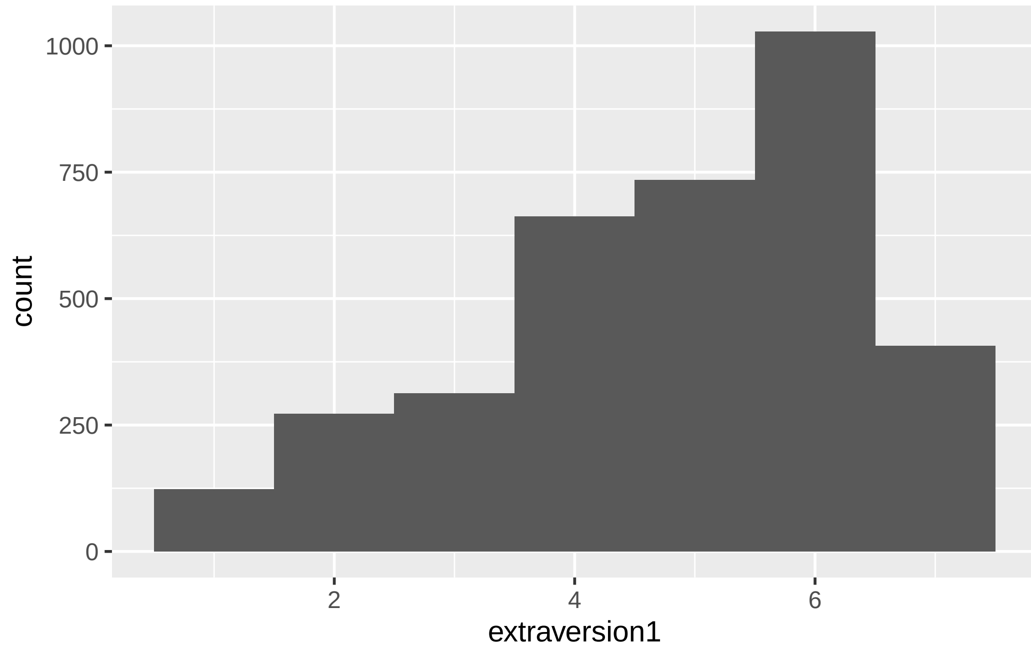

Figure 1: Histogram of responses to "extraverted, enthusiastic" TIPI item

The default theme is perfectly serviceable, but you can customize every element. Here I'll specify a couple of aspects using the `theme()` function, and I'll assign it to the name `theme_apa`. Then I can always add `theme_apa` as a layer to my plots going forward.

```
theme_apa <- theme(
  panel.background = element_blank(),
  axis.line = element_line()
)
```

```
my_data_complete |>
  ggplot(aes(x = extraversion1)) +
  geom_histogram(binwidth = 1, color = "white") +
  scale_x_continuous(breaks = 1:7) +
  theme_apa
```

