Social Psychology Lab Handbook

Barnard College
Department of Psychology
PSYC BC2137

Spring 2024

Table of contents

Syllabus

> Only when certain events recur in accordance with rules or regularities, as in the case of repeatable experiments, can our observations be tested—in principle—by anyone. We do not take even our own observations quite seriously, or accept them as scientific observations, until we have repeated and tested them. Only by such repetition can we convince ourselves that we are not dealing with a mere isolated 'coincidence,' but with events which, on account of their regularity and reproducibility, are in principle inter-subjectively testable.
>
> – Karl Popper (1959)

Course details

Pre-requisites: PSYC BC1001 Introduction to Psychology; PSYC BC1101 Statistics; PSYC BC1020 Research Methods
Co-requisite: PSYC BC2138 Social Psychology

Time, venue & instructor

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

Instructor: Dr. Rob Brotherton (rbrother@barnard.edu)
Office hour: Friday 10-11AM, Milbank 415M

Course overview

This lab will usually be taken concurrently with BC2138 Social 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, each involving an analysis of existing data. The first will involve planning and executing a correlational analysis. The second will involve performing an ANOVA analysis. For the third project, you will design your own analysis. Through these projects you will gain experience in formulating social-psychological research questions; 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 partners 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 2/14, and the slide must be submitted before the next class on 2/21.

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*
1/24	Course overview	
Project 1: Correlation		
1/31	Project planning	
2/7	Data preparation	
2/14	Analysis	Presentation slide
2/21	Presentations	Project 1 report
Project 2: ANOVA		
2/28	Project planning	
3/6	Data preparation & analysis	
3/13	*No class (Spring Break)*	
3/20	Visualization & interpretation	Presentation slide
3/27	Presentations	Project 2 report
Project 3: Own design		
4/3	Project planning	
4/10	Analysis	
4/17	Visualization & interpretation	Presentation slide
4/24	Presentation & report	Project 3 report

* 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 2020 dataset. By the end of the session you will have a plan for your analysis.

Goals

- Understand the purpose of a correlational design
- Identify the variables for your analysis
- Read some relevant research
- Articulate your hypothesis

Report template

You'll use a template to eventually write up your individual report paper (including the code that you will start writing next week to analyze the data). The template exists in posit.cloud, the cloud-based RStudio interface I mentioned last week. A link to to join the posit.cloud 'space' for the course will be emailed separately.

Once you have joined the 'space' you should see one link there called "Projects." Click into it, and then look for "report_1.qmd" under the "Files" tab in the bottom-right pane. Click that file to open up the first project template.

I strongly encourage you to start keeping notes for your Intro in that document as you start forming your ideas for the project today.

Project overview

Correlational designs

With this project you will examine a correlation between social psychological constructs using real survey data.

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 something correlates with

something else–for example, satisfaction with ones friendships is correlated with wellbeing or prejudice is correlated with endoresement of stereotypes towards a group–is to say that those things *tend* to go together. Not everyone who feels great satisfaction in their friendships will report greater wellbeing than anyone with less social satisfaction, 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 social 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. There are strengths, limitations, and trade offs involved in every decision along the way.

Step 1. Examine the data and identify your variables

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 2020 election, since it's the most recent survey.

The kind of correlation analysis you will perform examines whether two constructs are related. The full dataset contains more than 1,000 questions reflecting various constructs. Since this is your first project I am going to constrain your choice. You will investigate how perceived threats are associated with trust. Does distrust of the government and/or other people tend to go together with feeling threatened in some way?

One of your variables will be a question assessing **perceived threat**:

- How worried are you about losing your job in the near future? (V201540)
- So far as you and your family are concerned, how worried are you about your current financial situation? (V201594)
- How concerned are you about losing your health insurance in the next year? (V201621)
- How concerned are you about being able to pay health care expenses for you and your family in the next year? (V201622)
- How worried do you feel about how things are going in the country? (V201120)
- What do you think about the state of the economy these days in the United States? (V201324)
- How worried are you that the United States will experience a terrorist attack in the near future? (V202358)

Your second variable will be a question about **trust**:

- How often can you trust the federal government in Washington to do what is right? (V201233)

- How many of the people running the government are corrupt? (V201236)
- Generally speaking, how often can you trust other people? (V201237)

To see exactly how these constructs were measured you will look up the variable IDs (the codes beginning with V in parentheses above) in the Codebook.

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 about these (or related) constructs.

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 is just for practice and our time is limited, you don't need to read everything; this paper should give you an idea of what has been found:

Schlipphak, B. (2021). Threat perceptions, blame attribution, and political trust. *Journal of Elections, Public Opinion and Parties*. https://doi.org/10.1080/17457289.2021.2001474

Step 3. Articulate your design and hypothesis

By this point you should be able to state your:

- **Operational definitions** (that is, the specific questions that participants were asked and how they could answer, per the codebook)
- The **constructs** that those operational definitions measure (i.e. does the question measure perceived threat *in general*? Or some more specific form of perceived threat?)
- Your **hypothesis**

Your hypothesis is a formal statement of your expectation about how your constructs are (or aren't) associated, and it will be tested quantitatively by calculating a correlation statistic.

The main question that your hypothesis addresses is: do you think the two variables will be significantly correlated? That is, will you find an association consistent enough that it doesn't just seem to be attributable to chance variation in the data?[1]

If you do expect a significant correlation, you should also specify whether you expect it to be positive or negative, and how strong you expect it to be (i.e. weak, moderate, strong; see Appendix B).

[1] Even random data will produce spurious correlations by chance some of the time; see spuriouscorrelations.com

Lab 3: Data preparation

You will begin this session with variables from the ANES 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)
- Describe and visualize your variables

Working with data in R

Getting R ready

In addition to being a source of high-quality data relevant to social psychology, the ANES dataset is convenient for our purposes because someone went to the trouble of creating an R package which makes working with the data relatively straightforward (not that you won't still run into issues!): `anesr` (github.com/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.

Generally you would need to install these packages yourself, but one of the advantages of providing you the Projects templates in posit.cloud is that I already installed the packages behind the scenes, so you don't have to.[2]

[2] If I hadn't already installed the packages for you, it generally wouldn't be too much trouble. Most packages, like `tidyverse`, can be installed by running a single like of code:

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

Once the packages are installed, however, they still need to be "activated". The `library()` function activates a package, makings its functions and data available for use. (So you will need these two lines of code in your own analysis document.)

```
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 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_2020)
```

When you execute the code you won't see any output, but you should see the name `timeseries_2020` 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_2020
```

```
# A tibble: 8,280 x 1,381
   version         V200001 V160001_orig V200002 V200003 V200004 V200005 V200010a
   <chr>             <dbl>   <hvn_lbll> <hvn_l> <hvn_l> <hvn_l> <hvn_l>    <dbl>
 1 ANES2020TimeSe~  200015       401318       3       2      -2       0    0.828
 2 ANES2020TimeSe~  200022       300261       3       2      -2       0    1.09
 3 ANES2020TimeSe~  200039       400181       3       2      -2       0    0.672
```

The `anesr` package is slightly more complicated since it exists only on github.com, not R's official repository of packages. Therefore installing it requires two lines of code:

```
install.packages("devtools")
devtools::install_github("jamesmartherus/anesr")
```

The `devtools` packages provides the `install_github()` function that allows the `anesr` package to be installed from github!

