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 through 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

- Identify the variables for your correlation analysis
- Search the literature to find relevant research
- Formulate a brief research proposal with your group

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 measure 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 another 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 (rather than 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 Five" personality traits of Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism.

For this project, you will pick one of these traits and investigate its correlation with another question from the survey. The full dataset contains almost 2,000 questions in total. Since this is your first project, however, I am going to constrain your choice. For correlation, you will have two variables. You can think of one as the "predictor" and the other as the "outcome"; you want to see if the predictor has any association with the outcome.

Your **predictor** variable will be one of the Big Five traits:

- Extraversion (V162333 & V162338)
- Agreeableness (V162334 & V162339)
- Conscientiousness (V162335 & V162340)
- Neuroticism (V162336 & V162341)
- Openness (V162337 & V162342)

Your **outcome** variable will be one of the feeling thermometer questions for the two main presidential candidates:

- Democratic presidential candidate (V162078)
- Republican presidential candidate (V162079)

To see how each of these constructs were measured, you will look up the variable IDs in the Codebook. The codebook details the survey methodology exhaustively, including listing every question that was asked. It's not fun reading, but it is your guide to understanding what was measured and how it was measured.

Step 2. Read 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 know which variables you will analyze, you will see what other researchers have found of these (or related) personality traits.

A real research project would involve an exhaustive literature review, in which you attempt to find and understand all the research relevant to your question. Since this project of ours is just for practice and our time is limited, you don't need to read everything; pick one of these papers to skim to give you an idea of what has been found.

Gerber, A. S., Huber, G. A., Doherty, D., & Dowling, C. M. (2011). The big five personality traits in the political arena. *Annual Review of Political Science, 14,* 265-287.

Cooper, C. A., Golden, L., & Socha, A. (2013). The big five personality factors and mass politics. *Journal of Applied Social Psychology, 43(1)*, 68-82.

Step 3. Articulate your hypothesis

Having determined your variables and read 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.

To do for next time

Read up on the development of the TIPI (Gosling et al., 2003).

The more detailed your understanding of how the Big Five trait was measured, the better you will be able to interpret the results of your analysis.

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 demonstrate necessary data cleaning and manipulation in preparation for analysis. By the end of the class you should have modified the example code to work with your selected variables.

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. Usually packages can be installed from within R using the `install.packages()` function. However since the `anesr` 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 special function for installing packages from GitHub.

```r
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:

```
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.

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

Getting data into R

Getting data into R often involves reading in a .csv (comma-separated values) spreadsheet file that you downloaded to your computer. Indeed, you could download the ANES 2016 data file as a .csv from the ANES website and read it into R that way. However, the `anesr` package contains the data so you don't need to download it separately. Instead you can make it available by running this line of code:

```
data(timeseries_2016)
```

When you execute the 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.

```
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 view the data 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 the `select()` function.[1] This allows us to simply type in variable names separated by commas.

For this example I'll look at the correlation between extraversion and the Democratic party feeling thermometer. Extraversion has two TIPI items; their IDs (from the codebook) are `V162333` and `V162338`. The ID for the Democratic Party feeling thermometer is `V161095`. Since I'll probably forget which ID is which, I'll give the columns more meaningful names as I select them.

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

[1] The `select()` function, along with `filter()`, `mutate()`, `across()`, `everything()`, and others that you'll see in my example code, is part of the `tidyverse` family of packages (specifically these all come from the `dplyr` package, but we'll also use functions from other `tidyverse` packages like `tidyr` and `ggplot2`). There are other ways to do all these things without using `tidyverse` packages, just relying on what's referred to as "base" R functions. The `tidyverse` approach just makes this kind of data manipulation generally easier and makes the code more interpretable. If you're curious to see how base R and tidyverse functions differ in syntax, a good place to start is https://dplyr.tidyverse.org/articles/base.html.

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

```
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 to open a tab showing the data 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.[2]

The ANES coding scheme uses negative values for the various kinds of missing or inappropriate data, which makes things simple: only positive values are valid and should be retained.

To implement this as a `filter()`, we can use the `if_all()` function; i.e., we are going to select some columns and *if all* the values in those columns meet some condition the row will

[2] Another way would be to `mutate()` the data, changing the invalid response codes into the value `NA`, R’s special value to indicate missing data. This could be achieved like so:

```
my_data_complete <- my_data |>
  mutate(across(everything(), ~replace(., . < 0, NA)))
```

That would mutate (i.e. change values) across every column. You can read the second part (after the `~`) as “replace the original values (indicated by the placeholder `.`), where the value is less than zero, with `NA`.

be retained. To select the columns we can use the `everything()` function, since the positive-valid/negative-invalid rule is true of every column in our data. The part after the comma, `~ . >= 0`, articulates the condition. The `~` prefix is necessary because instead of naming one specific column to refer to its values we use `.` as a placeholder representing the values in each of the selected columns; the value must be greater than or equal to 0 to be retained.[3]

```
my_data_complete <- my_data |>
  filter(if_all(everything(), ~ . >= 0))
```

Notice that the number of rows in the data.frame has changed, because rows that didn't meet that condition have been dropped.

```
nrow(my_data)
```

```
[1] 4270
```

```
nrow(my_data_complete)
```

```
[1] 3540
```

After filtering to keep only rows with complete data, we're left with 3,540 valid responses.

Computing scale averages

Now that we have selected our columns and filtered out missing/invalid responses, 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.

[3] If the data wasn't as simple or if we just wanted to be more explicit about things, we could filter based on valid responses for each item. For example, valid reponses to the feeling thermometer item are are anything from 0 to 100; anything else is invalid. Therefore we could write a `filter()` condition stating that `feeling_thermometer` (the name of the column) values must be `%in%` the set of values from `0:100`. Likewise for each of the extraversion columns, rows will be retained only if their values are `%in%` the range `1:7`.

```
my_data_complete <- my_data |>
  filter(feeling_thermometer %in% 0:100,
         extraversion1 %in% 1:7,
         extraversion2 %in% 1:7)
```

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` (which I renamed `extraversion1`) is "extraverted, enthusiastic", while item `V162338` (renamed `extraversion2`) 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 (named `extraversion_mean`) consisting of the `rowMeans()` (i.e. an average for each row) `across()` the specified columns (those for which the column name `contains("extraversion")`.

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

Let's see how it looks.

```
my_data_complete
```

```
# A tibble: 3,540 x 4
   extraversion1 extraversion2 feeling_thermometer extraversion_mean
   <hvn_lbll>    <hvn_lbll>    <hvn_lbll>                      <dbl>
 1 6             3              0                                4.5
 2 6             2             15                                4
 3 6             6             50                                6
 4 6             4             30                                5
 5 5             1             70                                3
 6 5             2             15                                3.5
 7 5             7             85                                6
 8 6             5              0                                5.5
 9 7             7             15                                7
10 5             3             50                                4
# i 3,530 more rows
```

We have our two extraversion items (one reverse-coded), the feeling thermometer rating, and the computed extraversion mean scores for each of the 3,540 participants with complete data. We're ready to analyze the data!