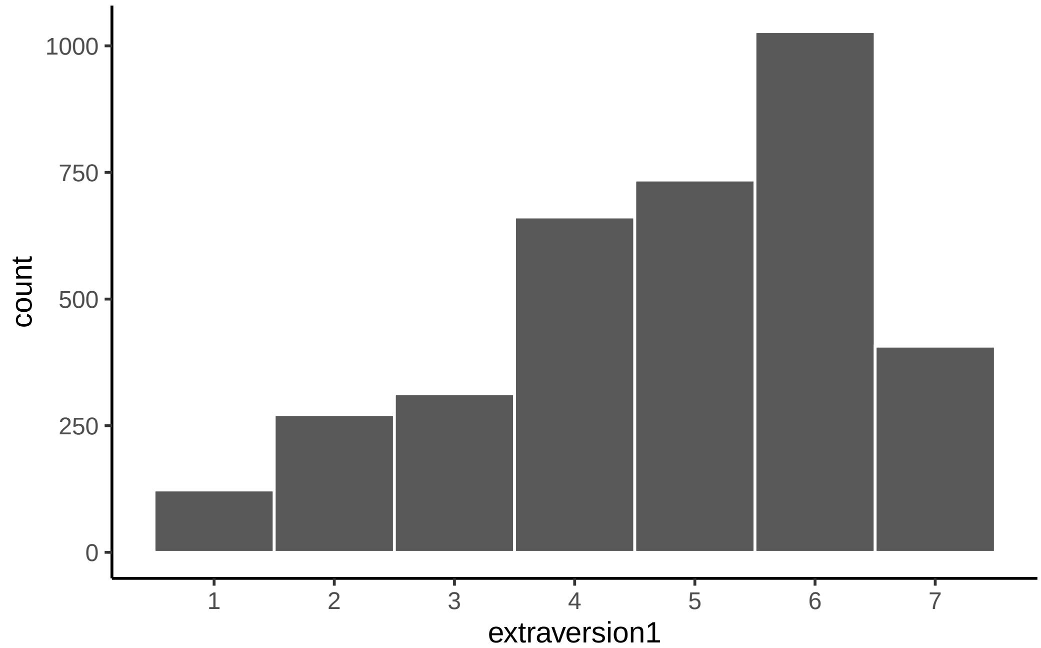

Figure 2: Histogram of responses to “extraverted, enthusiastic” TIPI item

```
my_data_complete |>
  ggplot(aes(x = 8 - extraversion2)) +
  geom_histogram(binwidth = 1, color = "white") +
  scale_x_continuous(breaks = 1:7) +
  theme_apa
```

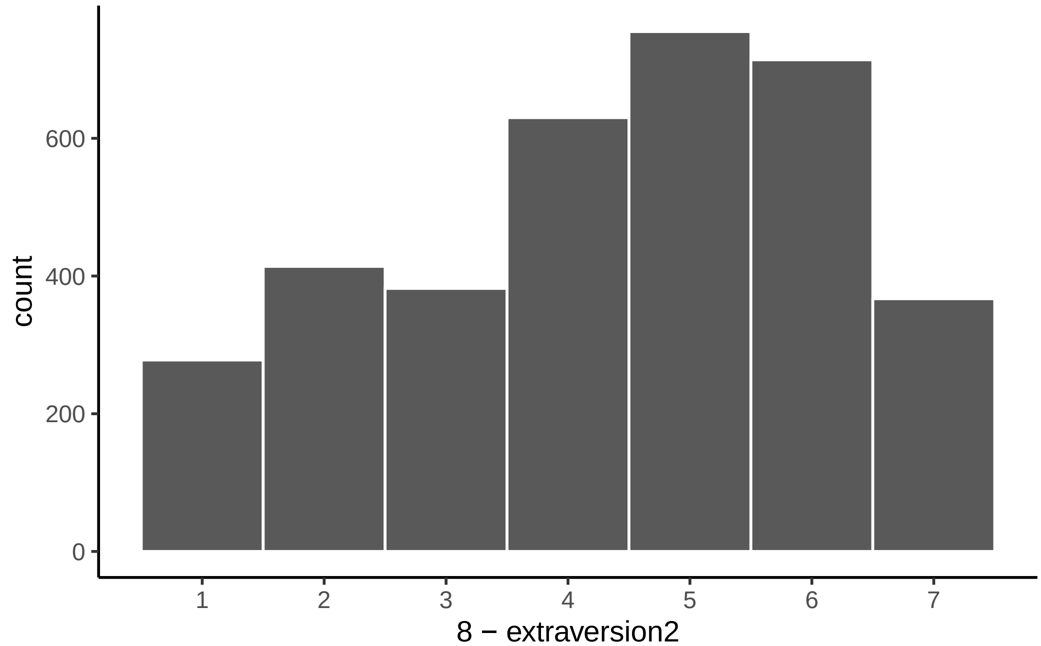

Figure 3: Histogram of responses to "reserved, quiet" TIPI item

```
my_data_complete |>
  ggplot(aes(x = extraversion_mean)) +
  geom_histogram(binwidth = 0.5, color = "white") +
  scale_x_continuous(breaks = 1:7) +
  theme_apa
```