```
 4 ANES2020TimeSe~  200046       300171       3       2       -2       0    0.492
 5 ANES2020TimeSe~  200053       405145       3       2       -2       1    1.19
 6 ANES2020TimeSe~  200060       400374       3       2       -2       0    0.339
 7 ANES2020TimeSe~  200084       407013       3       2       -2       0    0.525
 8 ANES2020TimeSe~  200091       407174       3       2       -2       0    0.729
 9 ANES2020TimeSe~  200107       406264       3       2       -2       0    1.42
10 ANES2020TimeSe~  200114       402782       3       2       -2       1    2.56
# i 8,270 more rows
# i 1,373 more variables: V200010b <dbl>, V200010c <dbl>, V200010d <dbl>,
#   V200011a <dbl>, V200011b <dbl>, V200011c <dbl>, V200011d <dbl>,
#   V200012a <dbl>, V200012b <dbl>, V200012c <dbl>, V200012d <dbl>,
#   V200013a <dbl>, V200013b <dbl>, V200013c <dbl>, V200013d <dbl>,
#   V200014a <dbl>, V200014b <dbl>, V200014c <dbl>, V200014d <dbl>,
#   V200015a <dbl>, V200015b <dbl>, V200015c <dbl>, V200015d <dbl>, ...
```

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 1,381 columns of data, one for each recorded variable. You'll only need two 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.[3] This allows us to simply type in variable names separated by commas.

For this example code I'll look at two variables similar to those I had you pick from: trust in news media (In general, how much trust and confidence do you have in the news media when it comes to reporting the news fully, accurately, and fairly?) and perceived threat to the media (How concerned are you that some people in the government today might want to undermine the news media's ability to serve as a check on governmental power?). Their variable IDs are V201377 and V201376 respectively. Since I'll probably forget which ID is which, I'll give the columns more meaningful names as I select them.

[3] 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.

```
my_data <- timeseries_2020 |>
  select(trust = V201377,
         threat = V201376) |>
  haven::zap_labels()
```

I did something else there as well: `haven::zap_labels()`. Even though when you look at the data it's a bunch of numbers, it's actually a special format called "haven-labelled", meaning there is some extra info stored about the numbers behind-the-scenes. That can be useful to have, but it actually interferes with some of the numeric filtering and mutating we need to do momentarily, so the `zap_labels()` function drops that label information and turns the data into plain old numbers.

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

```
my_data
```

```
# A tibble: 8,280 x 2
   trust threat
   <dbl>  <dbl>
 1     1      1
 2     1      1
 3     3      5
 4     4      3
 5     2      4
 6     4      4
 7     2      5
 8     1      1
 9     3      2
10     3      2
# i 8,270 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. You can see all the unique values recorded for a column in the data like so:

```
unique(my_data$trust)
```

```
[1]  1  3  4  2  5 -9 -8
```

The negatives are 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.[4]

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 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.[5]

```
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] 8280
```

[4] 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`.

[5] 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(trust %in% 1:5,
         threat %in% 1:5)
```

```
nrow(my_data_complete)
```

```
[1] 8211
```

After filtering to keep only rows with complete data, we're left with 8,211 valid responses.

Recoding values

Now that we have selected our columns and filtered out missing/invalid responses, the last thing to do is recode values so they all mean what we want them to mean.

Notice that valid responses for the "undermine the news media" threat item are 1 (not at all concerned) through 5 (extremely concerned). I want higher scores on that question to indicate greater perceived threat, so that's fine.

For the "trust in news media" question, responses are 1 (none) though 5 (a great deal). But I'm thinking of the psychological construct as *dis*trust rather than trust, so I want higher scores to indicate more distrust. The solution is simple: subtract the participant's answer from 6 (one more than the maximum score) so that a response of 1 becomes a 5 (the most distrust), 2 becomes 4, 3 stays 3, 4 becomes 2 and 5 becomes 1 (the least distrust).

```
my_data_complete <- my_data_complete |>
  mutate(distrust = 6 - trust)
```

Now I have my two variables, perceived threat to the news media, and (reverse-coded) distrust of the news media, for each of the 8,211 participants with complete data. We're ready to start exploring the data.

Start examining the data

Descriptive statistics

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

There are many ways of doing this, but for now I'll just use the `mean()` and `sd()` functions. I can refer to a particular column in a data.frame using the `$` operator, i.e. `my_data$threat` and so on.

```r
mean(my_data_complete$threat)
```

```
[1] 3.446718
```

```r
sd(my_data_complete$threat)
```

```
[1] 1.363856
```

```r
mean(my_data_complete$distrust)
```

```
[1] 3.54086
```

```r
sd(my_data_complete$distrust)
```

```
[1] 1.208428
```

Visualizing distributions

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.[6] 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 threat item. I pipe the data into the `ggplot()` function, specifying that I want the `threat` 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.

[6] 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.

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

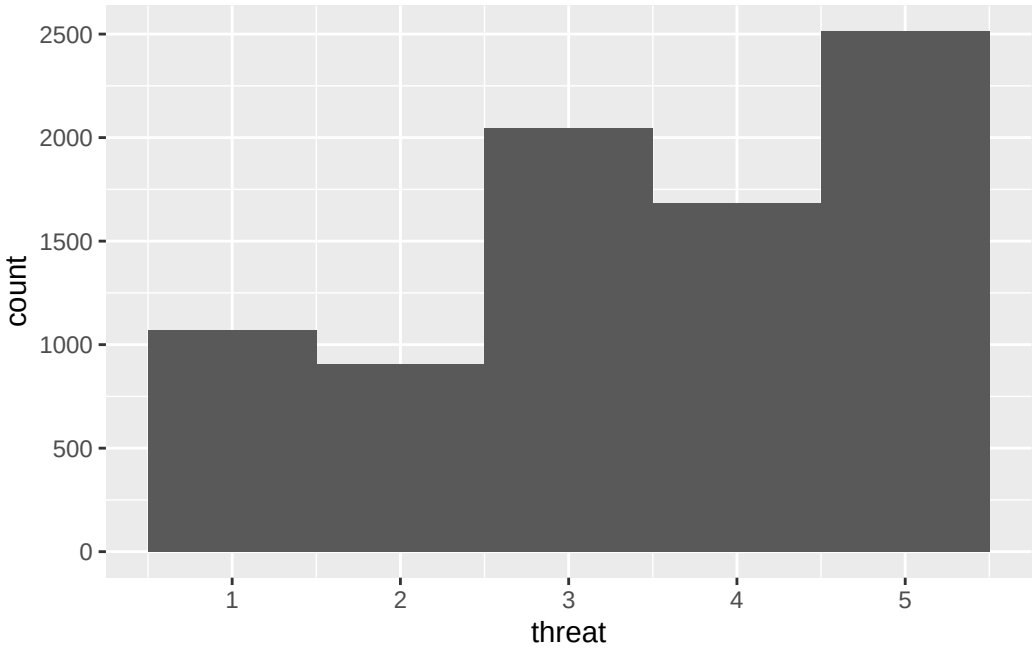


Figure 1: Histogram of responses to perceived threat to news media question

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.

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

I'll also customize the "breaks" and "labels" on the x-axis (to show the verbal scale responses rather than just numeric codes), and the labels for the x and y axes.