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 average 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:

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

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

Describing your data

The most common descriptive statistics are the mean (M) and standard deviation (SD). You should report these for each variable in your analysis.

You can find the mean of each column in a data.frame using R's built-in `colMeans()` function.

```
colMeans(my_data_complete)
```

```
     extraversion1       extraversion2 feeling_thermometer   extraversion_mean 
          4.787006            3.649718           48.317232            4.218362 
```

There's no built-in equivalent for finding the standard deviation of columns, but there is a basic `sd()` function, which you could apply to each column in turn:

```
sd(my_data_complete$extraversion1)
```

```
[1] 1.579163
```

```
sd(my_data_complete$extraversion2)
```

```
[1] 1.764589
```

```
# etc
```

This might be a perfectly appropriate approach, but with a lot of variables it might not be the most efficient (and it kind of violates the DRY principle: don't repeat yourself).

A slightly more complicated but very powerful approach is to use `tidyverse` functions to reshape the data and `summarize()` each of the variables. First, transform the structure of the data using `pivot_longer()`. This produces a data.frame with just two columns, one with all the numeric scores ("value"), and the other labeling which column each value came from ("variable"). Then we `group_by(variable)`, meaning that any subsequent computations will be performed separately for each variable. Finally we pipe the data.frame into the `summarize()` function. There you can create any number of named variables, each computing some kind of summary. Since the data is grouped, each variable ("extraversion1", extraversion2", etc) gets its own count, mean, and standard deviation.

```
my_data_complete |>
  pivot_longer(everything(),
               names_to = "variable",
               values_to = "value",
               values_transform = as.numeric) |>
  group_by(variable) |>
  summarize(count_valid = n(),
            mean = mean(value),
            sd = sd(value))
```

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

Visualizing the data

In addition to reporting the mean and standard deviation, it is useful to visualize the distribution of the data. This can reveal nuances that are not obvious in those single numeric summary values.

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.[4] 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.

Here's a simple histogram of the first extraversion item. I pipe the data into the `ggplot()` function, specifying that I want the `extraversion1` column to be represented as the `x` aesthetic. Then I add geometry using `geom_histogram`. That geom function automatically computes bins and counts; here I just specify I want a `binwidth` of 1, i.e. each column of the histogram will represent one scale point. Note that ggplot layers are added using `+` rather than the usual `|>` pipe.

[4] 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.

```
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.
```

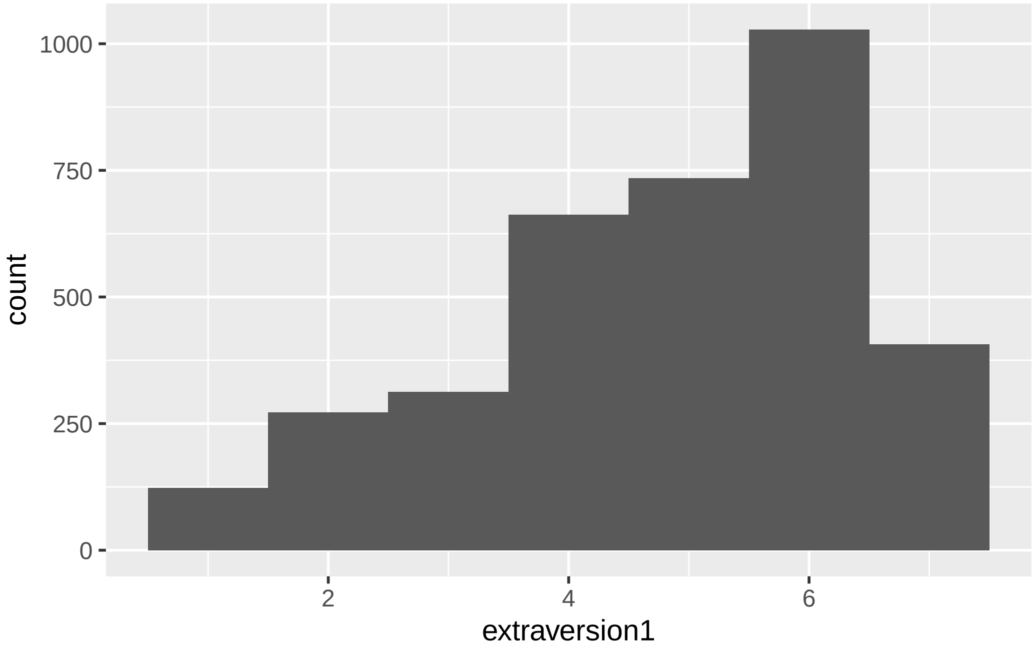


Figure 1: Histogram of responses to “extraverted, enthusiastic” TIPI item

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()
)
```

I’ll also customize the “breaks” on the x-axis (where the ticks and numeric labels go) and the axis labels.

```
my_data_complete |>
  ggplot(aes(x = extraversion1)) +
  geom_histogram(binwidth = 1, color = "white") +
  scale_x_continuous(breaks = 1:7) +
  labs(x = "Responses to extraversion item 1: extraverted, enthusiastic",
       y = "Number of responses") +
  theme_apa
```

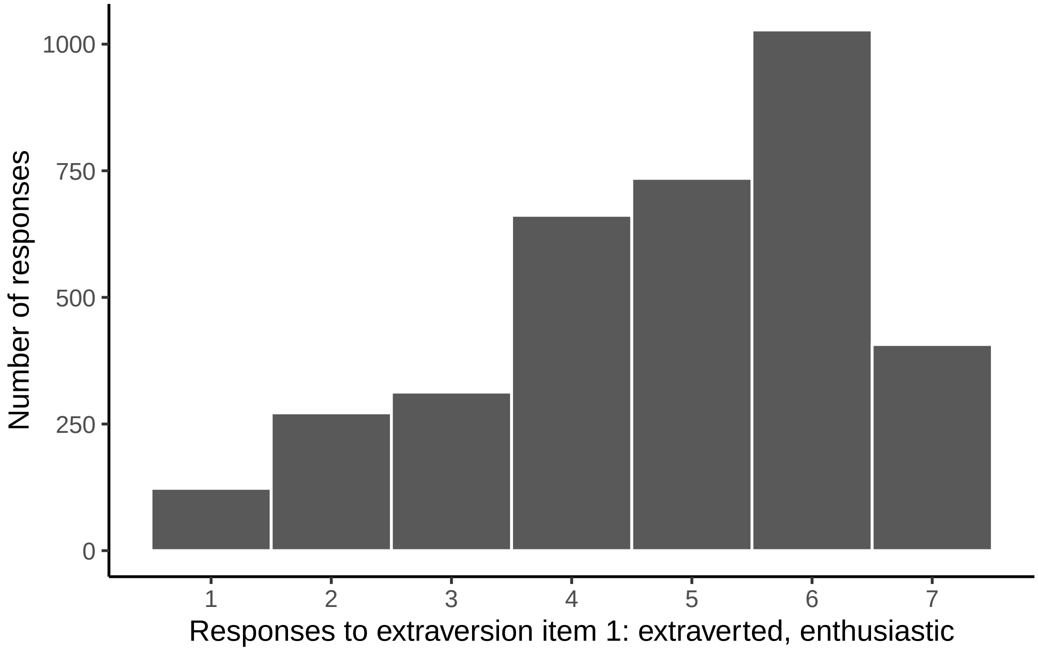


Figure 2: Histogram of responses to “extraverted, enthusiastic” TIPI item

Here’s a histogram of the other TIPI extraversion item.