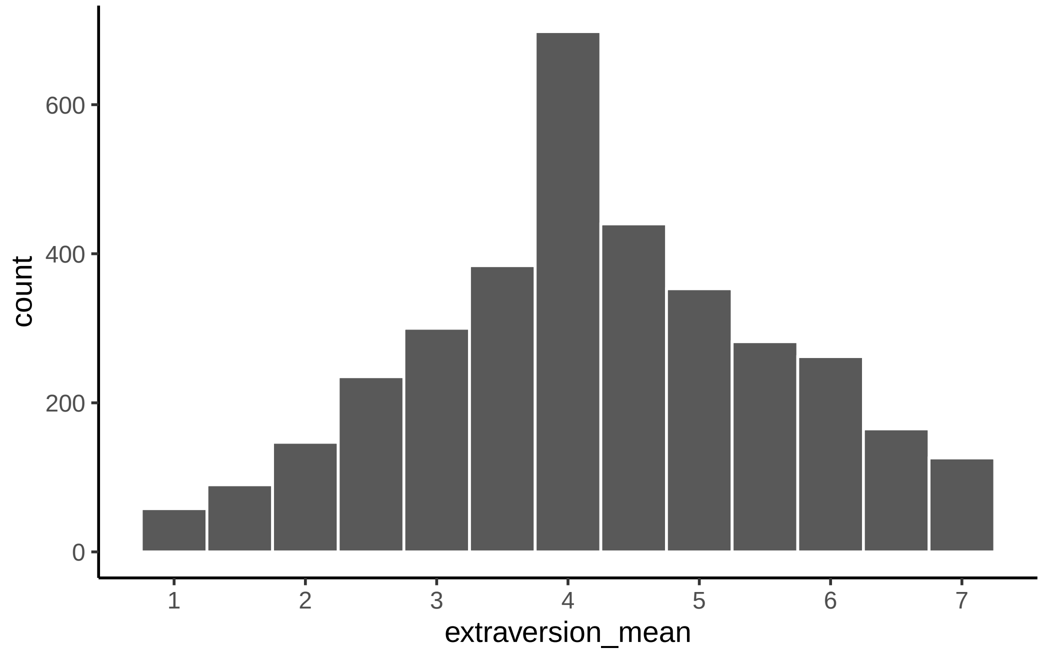

Figure 4: Histogram of average scores on TIPI Extraversion subscale

```
my_data_complete |>
  ggplot(aes(x = feeling_thermometer)) +
  geom_histogram(binwidth = 1, color = "white") +
  scale_x_continuous(breaks = seq(from = 0, to = 100, by = 10)) +
  theme_apa
```

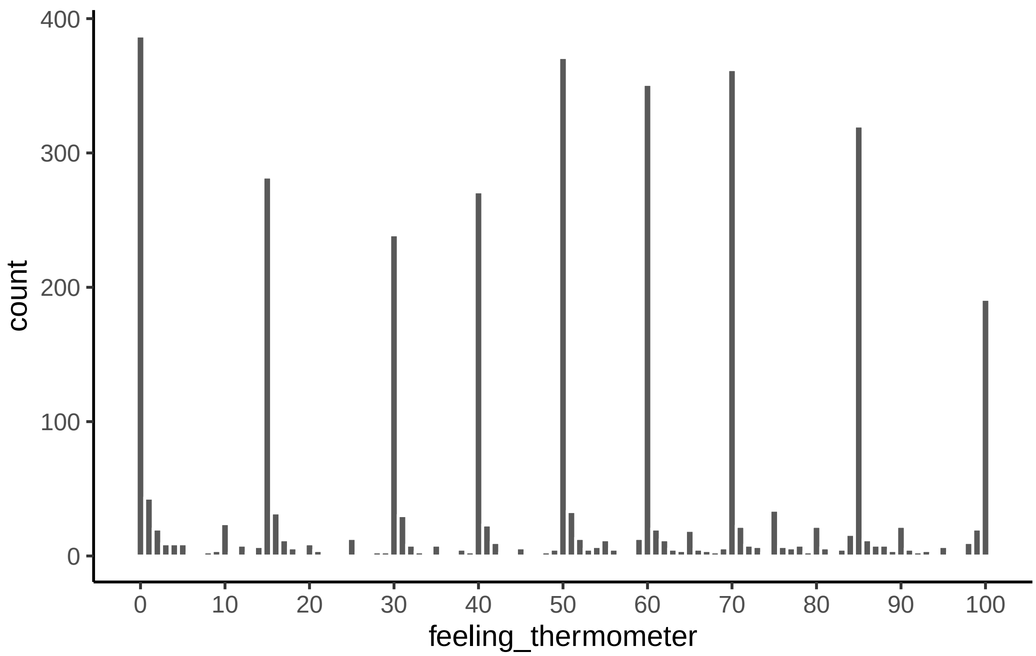

Figure 5: Histogram of responses to Democratic Party feeling thermometer

Correlation analysis

The correlation statistic

The correlation statistic can be computed with a single line of code, as you'll see. But it's important to understand the math happening behind the scenes.