```r
my_data_complete |>
  ggplot(aes(x = threat)) +
  geom_histogram(binwidth = 1, color = "white") +
```

```
  scale_x_continuous(breaks = 1:5,
                     labels = c(" Not at all", "A little", "Moderately", "Very", "Extremely")
  labs(x = "How concerned are you that some people in the government today might want to\nunc
       y = "Number of responses") +
  theme_apa
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <80>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media’s ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media’s ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media’s ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media’s ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media’s ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media’s ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media’s ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media’s ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media’s ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media’s ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <99>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <99>

Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <e2>

Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <80>

Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'undermine the news media's ability to serve as a check
on governmental power?' in 'mbcsToSbcs': dot substituted for <99>
```

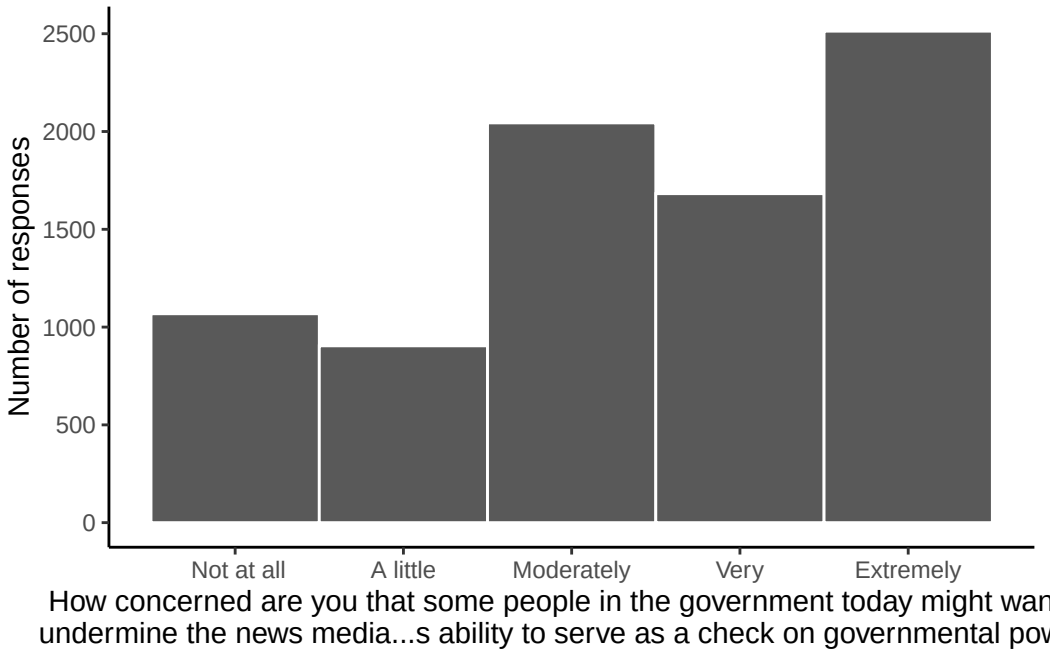


Figure 2: Histogram of responses to perceived threat to news media question

Here's a histogram of the distrust item.

```
my_data_complete |>
  ggplot(aes(x = distrust)) +
  geom_histogram(binwidth = 1, color = "white") +
  scale_x_continuous(breaks = 1:5,
                     labels = c("A great deal", "A lot", "A moderate amount", "A little", "No
  labs(x = "In general, how much trust and confidence do you have in the news media \n when i
       y = "Number of responses") +
  theme_apa
```

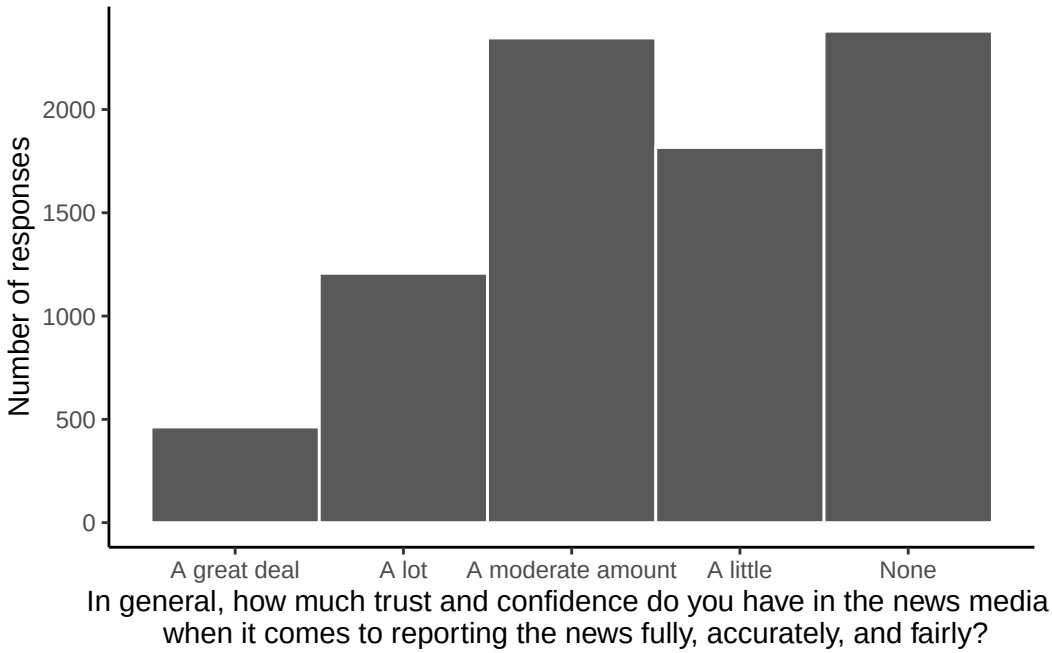


Figure 3: Histogram of responses to "distrust" question

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, thought about how to interpret your finding, and be ready to present and write up your project.

Goals

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

Correlation

Running with my example from last week, I made a data.frame with just the two variables I needed; filtered the data down to complete, valid responses; and recoded the trust item so higher scores indicate greater distrust. To refresh your memory, here's the setup and data preparation pipeline from start to finish:

```
library(tidyverse)
library(anesr)
data(timeseries_2020)

my_data_complete <- timeseries_2020 |>
  select(trust = V201377,
         threat = V201376) |>
  haven::zap_labels() |>
  filter(if_all(everything(), ~ . >= 0)) |>
  mutate(distrust = 6 - trust)
```

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.

If you need to refresh your memory from a past statistics class, refer to the correlation statistic Appendix.

Computing a correlation

The correlation between two variables can be found using the `cor()` function.