```
my_data_complete |>
  ggplot(aes(x = extraversion2)) +
  geom_histogram(binwidth = 1, color = "white") +
  scale_x_continuous(breaks = 1:7) +
  labs(x = "Responses to extraversion item 2: reserved, quiet (reverse-coded)",
       y = "Number of responses") +
  theme_apa
```

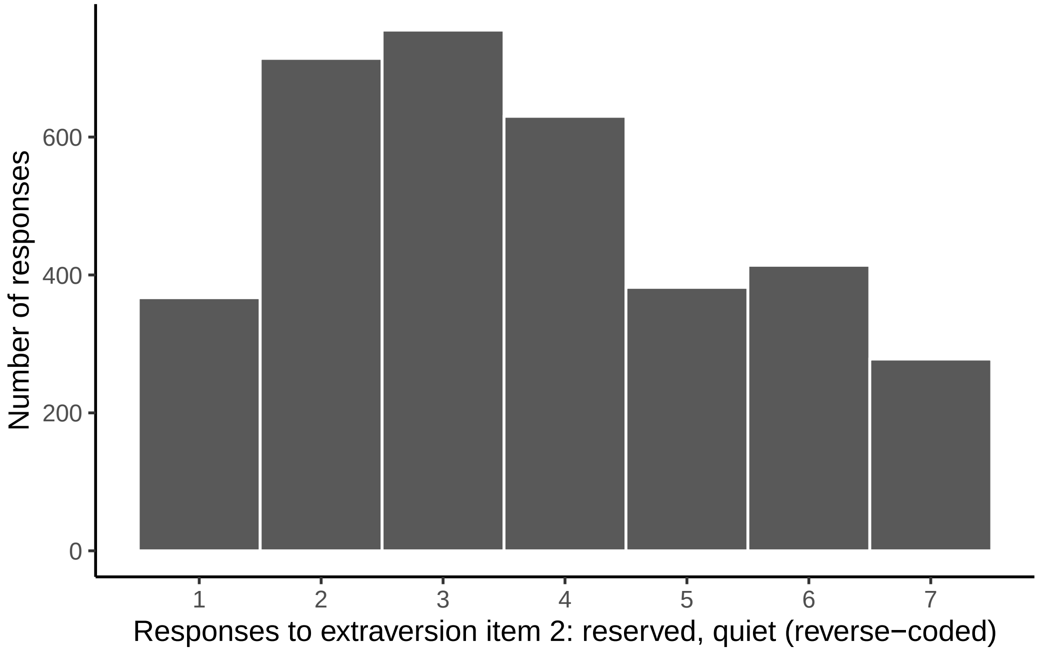


Figure 3: Histogram of responses to "reserved, quiet" TIPI item

And here's a histogram of the average extraversion scores I computed.

```r
my_data_complete |>
  ggplot(aes(x = extraversion_mean)) +
  geom_histogram(binwidth = 0.5, color = "white") +
  scale_x_continuous(breaks = 1:7) +
  labs(x = "Average scores across both TIPI extraversion items",
       y = "Count") +
  theme_apa
```

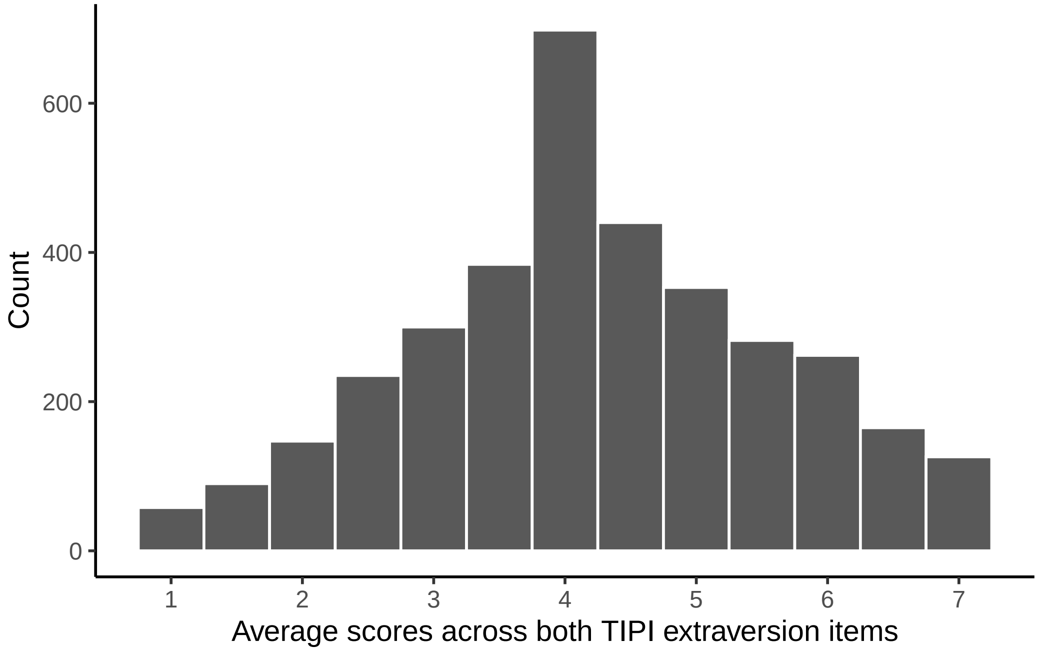


Figure 4: Histogram of average scores on TIPI Extraversion subscale

Notice that while both individual extraversion items were a bit skewed, the distribution of averages is approximately normally-distributioned (albeit with a big spike in the middle).

Lastly, I'll make a histogram of the feeling thermometer variable.