Correlation in R

The data is ready to be analyzed. The correlation between two variables can be found using the `cor()` function.

```
cor(x = my_data_complete$extraversion_mean,
    y = my_data_complete$feeling_thermometer)
```

```
[1] -0.01012898
```

If you got an answer of `NA` instead of a number, it is probably because your data has some missing data. You just need to tell `cor()` to only use data for which both pairs of values are nonmissing:

```
cor(x = my_data_complete$extraversion_mean,
    y = my_data_complete$feeling_thermometer,
    use = "pairwise.complete.obs")
```

```
[1] -0.01012898
```

```
my_data_complete |>
  ggplot(aes(x = extraversion_mean, y = feeling_thermometer)) +
  geom_point(position = position_jitter(width = 0.4, height = 0),
             alpha = 0.1) +
  scale_x_continuous(breaks = 1:7) +
  scale_y_continuous(breaks = seq(from = 0, to = 100, by = 10)) +
  theme_apa
```

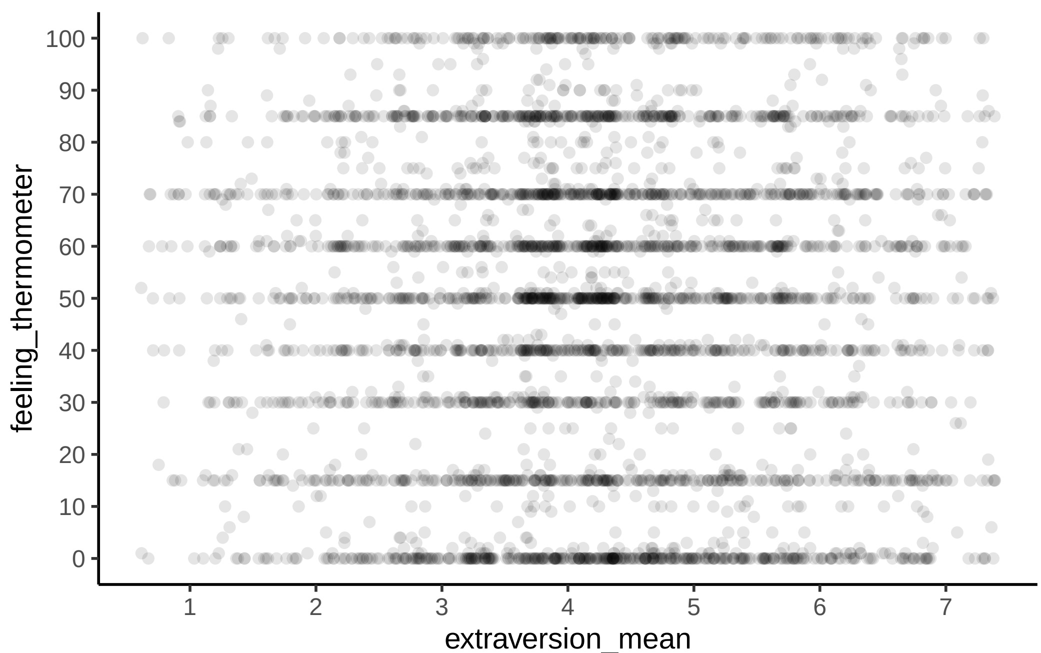

Figure 6: Scatterplot (with jitter) of average extraversion scores and feeling thermometer scores

Lab 5: Presentations

This week each project team will present their proposed project to the rest of the class.

Presentation

Guide to presenting

Each team will give a two-minute presentation, accompanied by a single PowerPoint slide, which should encapsulate the motivation, methods, anticipated findings, and interpretation of your proposed project. Two minutes is not a lot of time. Apparently, people speak at a rate of around 125 to 150 words per minute on average. So a 2-minute presentation will be no more than around 300 words. The single-slide, not-many-words format demands clarity, conciseness, and being bold to spark the audience's interest in your topic.

Avoid simply reading excerpts from your paper. That would be boring, and would probably take up too many words. Make it fun and interesting. Try to grab the audience's attention and hit them with just the most important points of your ideas.

You will also have a single PowerPoint slide to accompany your presentation. Make it count. You can't just cram a load of text on there, because nobody will be able to read it. Plus, it'd distract from what you're saying. Make it a visual aid that somehow supports or clarifies what you're saying. It might be a visual representation of your design, a key piece of your experimental stimuli, a graph of your expected results, or just a pertinent meme which conveys the motivation for your question.

After your two-minute talk you'll take a few questions from the audience, and your responsiveness will contribute toward your grade as well as the quality of your presentation itself (remember a perfectly acceptable answer if often: "Good question; I don't know the answer!"). It's not usually an issue, but just in case your audience is left speechless, I suggest coming with a couple of questions or thoughts of your own that you can throw at the audience to spark more questions.

Each pair will be allotted five minutes total for their talk and Q&A. Going over time and/or failing to leave time for questions will impact your presentation grade. It is up to each pair to decide how to divide up the two-minute talk, and to practice to make sure the presentation is to time.