```
cor(x = my_data_complete$threat,
    y = my_data_complete$distrust)
```

```
[1] -0.4955168
```

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$threat,
    y = my_data_complete$distrust,
    use = "pairwise.complete.obs")
```

```
[1] -0.4955168
```

The `cor.test()` function goes further than `cor()`, giving you the p-value necessary for determining statistical significance[7] and some other information about the correlation.

```
cor.test(x = my_data_complete$threat,
         y = my_data_complete$distrust)
```

[7] 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.

```
    Pearson's product-moment correlation

data:  my_data_complete$threat and my_data_complete$distrust
t = -51.687, df = 8209, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.5116630 -0.4790208
sample estimates:
       cor
-0.4955168
```

Visualizing a correlation

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

```
my_data_complete |>
  ggplot(aes(x = threat, y = distrust)) +
  geom_point(position = position_jitter(width = 0.45, height = 0.45, seed = 1),
             alpha = 0.1) +
  geom_smooth(method = "lm") +
  scale_x_continuous(breaks = 1:5,
                     labels = c(" Not at all", "A little", "Moderately", "Very", "Extremely")
  scale_y_continuous(breaks = 1:5,
                     labels = c("A great deal", "A lot", "A moderate amount", "A little", "No
  labs(x = "Concern about government undermining the news media",
       y = "Trust in news media") +
  theme(panel.background = element_blank(),
        axis.line = element_line())
```

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

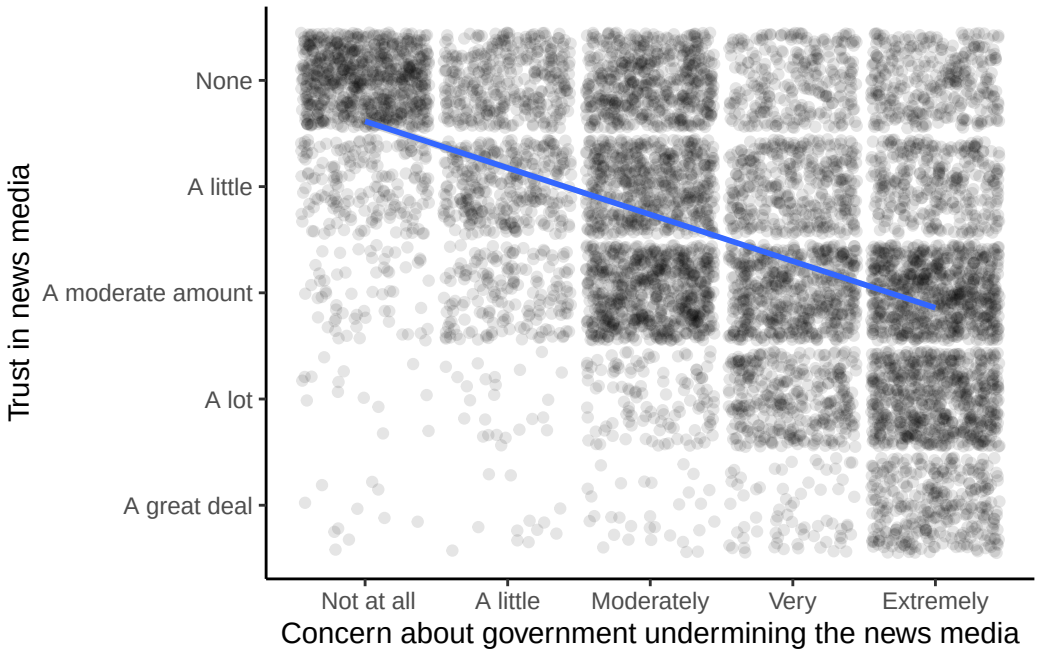


Figure 4: Scatterplot (with jitter) of perceived threat and distrust

Most of that will look familiar from the previous plots we made. The main difference is that instead of making a histogram I'm making a scatterplot, for which the "geometry" is points rather than histogram bars. Therefore I use `geom_point()` rather than `geom_histogram()` for the geometry layer. One new element is the `position = position_jitter()` part inside `geom_point()`. Its purpose is to add some random noise to each individual data point, moving it to the left or right a little bit along the x-axis and up or down a little along the y-axis. This is helpful here since there are many data points but only 5 possible answers along each axis. Try making the graph without including the jitter; it'll just look like a grid of 25 points. Any pattern in the data will be impossible to see. It may seem counterintuitive to change the data by adding randomness, but for the purposes of the visualization, doing so actually makes any patterns easier to see.

Another new element is an additional geometry layer: `geom_smooth()`. That adds a best fit line to the scatterplot. There are different ways of 'smoothing' the relationship between the variables. By default, the function computes a complex, nonlinear relationship and so the line will not be quite straight. For our purposes we want a simple, linear line of best fit, so it is necessary to specify `method = "lm"` to general a linear model.

Interpreting your findings

Remember, a correlation quantifies the general quantitative relationship between two sets of numbers; people's answers to your perceived threat question are associated with their answer to the distrust question to the extent indicated by your correlation coefficient. To fully and fairly interpret this for your presentation and report you'll need to consider a number of things:

- What features of how the questions were asked might have affected people's answers?
- What, if anything, about the context in which the questions were asked might have influenced people's answers?
- How does it fit with the previous research you read about, and with your intuition about how the variables should be related?

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 report. 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 research 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 is 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…").

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 the general research question; summary of relevant research; and hypothesis)
- Method (a description of your variables, 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 scores out of 100 for your presentation and report. See Appendix for general qualitative criteria which will be assessed in the context of the expectations detailed above.

Project 2: ANOVA

Lab 6: Project planning

In this lab, you will start your second project: conducting an ANOVA interaction analysis using the same ANES data as before. In this project you'll investigate how political partisanship and life experiences might interact to predict social trust.

Goals

- Understand the purpose of ANOVA
- Identify variables for your analysis
- Search the literature to find relevant research
- Articulate your hypotheses

Project overview

Understanding ANOVA

ANOVA stands for ANalysis Of VAriance. It is a statistical procedure which quantifies the relative contribution of difference sources of variability (variability between groups and variability within groups) to understand whether there are differences *between* groups over and above what would be expected by chance (i.e, based on the random variability *within* the groups).

ANOVA will always have a single, continuous dependent variable (DV). ANOVA can have any number of independent variables (IV; also called factors), each with any number of conditions (also called levels). The design will be described as something like "a 3x5 ANOVA", which would mean that there were 3 factors with 5 levels each. For this project, your design will be a 2x2 ANOVA, meaning 2 factors, each with 2 levels.