```r
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)) +
  labs(x = "Responses to Democratic Party feeling thermometer",
       y = "Count") +
  theme_apa
```

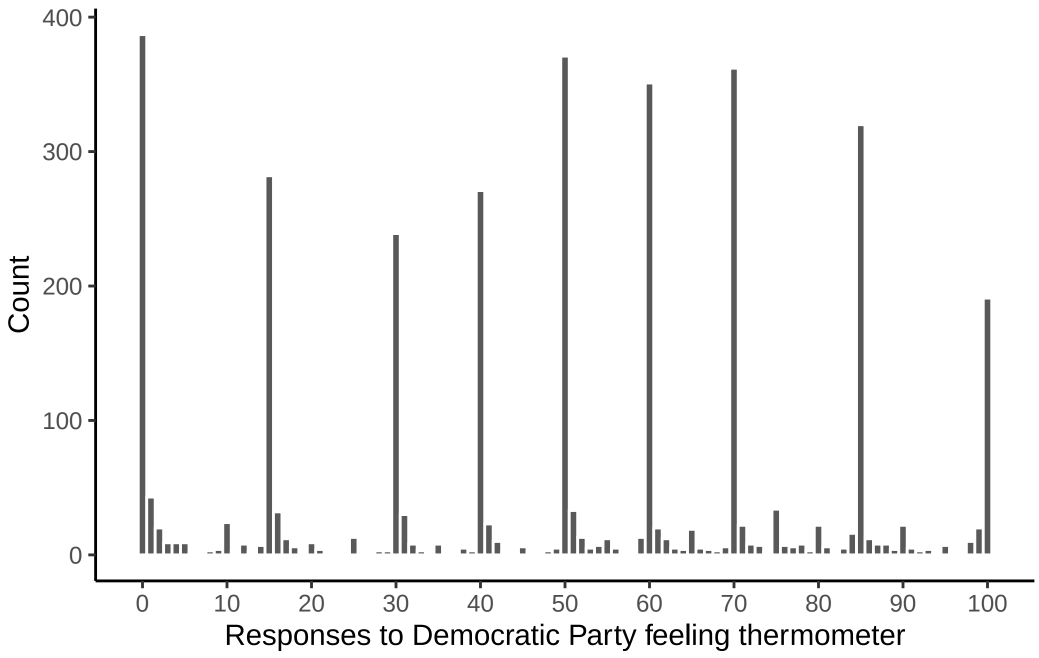


Figure 5: Histogram of responses to Democratic Party feeling thermometer

I chose a binwidth of 1, which isn't necessarily the most appropriate value for a 0 to 100, but it does reveal an interesting distribution of responses. People's responses are not evenly distributed across the 0 to 100 scale; rather, some values (particularly multiples of 10) are chosen much more frequently than others.

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.

```r
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:

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

```
[1] -0.01012898
```

The `cor.test()` function goes further than `cor()`, giving you the *p*-value necessary for determining statistical significance[5] and some other information about the correlation.

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

```

    Pearson's product-moment correlation

data:  my_data_complete$extraversion_mean and my_data_complete$feeling_thermometer
t = -0.60251, df = 3538, p-value = 0.5469
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.04305838  0.02282241
sample estimates:
        cor 
-0.01012898 
```

Lastly, let's make a scatterplot visualizing the correlation.

[5] Remember that, by convention, psychologists generally use $\alpha = .05$ as the criterion for statistical significance, meaning that if our data has less than a 5% chance of occurring under the null hypothesis we reject the null and tentatively accept the alternative hypothesis that the variables are associated.

```
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
```

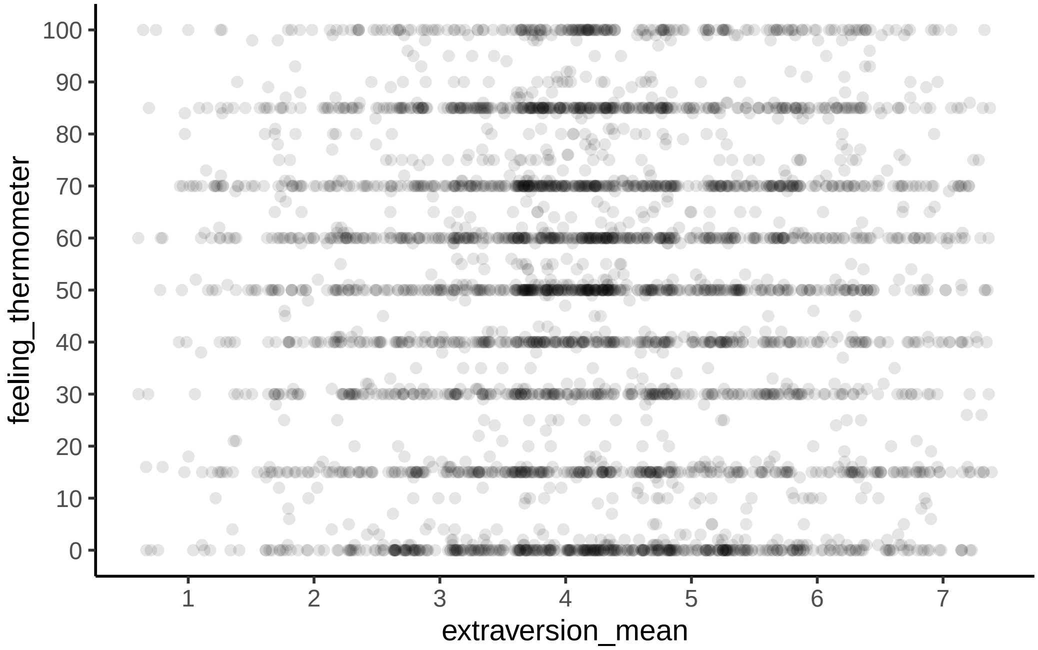


Figure 6: Scatterplot (with jitter) of average extraversion scores and feeling thermometer scores

You can see horizontal bands which correspond to those big spikes on the feeling thermometer that the histogram revealed. Consistent with the correlation coefficient which was close to zero with a nonsignificant *p*-value, visually it doesn't look like there's much of an association between the feeling thermometer responses and extraversion scores.

Lab 5: Presentation & report

This week each project team will present their project to the rest of the class. After that, you will be ready to write up your findings. Note that presentations (and all the preceding work) will be a group effort, but written reports must be completed individually.

Presentation

Guide to presenting

Each team will give a short presentation which should encapsulate the motivation, methods, anticipated findings, and interpretation of your proposed project. Aim for clarity, conciseness, and being bold to spark the audience's interest in your topic and findings.

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.

Make your slides 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 presentation the group will 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! But here are some thoughts…"). 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 ("You might be wondering…").

It is up to each group to decide how to divide up the 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. 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.