Guide to watching presentations

As an audience member, you are still being graded for class participation this week. That means giving everyone else's presentation the attention and enthusiasm it deserves, and rewarding their hard work with questions. (Going to the trouble of putting together a presentation only for nobody to have anything to say about it is not a good feeling.)

Good questions to ask are things like "Could you clarify X", "Had you considered Y", or "How might this relate to Z." One reason for presenting your project is to hopefully get some useful feedback from the audience with which to refine your final paper, so try to give the kind of feedback you hope to receive.

Written report

You will produce a miniature research paper reporting your project. Note that each team member will produce their own individual report; even though the project has been collaborative, your write up will be your own.

Your report should consist of the following sections:

- Introduction (two or three paragraphs, including summary of relevant research and hypothesis)
- Method (a description of the variables you selected, the number of valid responses, and any other information about the procedures that generated the data that you think necessary to report)
- Results (all the descriptive statistics, figures, and statistics you produced)
- Discussion (a paragraph or two interpreting your results and drawing conclusions)

Project 2: Multiple Regression

Lab 6: Project planning

In this lab, you will start your second project: conducting a multiple regression analysis on ANES data.

Goals

- Understand the idea of multiple regression
- Identify variables for your analysis
- Search the literature to find relevant research
- Formulate a brief research proposal

Lab 7: Data cleaning & analysis

Goals

- Control conditions
- Factorial designs

```
lm(dv ~ iv1 * iv2)
```

Lab 8: Visualization & reporting

Goals

- Style and contents of an APA Introduction section
- How to frame previous findings to create a coherent and compelling justification
- APA style for citations & references

Lab 9: Presentations

Present in class.

Write up due by next class.

Project 3: Scale design

Lab 10: Scale planning

Goals

- Understand the process of scale design and validation
- Pick a personality trait to design a scale around
- Write a brief proposal of your project and expectations

Lab 11: Questionnaire design

Goals

- Generate items
- Consider face validity
- Decide on reponse options
- Create your survey using Google Forms

Lab 12: Data collection and analysis

Lab 13: Presentations

Present in class.

Write up due by next class.

Getting started with R

posit.cloud

You will use posit.cloud to work with data in R.

Let's do something cool

Once you have a posit.cloud account, click this link.

Wait, what are you talking about?

There are a few different names involved here, so to try and clear things up:

- **R** is a coding language
- **RStudio** is a software interface for using R
- **Posit** is the name of the company that makes RStudio
- **posit.cloud** provides a way of using RStudio in your web browser

You can install R and RStudio on your own computer for free and do things that way, but using the cloud-based RStudio via posit.cloud simplifies things immesnely.

Fundamentals of R for data analysis

R is a programming language well-suited to interactive data exploration and analysis. It might seem daunting if you've have no experience with coding, but the basic idea is that you have some data, like you are familiar with from a regular Excel or Google Sheets spreadsheet, and you perform operations on your data using functions a lot like you would in Excel/Sheets. For example, you might compute an average in Sheets by typing `=AVERAGE(A1:A10)`. In R you might type `mean(my_data$column_a)`. The specifics of the function names are different, but the basic idea is the same.

There are two other ideas that will help you get started coding in R.

Assignment

The first is the assignment operator: `<-`. You assign things to a name by typing something like:

```
name <- thing
```

The `thing` there might be a set of numbers, an entire dataset, or something else. Giving it a name allows to you perform subsequent operations more easily, and choosing appropriate names makes your code easier to understand.

```
original_numbers <- 1:10
original_numbers
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

```
doubled_numbers <- original_numbers * 2
doubled_numbers
```

```
 [1]  2  4  6  8 10 12 14 16 18 20
```

Functions

Almost everything happens inside functions.

```
mean(original_numbers)
```

```
[1] 5.5
```

```
mean(doubled_numbers)
```

```
[1] 11
```

Piping

The second is the pipe operator: `|>`. You can string together different operations in a pipeline, with the result of each line getting "piped" into the next function. For example, below I take some data (named `data`) and perform a series of operations, first selecting a subset of columns, then filtering rows based on whether the values in certain columns meet specified criteria, then I create (`mutate`) a new column averaging across existing columns; and lastly, I summarize the new column down to an average value.

```
data |>
  select(column_a, column_b) |>
  filter(if_all(c(column_a, column_b), ~!is.na(.))) |>
  mutate(column_c = rowSums(across(everything()))) |>
  summarize(mean_sum = mean(column_c))
```

There's a lot going on there, and the specifics will become clearer as we work though this project. But using the pipe operator this way can make for relatively readable code.