Quasi-independent variables

Researchers often "manipulate" at least one of the IVs in a study, making it an "experiment." In this context, the name "independent variable" implies that the researcher randomly determines which experimental group a participant will be in; therefore, the manipulated variable is

theoretically "independent" of other variables in the study. The ANES, however, consists only of survey data; there is no experimental random assignment.[8] Categorical variables that are not randomly assigned are called *quasi*-independent variables; they can still be used to look for differences between groups of participants, but since participants' membership of a group is an existing characteristic of theirs rather than something randomly determined, it isn't truly "independent" of other variables.

It may sound like a limitation, but there are good reasons to use quasi-IVs in research. It isn't always possible or ethical to randomly assign group membership, yet there is still value in comparing existing groups. As an example, a participant's age can't be randomly assigned, but it is still often useful to compare different age groups to look for differences on some dependent variable. That said, there are important implications for causal inference that you should consider when it comes time to interpret your findings.

Interaction

The value of a factorial ANOVA design is in revealing the "interaction" between variables: does the effect of one (quasi)-IV depend on another (quasi)-IV? For this project, we will stick with the idea of trust: it will be the DV. One of the quasi-IVs will be an indicator of political ideology/partisanship. It makes sense that some types of social trust could depend on your political preferences (or something related, like whether your preferred political party is currently "winning"). But political affiliation isn't the only thing that could be associated with trust. Maybe one's life experiences also play a role. Having had, or not had, some particular experience might be associated with expressing more or less trust. But crucially, maybe the two variables *interact*: maybe the relationship (if any) between partisanship and trust depends on what kind of experience a person has had (or not had). It could be that an experience affects trust among people who lean one way but not people who lean the other.

Step 1. Examine the data and identify your variables

Dependent variable

You dependent variable should be continuous (measured on a numeric scale). For this project, we will stick with the idea of trust.

- How often can you trust the federal government in Washington to do what is right? (V201233)
- How many of the people running the government are corrupt? (V201236)

[8] That's not *quite* true. Some elements of the survey are randomized, such as the order of certain questions or answer options. And the ANES often tests different versions of questions before including them in the full survey; answers to different versions of questions can be compared to check whether seemingly minor differences to wording substantively affect people's answers.

- Generally speaking, how often can you trust other people? (V201237)
- In general, how much trust and confidence do you have in the news media when it comes to reporting the news fully, accurately, and fairly? (V201377)

(Quasi)independent variables

The first of your quasi-independent variables will be some indication of participants' political preferences.[9]

- What political party are you registered with, if any? (V201018)
- Who did you vote for? (V202073)
- If you had to choose, would you consider yourself a liberal or a conservative? (V201201)

Your second quasi-independent variable will be one of the "life experiences" that the ANES survey asks about:

- Do you personally know someone who moved to the U.S. from another country, or not? (V202561)
- Do you currently owe money on student loans, or not? (V202562)
- Have you ever received food stamps or another form of public assistance, or not? (V202563)
- Do you have a pension or a retirement account, such as an IRA, 401k, or similar, or not? (V202564)
- Do you regularly choose products because they are made in America, or not? (V202565)
- Have you displayed an American flag on your house or in your yard in the past year, or not? (V202566)
- Have you gone hunting or fishing in the past year, or not? (V202567)
- Have you used public transportation in the past year, or not? (V202568)
- During the past 12 months, were you or any of your family members stopped or questioned by a police officer, or did this not happen in the past 12 months? (V202456)
- Have you ever been arrested, or has that never happened to you? (V202457)
- During the past 12 months, have you contacted or tried to contact a member of the U.S. Senate or U.S. House of Representatives, or have you not done this in the past 12 months? (V202030)
- During the past 12 months, have you joined in a protest march, rally, or demonstration, or have you not done this in the past 12 months? (V202025)
- During the past 12 months, have you ever gotten into a political argument with someone, or have you not done this in the past 12 months? (V202024)
- Many people say they have less time these days to do volunteer work. What about you, were you able to devote any time to volunteer work in the past 12 months or did you not do so? (V202033)

[9] There's a whole literature about ideology and partisanship and I don't expect you to get into the weeds with it, but I expect you'll have some thoughts about the various strengths/limitations of these few questions.

Step 2. Read relevant research

Since you choice is more open-ended, I can't point you towards a particular paper like I did for the previous project. You'll have to think about your constructs and see if you can find relevant research. In particular, what psychological construct do you think your "life experience" variable might reflect or relate to?

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.[10]

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 search for `psycinfo` in a CLIO quicksearch). PsycInfo gives you the ability to do more focused searching than Google Scholar.

- You can add many keywords and 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).

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

Step 3. Articulate your design and hypothesis

As before, you should be able to state your:

- **Operational definitions** (that is, the specific questions that participants were asked, per the codebook)
- The **constructs** that those operational definitions measure
- Your **hypotheses**

In the context of a 2x2 ANOVA design, you will have three different hypotheses. For each IV, you will hypothesize the existence (or not) of a main effect; that is, do you expect to see a difference between the different categories of that IV by itself (ignoring differences on the other IV). In addition, you will have a hypothesis about the "interaction" of the two IVs; that is, do you expect them to have a *combined* effect? In more technical terms, do you expect that the effect of one IV depends on the level of the other IV?

Lab 7: Data preparation & analysis

You will start this lab with your variables determined. By the end, you will have prepared the data in R and computed the required descriptive statistics and the ANOVA itself.

Goals

- Select required variables
- Recode categorical variables
- Compute your ANOVA

Data preparation

Set up

As before, the first step is to prepare your R environment by loading required packages and loading the data. The will be exactly the same as with the previous project.

```
library(anesr)
library(tidyverse)
data(timeseries_2020)
```

Select variables

For this example, I'm going to use the "trust the media" question–the same one I used for the correlation example–as my dependent variable.[11] For the partisanship quasi-IV, I'll use the party registration question. And for the life experience, I'm going to use one I didn't let you choose from (because not many people said yes, and… well, you'll see): "Have you ever been bitten by a shark, or not?" (V202569).

[11] In general, how much trust and confidence do you have in the news media when it comes to reporting the news fully, accurately, and fairly?

```r
my_data_raw <- timeseries_2020 |>
  haven::zap_labels() |>
  select(trust = V201377,
         party = V201018,
         experience = V202569)
```

Recode variables

In the raw data responses to the categorical quasi-IVs are coded as numbers: 1s and 2s for "Democrat"/"Republican" and "yes"/"no", and some other numeric codes for other responses or invalid/missing answers. You'll need to check the codebook to determine exactly what all the numbers mean. Our first step is to "recode" these raw numbers into more meaningful answers, keeping only the ones we need and turning other answers, or negative numbers representing missing data, into `NA`–R's representation of missing data.