Format

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 (a technical report of any descriptive statistics, figures, and statistics you produced)
- Discussion (a paragraph or two interpreting your results and drawing conclusions)

Deadline

The report is due by the next class (see late policy from Syllabus).

Grading

You will receive a score out of 100 for your report.

Grade	Point range	Description
A+	97-100	Outstanding and exceptional work. The report clearly articulates the problem, purpose, methods, and results. There is evidence of critical thought, and the report goes beyond the assignment requirements in terms of analysis or presentation. The report demonstrates a sophisticated understanding of the concepts and techniques used. The report is free of errors and is clearly and professionally written.
A	90-97	Excellent work. The report clearly articulates the problem, purpose, methods, and results. There is evidence of critical thought. The report demonstrates a strong understanding of the concepts and techniques used. The report is virtually free of errors and is clearly and professionally written.
B	80-89	Above average work. The report articulates the problem, purpose, methods, and results. There is some evidence of critical thought. The report demonstrates a good understanding of the concepts and techniques used. There are minor errors in the report, but the writing is generally clear and professional.
C	70-79	Satisfactory work. The report articulates the problem, purpose, methods, and results but may lack clarity or detail. There is minimal evidence of critical thought. The report demonstrates an acceptable understanding of the concepts and techniques used. There are noticeable errors in the report, and the writing could be improved.
D	60-69	Below average work. The report does not clearly articulate the problem, purpose, methods, or results. There is little to no evidence of critical thought. The report demonstrates a minimal understanding of the concepts and techniques used. There are significant errors in the report, and the writing is unclear.
F	<60	Unsatisfactory work. The report does not articulate the problem, purpose, methods, or results. There is no evidence of critical thought. The report demonstrates a lack of understanding of the concepts and techniques used. The report is filled with errors, and the writing is poor.

Project 2: Multiple Regression

Lab 6: Project planning

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

Goals

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

Project overview

Multiple regression goes beyond simple correlations and captures the relationships between multiple **predictor** variables and a single **outcome** variable. It allows us to see the combined impact of multiple variables on one outcome, and how these predictors may interact with each other to affect the outcome. This can provide valuable insights into how the relationship between one predictor and the outcome might depend on another predictor.

For instance, researchers might examine how one or more Big Five personality traits interact with age to predict job satisfaction. Similarly, variables such as income, education, and openness might be used in a multiple regression analysis to predict political ideology. Like with correlations, this doesn't mean that every extroverted, conscientious, and emotionally stable person will always be satisfied with their job, or that all educated, and open individuals are politically aligned in the same way. It's about general tendencies rather than strict rules.

Like correlation, regression studies involve determining which psychological constructs to study, how to operationally define those constructs, and how to measure them. They then use these definitions and measurements to explore relationships, using appropriate statistical analyses.

With this project, you will dive deeper into the interplay between personality traits and other constructs, using multiple regression analysis to explore how a combination of predictors contributes to an outcome of your choice by analyzing existing data. This will empower you to better understand the complex interactions that shape human behavior and personality, beyond simple one-to-one relationships.

Step 1. Choose your dataset and variables

We will again use data from the ANES. I suggest that you use the 2016 dataset again; you may even look a the same variables you did with Project 1 and just add one or more new predictors. However, if you're feeling ambitious or limited by what's available in the 2016 dataset you may choose a different one (the most recent is from 2022), or even use the cumulative timeseries data which combines data from across the many years the study has been conducted (this would be a good choice if you want to see how the passage of time might have contributed to a change on some outcome of interest).

You should pick one outcome variable (which should be continuous, i.e. scores cover some numeric range rather than discrete categories), and at least two / up to four predictor variables (at least one of which is a TIPI Big Five trait).

Step 2. Find relevant research

As with Project 1, your approach and expectations should be informed by what has come before. 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 out about these (or related) personality traits and outcomes.

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 computing the regression model.

The general hypothesis of a multiple regression is that there is a relationship between the predictor variables and the outcome variable; in other words, the predictors allow us to predict scores on the outcome more accurately than you would expect if the variables we unrelated.

A choice you will make at this point is whether you will study the simple additive effect of your predictors, or whether you will look at the **interaction** between your predictors. An interaction effect means that two (or more) variables combined have a significantly larger effect on the outcome variable as compared to the sum of the individual variables alone. If you have two predictors, I would suggest looking at their interaction; if you want to include more than two predictors, just look at their additive effect.

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
- Your theoretical perspective (based on the research you found)
- Your expectations (this should follow from your theoretical perspective)

Lab 7: Data cleaning & analysis

Goals

- 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)

Data wrangling, description, and visualization

Data wrangling

Building on the correlation example, we will include additional variables of interest - conscientiousness and agreeableness - to examine how these factors, along with extraversion, collectively predict feelings towards the Democratic party. Similar to the correlation project, we will start by cleaning and filtering the data, recoding the negatively-worded items (taking care to note which ones need recoding; it's not always the second question), and computing mean scores for each Big 5 trait.

Here is the pipeline to prepare the data:

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

my_data_complete <- timeseries_2016 |>
  select(extraversion1 = "V162333",
         extraversion2 = "V162338",
         conscientiousness1 = "V162335",
         conscientiousness2 = "V162340",
         agreeableness1 = "V162334",
         agreeableness2 = "V162339",
         feeling_thermometer = "V161095") |>
  filter(if_all(everything(), ~ . >= 0))  |>
  mutate(extraversion2 = 8 - extraversion2,
         conscientiousness2 = 8 - conscientiousness2,
         agreeableness1 = 8 - agreeableness1,
         extraversion_mean = rowMeans(across(contains("extraversion"))),
         conscientiousness_mean = rowMeans(across(contains("conscientiousness"))),
         agreeableness_mean = rowMeans(across(contains("agreeableness"))))
```

Describing your variables

Just as in the previous lab, you'll need to compute the mean and standard deviation for each of your variables. Use the same process, replacing the variable names with your new ones:

```
my_data_complete |>
  pivot_longer(everything(),
               names_to = "variable",
               values_to = "value",
               values_transform = as.numeric) |>
  group_by(variable) |>
  summarize(count_valid = n(),
            mean = mean(value),
            sd = sd(value))
```

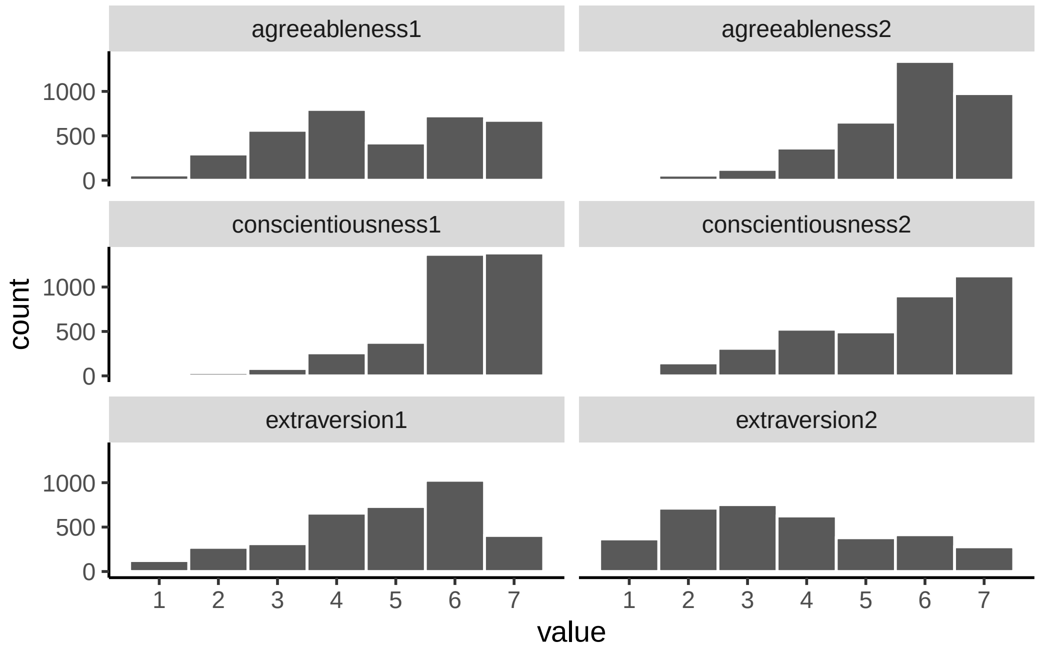
```
# A tibble: 10 x 4
   variable               count_valid  mean    sd
   <chr>                        <int> <dbl> <dbl>
 1 agreeableness1                3530  4.73  1.67
 2 agreeableness2                3530  5.68  1.23
 3 agreeableness_mean            3530  5.21  1.14
 4 conscientiousness1            3530  5.99  1.16
 5 conscientiousness2            3530  5.41  1.54
 6 conscientiousness_mean        3530  5.70  1.12
 7 extraversion1                 3530  4.79  1.58
 8 extraversion2                 3530  3.65  1.77
 9 extraversion_mean             3530  4.22  1.38
10 feeling_thermometer           3530 48.3  30.0
```

Visualizing the data

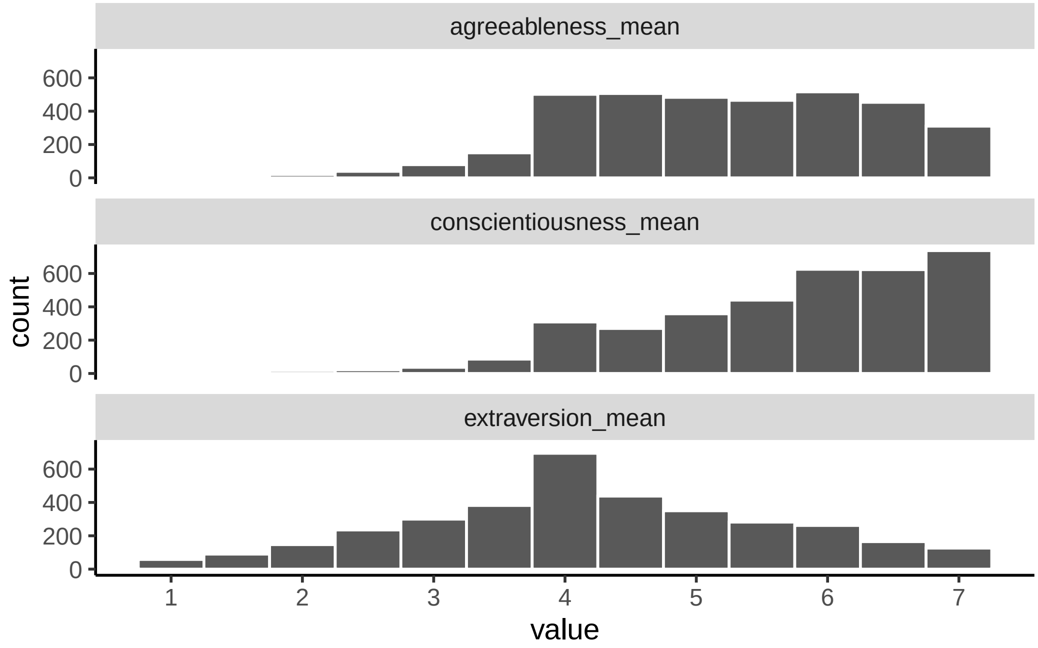
You can create histograms for each of your new variables, just like you did for extraversion. Since there are

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

```
my_data_complete |>
  select(extraversion1:agreeableness2) |>
  pivot_longer(everything(),
               names_to = "variable",
               values_to = "value",
               values_transform = as.numeric) |>
  ggplot(aes(x = value)) +
  geom_histogram(binwidth = 1, color = "white") +
  scale_x_continuous(breaks = 1:7) +
  facet_wrap(~variable, nrow = 3) +
  theme_apa
```



```
my_data_complete |>
  select(contains("mean")) |>
  pivot_longer(everything(),
               names_to = "variable",
               values_to = "value",
               values_transform = as.numeric) |>
  ggplot(aes(x = value)) +
  geom_histogram(binwidth = 0.5, color = "white") +
  scale_x_continuous(breaks = 1:7) +
  facet_wrap(~variable, nrow = 3) +
  theme_apa
```



The regression analysis

Simple linear regression

Now we'll perform a multiple regression analysis. This allows us to examine the relationship between one dependent variable (in this case, feeling_thermometer) and several independent variables (extraversion_mean, conscientiousness_mean, and agreeableness_mean).