We'll use `dplyr`'s `mutate()` function, which creates new variables or changes (*mutates*) existing ones. To achieve the desired result we'll use `dplyr`'s `case_when()` function, which allows us to specify a condition and the value to assign if that condition is met. So for example `party == 1 ~ "Democratic"` can be read as "if the value of `party` is `1` then assign the value *'Democratic'*." Since the original variable is named "party" and I'm using that name for the mutated variable, the column will be changed in place. Note that I only specify the numeric codes that I want to retain for my analysis; answers other than 1 or 2 will become `NA` since I don't specify otherwise.

```r
my_data <- my_data_raw |>
  filter(if_all(everything(), ~ . > 0)) |>
  mutate(party = case_when(
    party == 1 ~ "Democratic",
    party == 2 ~ "Republican"
  )) |>
  mutate(experience = case_when(
    experience == 1 ~ "Yes",
    experience == 2 ~ "No"
  ))
```

Analysis

Descriptive statistics

Before computing the ANOVA itself, the first step is to compute descriptive statistics–the mean and standard deviation–for each grouping of participants in our data. With a 2x2 design there are a few different ways to group up the data, corresponding to the 3 hypotheses we aim to test.

First we'll find the "marginal means". These are relevant to testing and interpreting the "main effects". Finding this kind of mean is slightly more complicated than when we just needed to find the mean of each column in a data.frame like we did for the correlation project. This time, we want the mean from one column broken into groups based on the value of a different column; mean trust by party affiliation, and mean trust by life experience.

As usual there are a lot of ways of doing this, but one of the most powerful is `dplyr`'s `summarize()` function. It allows you to compute summary values using other functions, such as `mean(trust)` to compute the mean of the trust column, and you can specify a grouping variable using the `.by` argument. And you aren't limited to computing a single summary variable; here I also compute the standard deviation (using the `sd()` function) and number of observations in each group (using the `n()` function).

```
my_data |>
  summarize(mean = mean(trust),
            sd = sd(trust),
            n = n(),
            .by = party) |>
  drop_na()
```

```
# A tibble: 2 x 4
  party       mean    sd     n
  <chr>      <dbl> <dbl> <int>
1 Republican  1.75 0.980  1176
2 Democratic  3.14 1.08   1666
```

```
my_data |>
  summarize(mean = mean(trust),
            sd = sd(trust),
            n = n(),
            .by = experience)
```

```
# A tibble: 2 x 4
  experience  mean    sd     n
  <chr>      <dbl> <dbl> <int>
1 No          2.53  1.22  3763
2 Yes         2.32  1.04    22
```

In addition to those marginal means we need to find all 4 "cell means," a mean trust score for each of the possible combinations of the political party and life experience groups: Democrats who have been bitten by a shark; Democrats who haven't been bitten by a shark; Republicans who have been bitten by a shark; Republicans who haven't been bitten by a shark. This sounds like a lot of work, but thankfully it can be achieved with a very minor tweak to the `summarize()` approach we used before: we just supply both grouping variables to the `.by` argument at the same time, "collecting" the variable names together with the `c()` function.

Thinking ahead, I'm going to want to make a graph showing these 4 means, and I'd like the graph to have "error bars" representing the confidence interval for each mean. There's no built-in function to compute a confidence interval, so I'm going to make my own.

```
ci <- function(x) {
  qt(0.975, df = length(x) - 1) * sqrt( var(x) / length(x))
}
```

Now I'm ready to compute the mean, SD, number of observations, and confidence interval for each of the 4 groups.

```
my_data_summary <- my_data |>
  summarise(mean = mean(trust),
            sd = sd(trust),
            n = n(),
            ci = ci(trust),
            .by = c(party, experience)) |>
  drop_na()

my_data_summary
```

```
# A tibble: 4 x 6
  party      experience  mean    sd     n     ci
  <chr>      <chr>      <dbl> <dbl> <int>  <dbl>
1 Republican No          1.75 0.981  1170 0.0563
2 Democratic No          3.14 1.08   1659 0.0522
3 Republican Yes         1.5  0.837     6 0.878
4 Democratic Yes         2.71 0.488     7 0.451
```

Note that I assigned this data summary to a new name, `my_data_summary`, so it becomes a new data.frame object in my environment. That's going to be useful later, because it is these summary statistics that I will use to make a graph of the results.

ANOVA

The `aov()` function computes an ANOVA. By itself, `aov()` doesn't output all the information we want to see; that's why I pipe it into the `summary()` function below.

The first argument to the `aov()` function is a formula, in the form `DV ~ IV1 * IV2`. The second argument is `data`, to which I supply the name of the data.frame containing my data; that's how the formula in which we name the columns can work, since supplying the data.frame to the `data` argument tells the function where to find those columns.[12]

```
aov(trust ~ party * experience, data = my_data) |>
  summary()
```

```
                   Df Sum Sq Mean Sq  F value Pr(>F)
party               1 1343.3  1343.3 1238.597 <2e-16 ***
experience          1    1.5     1.5    1.421  0.233
party:experience    1    0.1     0.1    0.098  0.754
Residuals        2838 3077.9     1.1
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
943 observations deleted due to missingness
```

[12] It would be more elegant if I could pipe the data into `aov()` and then pipe that into `summary()`, like this:

```
my_data |>
  aov(trust ~ party * experience) |>
  summary()
```

Unfortunately that won't work, because by default the pipe operator inserts the output of the previous line of code into the *first* argument of the next line. Since `aov()` is a base R function and the pipe is a more recent innovation, `aov()` wasn't designed with this in mind. If it was, `data` would be the first argument and the pipe would work. There is a way to make it work, by using the `_` special character, which R's pipe operator understands as a placeholder for the pipe's output, to specifically place the output into `aov()`'s `data` argument:

```
my_data |>
  aov(trust ~ party * experience, data = _) |>
  summary()
```

It's not very elegant in this case, which is why I didn't bother, but knowing about the `_` placeholder can be useful in general.

You should see three lines with your variable names, followed by a line for "Residuals" (which you can ignore). Those three lines are your three hypothesis tests: two main effects and the interaction. Each line shows the "degrees of freedom" ("Df"), F value, and p-value ("Pr(>F)"). (Sum Sq and Mean Sq don't need to be reported). If the p-value is less than 0.05, that test is "statistically significant."

Interpreting this can be difficult, so we'll follow this up in the next lab with some visualization which can make understanding the pattern of results easier.

Lab 8: Visualization & interpretation

By the end of the session you will have created a visualization of your 2x2 interaction, thought about how to interpret the findings, and be ready to present and write up your project.

Goals

- Visualize the interaction
- Interpret the findings

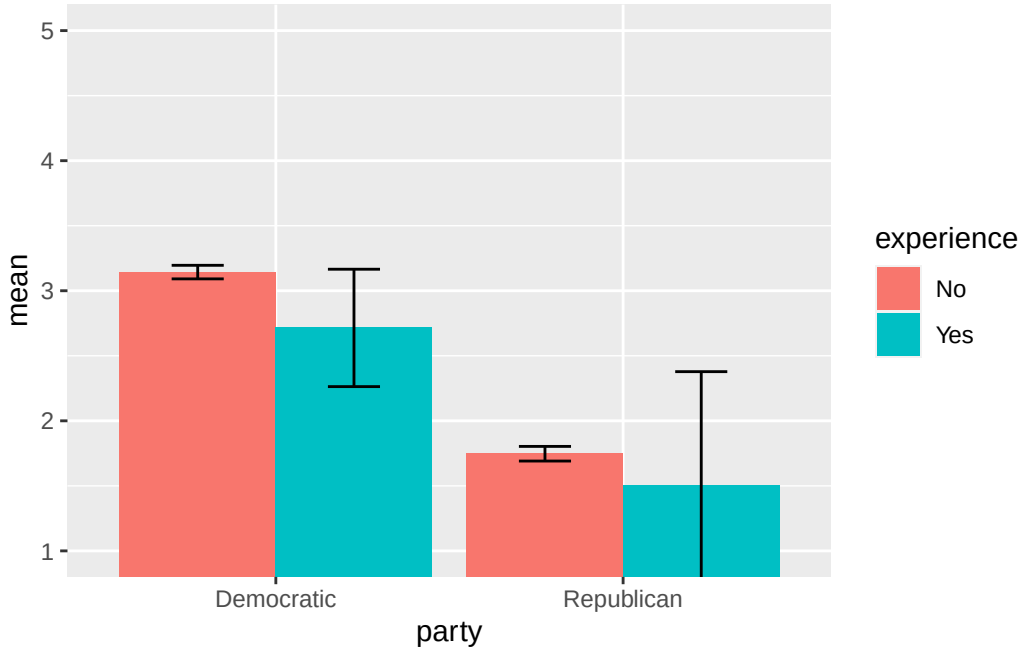
Visualizing ANOVA

In the example code for the previous lab session, I created a summary of descriptive statistics named `my_data_summary`. It included the average trust score for each of the 4 combinations of my 2 IVs, as well as a 95% confidence interval. I'm going to use that summary data.frame (rather than the full data) to create a graph of the interaction.

There are two main ways of visualizing the interaction: as a bar graph, or as a line graph. They show exactly the same information just in a slightly different way, and which you find to be better–i.e., more intuitive and effective for conveying your findings–is a matter of personal preference.

Bar graph

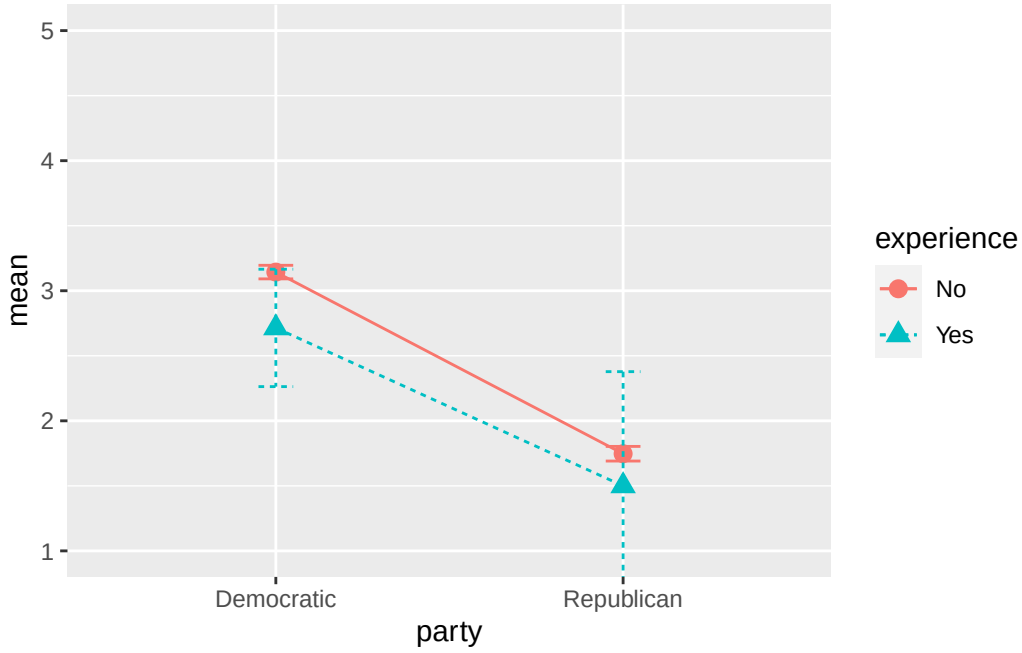
```
my_data_summary |>
  ggplot(aes(x = party, y = mean, fill = experience)) +
  geom_col(position = position_dodge()) +
  geom_errorbar(aes(ymax = mean + ci, ymin = mean - ci), position = position_dodge(width = 0
  coord_cartesian(ylim = c(1, 5))
```



Line graph

Here I use several aesthetic mappings for the "life experience" variable: the `color` of the lines and points, the `shape` of the points and `linetype` of line (solid or dashed) all differ. It's redundant to map the same variable to so many aesthetics, but the redundancy can be helpful. If the graph was printed in black and white, for example, it would still be possible to easily tell the groups apart.

```
my_data_summary |>
  ggplot(aes(x = party, y = mean, group = experience, color = experience, linetype = experie
  geom_point(size = 3) +
  geom_line() +
  geom_errorbar(aes(ymax = mean + ci, ymin = mean - ci), width = 0.1) +
  coord_cartesian(ylim = c(1, 5))
```



My graphs here are very rudimentary; remember, you should customize aspects of the theme and labels to better convey the data and conform to the usual APA style for figures.

Interpreting ANOVA

As a reminder, a 2x2 ANOVA tests 3 different hypotheses: 2 main effects and the interaction. The outcomes of all 3 are independent: you may find that both main effects are significant, or neither, or one main effect is significant but not the other. And the interaction may be significant (or not) regardless of the (non)significance of the main effects.

This can get quite complicated to think about, so try to break it down in the context of your design. Think about:

- The main effect of partisanship (do Democrats and Republicans differ in trust?)
- Main effect of life experience (do people who have and have not had the experience in question differ in trust?)
- Interaction: does the relationship between the life experience and trust *depend on* partisanship? I.e., is the relationship different for Democrats vs Republicans?

Your report should mention all three of the above issues.

Lab 9: 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

The guide to delivering an effective presentation is the same as last time. My main advice this time is to focus on making your interpretation of the main effects and interaction as clear and convincing as possible. As you've probably seen, it's tricky to get your head around, and even trickier to explain to someone else. The best presentations will make the logic of your interaction as easy to follow as possible.

Written report

You will again 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 the general research question; summary of relevant research; and hypothesis)
- Method (a description of your variables, 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

As before you will receive a scores out of 100 for your presentation and report. See Appendix for general qualitative criteria which will be assessed in the context of the expectations detailed above.