```
model <- lm(feeling_thermometer ~ extraversion_mean + conscientiousness_mean + agreeablene

summary(model)
```

```
Call:
lm(formula = feeling_thermometer ~ extraversion_mean + conscientiousness_mean +
    agreeableness_mean, data = my_data_complete)

Residuals:
    Min      1Q  Median      3Q     Max
-54.523 -24.494   2.127  22.216  55.932

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)            51.218290   3.342704  15.322  < 2e-16 ***
extraversion_mean      -0.006588   0.369615  -0.018 0.985780
conscientiousness_mean -1.633627   0.476950  -3.425 0.000621 ***
agreeableness_mean      1.234787   0.465963   2.650 0.008086 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 29.96 on 3526 degrees of freedom
Multiple R-squared:  0.00422,   Adjusted R-squared:  0.003373
F-statistic: 4.981 on 3 and 3526 DF,  p-value: 0.001893
```

The summary() function will output the results of your regression analysis. For each predictor, you'll see an estimate of the relationship between that predictor and the outcome variable, controlling for the other predictors. For instance, if you have a predictor like 'Age', the Estimate for 'Age' would indicate how much the response variable changes on average with a one-unit increase in 'Age', holding all other predictors constant.

You'll also see a t-value and a p-value for each predictor, which tell you whether each predictor is significantly related to the outcome variable, controlling for the other predictors.

The last part of the output gives the overall model fit. Multiple R-squared is the proportion of variance in the outcome variable that can be explained by the predictor variables, and Adjusted R-squared is a version of R-squared adjusted for the number of predictors.

Finally, the F-statistic and its corresponding p-value assess the overall significance of the model. If the p-value associated with this F-statistic is less than your significance level, you can reject the null hypothesis that all the regression coefficients are zero.

Visualizing a regression model

Finally, as you did in the correlation project, you'll want to create a scatterplot to visualize the relationships between your predictors and the outcome variable. This will be more complex than in the correlation project, since you now have more variables to include.

Regression with interaction

When specifying a regression model which includes the interaction between two predictors, the only difference is that you separate the names of the predictor variables with a `*` rather than `+`.

```r
model <- lm(feeling_thermometer ~ conscientiousness_mean * agreeableness_mean, data = my_d

summary(model)
```

```

Call:
lm(formula = feeling_thermometer ~ conscientiousness_mean * agreeableness_mean, 
    data = my_data_complete)

Residuals:
   Min     1Q Median     3Q    Max 
-54.18 -24.45   1.88  21.91  56.72 

Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                                61.3864    11.6794   5.256 1.56e-07
conscientiousness_mean                     -3.4191     2.0280  -1.686   0.0919
agreeableness_mean                         -0.8155     2.3145  -0.352   0.7246
conscientiousness_mean:agreeableness_mean   0.3546     0.3920   0.905   0.3658
                                             
(Intercept)                               ***
conscientiousness_mean                    .  
agreeableness_mean                           
conscientiousness_mean:agreeableness_mean    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 29.96 on 3526 degrees of freedom
Multiple R-squared:  0.004451,	Adjusted R-squared:  0.003604 
F-statistic: 5.255 on 3 and 3526 DF,  p-value: 0.001288
```

If the interaction term is significant, it's also important to conduct a post hoc analysis to probe the interaction. This often involves plotting the interaction or calculating the effect of one variable at different levels of the other variable, to get a clearer understanding of how the predictors interact to affect the outcome.

Keep in mind that interpreting interaction terms can be complex. It is crucial to consider the nature of your variables, the context of your research, and the practical significance of the interactions, not just the statistical significance.

Lab 8: Visualization & interpretation

Goals

- Check the assumptions of multiple regression have been met
- Visualize your regression model
- Interpret the findings

Checking model assumptions

Lab 9: Presentation & report

Present in class.

Write up due by next class.

Project 3: Scale design

Lab 10: Scale planning

My hope is that at this point you are feeling a sense of accomplishment at what you have achieved through the secondary data analysis projects so far, but that you're also feeling a bit limited by choice of variables you had and the inability to decide what exactly to measure or how to measure it. In this lab we'll begin the final project: devising and piloting a novel measure of a personality trait of your choice.

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

Scale design

Scale design is a fundamental aspect of empirical psychology. Psychological constructs such as personality traits, intelligence, attitudes, or mental health conditions are abstract and not directly observable. Psychologists use scales, or tests, to measure these constructs, translating abstract ideas into quantifiable variables. For instance, the Big Five personality traits are typically measured using self-report scales like the Ten-Item Personality Inventory (TIPI) or the NEO-PI-R.

Creating a good psychological scale involves several steps. First, researchers need to clearly define the construct they want to measure. Then, they develop a set of items, or questions, that reflect the construct. These items should be clear, easy to understand, and specific to the construct. After developing the items, researchers typically pilot the scale on a small sample to check that the items measure what they're supposed to. The scale is then refined based on this pilot data. Once the scale is finalized, it can be used in research.

Importantly, creating a scale is not a one-time process. Scales need to be validated – tested to make sure they're measuring what they're supposed to. This involves collecting data and conducting statistical analyses to check the reliability (consistency of the scale) and validity (whether the scale measures the construct it is supposed to) of the scale.

In this project, you will take on the role of a scale developer, designing a new scale to measure a construct of your choosing. You'll begin by defining your construct, generating a set of items, and establishing a response format. You'll then create a plan to pilot and validate your scale. This project will provide you with firsthand experience of the challenges and rewards of psychological measurement and get you thinking about underexplored aspects of personality.

Step 1. Decide what aspect of personality you want to measure

Brainstorm ideas with your group. Something relevant to your personality experience.

To give you some ideas, here are a few existing personality scales.

Step 2. Find relevant research

Once you have an idea for what personality construct you want to measure.

Step 3. Come up with a plan

Lab 11: Questionnaire design

Goals

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

Lab 12: Data collection and analysis

Lab 13: Presentation & report

Present in class.

Write up due by next class.

Getting started with R

posit.cloud

You will use posit.cloud to write R code and work with data in RStudio. To use it you'll just need to sign up for a free account.

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 immensely.

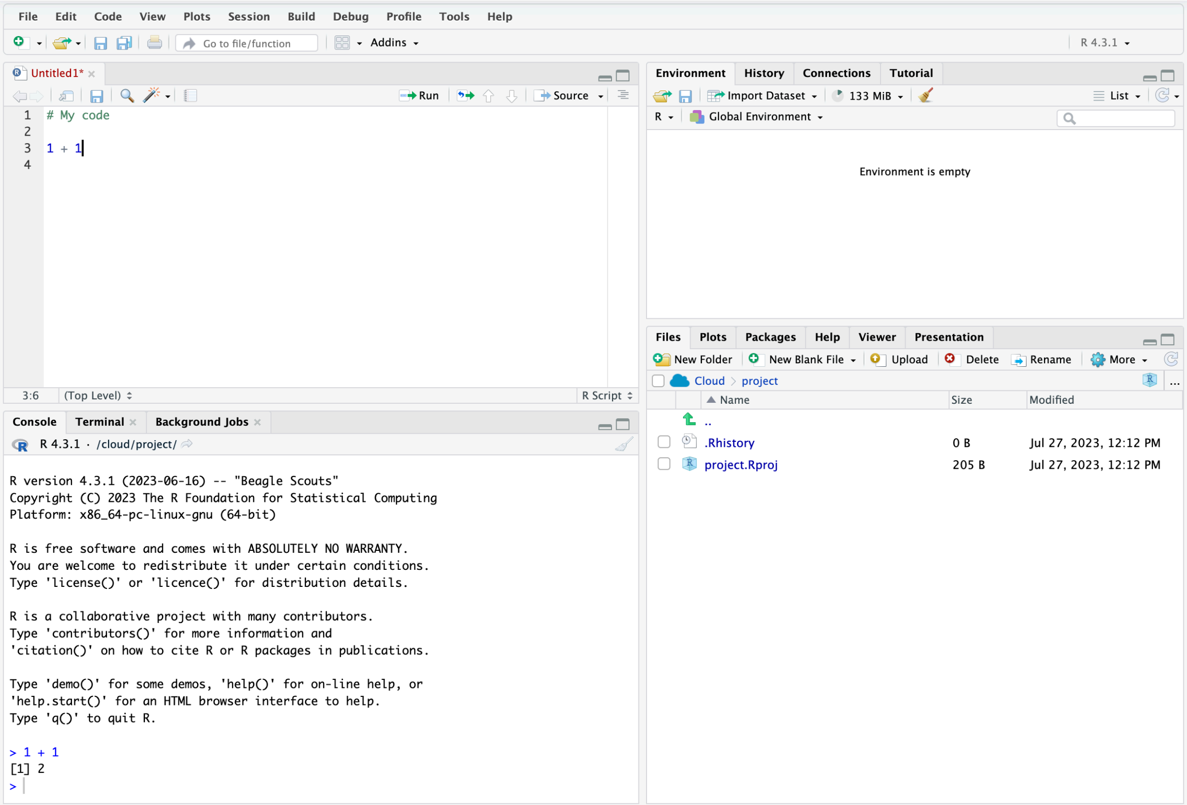
Fundamentals of R for data analysis

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.

Here are some of the basics to help you get started coding in R.

RStudio

RStudio is the interface we'll use to write and run R code and see its output. The interface has 4 panels, each with a few tabs:



```
# My code

1 + 1
```

```
R version 4.3.1 (2023-06-16) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 1 + 1
[1] 2
>
```

Environment is empty

Name	Size	Modified
..		
.Rhistory	0 B	Jul 27, 2023, 12:12 PM
project.Rproj	205 B	Jul 27, 2023, 12:12 PM

- Top-left: Code editor / data viewer
    - You will type code here
    - You can run a line of code by clicking on it and pressing Ctrl/Cmd + Enter on your keyboard
- Bottom-left: R console
    - You can type code directly and run it by pressing enter.
    - You won't be saving your code as a document like when you type in in the editor, so this is useful for just testing something before you commit it to your working document
- Top-right: Environment
    - As you excute code you may be creating objects like sets of numbers of data.frames. Those objects will appear here.

    - You can click the name of some objects, like data.frames, and it will open a view of the data as a tab in the editor pane
- Bottom-right: Files/folders, plot viewer, help window
    - You can navigate the file tree, and you will see any plots you create appear here

Assignment

R has a fancy assignment operator: `<-`.[1] 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
```

[1] Most other coding languages tend to use a boring `=` for assignment. Sure it's nice not having to type an extra character, but there's a keyboard shortcut to quickly add an `<-` in RStudio: Option/Alt + `-`. And philosophically, the `<-` arrow conveys the inherent directionality of the assignment operation. The object is assigned to the name; the object and its name are not equal and so the `=` arguably gives a misleading impression of the two things being one and the same. (Also, to let you in on a secret, `=` also works for assignment in R.)

```
mean(doubled_numbers)
```

```
[1] 11
```

You can also nest functions inside one another.

```
sqrt(mean(original_numbers))
```

```
[1] 2.345208
```

A function generally has one or more "arguments", to which you supply parameters. For example, the `mean()` function's first argument is the set of numbers you want to compute the mean of; in the previous examples `original_numbers` and `doubled_numbers` were the parameters I supplied. You don't necessarily have to type the name of the argument, but it can be helpful. The `seq()` function, for example, produces a sequence of numbers according to three arguments, `from`, `to`, and `by`.

```
seq(from = 1, to = 10, by = 2)
```

```
[1] 1 3 5 7 9
```

When you don't type the names of the arguments, R matches them by position, so this gives exactly the same output as the previous line of code:

```
seq(1, 10, 2)
```

```
[1] 1 3 5 7 9
```

You can get help with a function (to see what arguments it accepts, for example) by typing a question mark followed by the function name (without parentheses) in your console.

```
?mean
```

Running the code will bring up the function's help documentation in RStudio's Help pane.

Piping

You can string together different operations in a pipeline using the pipe operator: |>.[2] The result of each line of code gets "piped" into the function on the next line as its first argument. 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.

[2] If you're looking at R code from beyond this handbook (e.g. looking up help elsewhere) you may see a different pipe: %>%. The |> pipe, called the "native" pipe, was only included as a feature of base R relatively recently. Until then, the %>% pipe was provided by an external package (called magrittr. Get it?). In practice the pipes work similarly, so you can often just replace %>% with |> and it'll work fine, but it's worth being aware of.

Grading rubric

For each presentation and project report you will receive a score out of 100 according to the rubric below.

Note that presentations will be given jointly. In general, grades will be the same for all group members as it is expected that group members will contribute equally to the presentation; however, exceptions may be made when it is clear that group members did not all contribute equally.

Also note that even though projects will be a group effort, the written reports will be completed individually.

Grade	Point range	Description
A+	97-100	Outstanding and exceptional work. The report clearly articulates the problem, purpose, methods, and results. There is evidence of critical thought, and the report goes beyond the assignment requirements in terms of analysis or presentation. The report demonstrates a sophisticated understanding of the concepts and techniques used. The report is free of errors and is clearly and professionally written.
A	90-97	Excellent work. The report clearly articulates the problem, purpose, methods, and results. There is evidence of critical thought. The report demonstrates a strong understanding of the concepts and techniques used. The report is virtually free of errors and is clearly and professionally written.
B	80-89	Above average work. The report articulates the problem, purpose, methods, and results. There is some evidence of critical thought. The report demonstrates a good understanding of the concepts and techniques used. There are minor errors in the report, but the writing is generally clear and professional.
C	70-79	Satisfactory work. The report articulates the problem, purpose, methods, and results but may lack clarity or detail. There is minimal evidence of critical thought. The report demonstrates an acceptable understanding of the concepts and techniques used. There are noticeable errors in the report, and the writing could be improved.

Grade	Point range	Description
D	60-69	Below average work. The report does not clearly articulate the problem, purpose, methods, or results. There is little to no evidence of critical thought. The report demonstrates a minimal understanding of the concepts and techniques used. There are significant errors in the report, and the writing is unclear.
F	<60	Unsatisfactory work. The report does not articulate the problem, purpose, methods, or results. There is no evidence of critical thought. The report demonstrates a lack of understanding of the concepts and techniques used. The report is filled with errors, and the writing is poor.