Project 3: Own design

Lab 10: Project planning

My hope is that at this point you are feeling a sense of accomplishment at what you have achieved through the projects so far, but that you're also feeling a bit limited by the constrained limited choice of variables/topics you had. In this lab we'll begin the final project: you will execute a project of your own design, with the freedom to choose any variables you like from the full data set.

Goals

- Decide on an analysis and identify the relevant variables
- Find some relevant research
- Articulate your hypotheses

Project overview

For this project you have free choice of design and constructs. You may look at the correlation(s) between two (or more) variables, or you may use an ANOVA design to look at differences between groups based on categorical variables. The process for getting started planning your design is the same as before.

Step 1. Decide on your constructs/operational definitions

The full ANES 2020 dataset contains more than 1,000 columns, each representing a different variable that was measured. To navigate this, see the codebook, or the Methodology Report which organizes things into "modules". Not all of these are equally interesting from a social-psychological perspective, but you will find many questions that pertain to fundamental social-psychological concepts. Here are some suggestions–constructs the ANES data measures in some way that you might consider from a social psychological perspective:

- Feeling thermometers (affect towards various political actors/groups)
- Political knowledge
- Media consumption
- Authoritarianism

- Stereotypes
- Policy preferences
- Conspiracy beliefs
- Religiosity
- Life experiences

Step 2. Find relevant research

You should search the literature (using Google Scholar or PsycINFO, as before) and find at least one published research paper that pertains to the constructs you want to look at.

Step 3. Articulate your hypotheses

As before, you should be able to describe:

- The social psychological constructs you are interested in
- The operational definitions used to measure those constructs in the ANES survey
- You hypothesis about how your constructs relate

Lab 11 & 12: Analysis, visualization & interpretation

Groups should have decided on what variables to use and what analysis is appropriate. By the end of these sessions you should have coded your analysis, interpreted the results, and be ready to present and write up your findings.

Data wrangling tips

Since everyone will be doing different things, Professor Brotherton work with groups individually to provide guidance on data preparation and analysis where help is needed. However, here are some hints about how to achieve common tasks.

Compute an average of several variables.

Some psychological measures consist of more than one question, and you need to compute the average of each participant's answers to all the relevant questions. `mutate()` and `rowMeans()` can be used.

```
my_data |>
  mutate(mean = rowMeans(across(all_of(scale_vars))))
```

Compute a sum score.

If you need to add scores across several questions, `rowSums()` can be used.

```
my_data |>
  mutate(sum = rowSums(across(all_of(scale_vars))))
```

Ntiles.

Occasionally you might like to split a continuous measure into "ntiles," meaning a number of roughly equally-sized groups. The `ntile()` function can be used to this.

```
my_data |>
  mutate(income_bracket = ntile(income, 2))
```

Arbitrary groups

```
my_data |>
  mutate(age_group = case_when(age < 30 ~ "Young", age >= 30 ~ "Old"))
```

Lab 13: Presentation & report

As for previous projects, 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

See again the general tips for delivering an effective presentation. My main advice this time is to remember that every group will likely be working on quite different social psychological topics; your audience will not be experts in the topic you have chosen. Focus on making sure you give a clear overview of your constructs, operational definitions, and hypotheses, all in the context of whatever relevant previous research you have discovered.

Written report

You will again 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 the general research question; summary of relevant research; and hypothesis)
- Method (a description of your variables, 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

As before you will receive a scores out of 100 for your presentation and report. See Appendix for general qualitative criteria which will be assessed in the context of the expectations detailed above.

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.

Once the project is up and running, click on `anes.R` in the bottom-right pane to open some analysis code.

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.

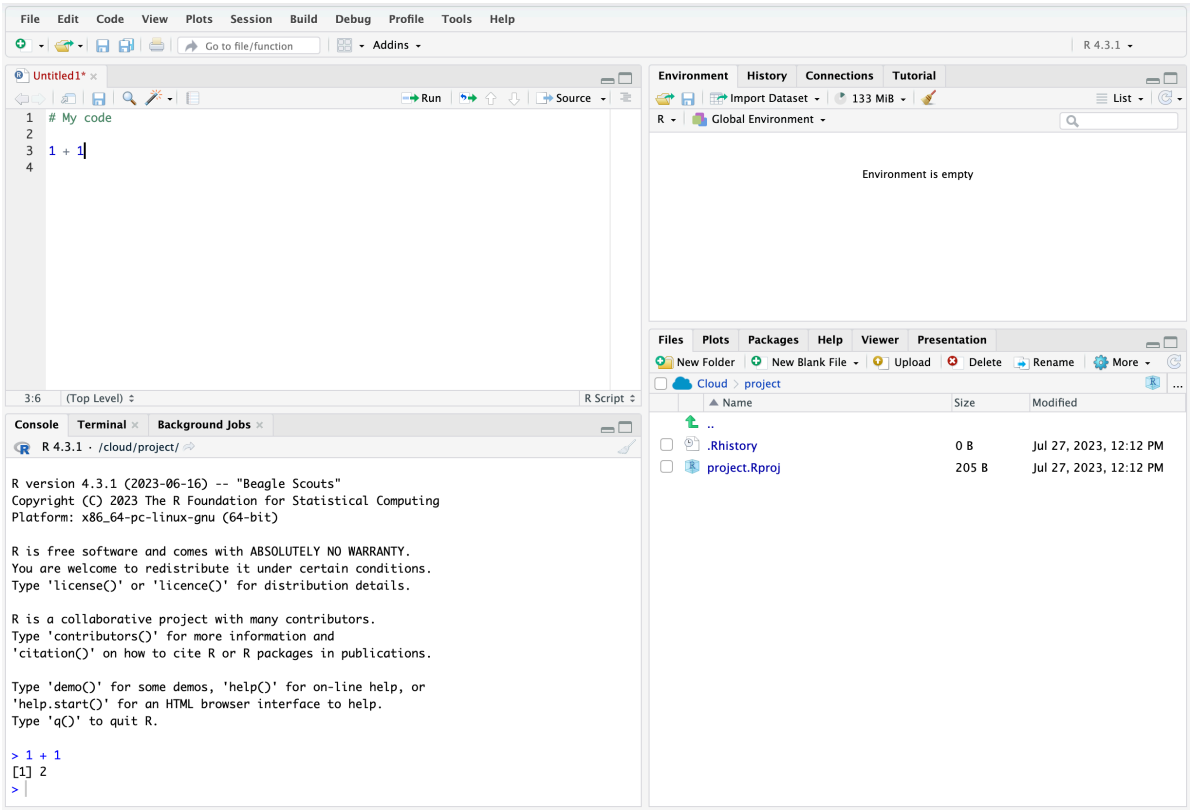
Fundamentals of R for data analysis

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:



- 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.

[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(original_numbers)
```

```
[1] 5.5
```

```
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.

Describing data

Once you have your data, you need some tools to describe and interpret it. Of course, you could just list the individual observations and try to form an intuition about the general outcome, but there are more formal means to determine whether the experimental manipulation had an effect. A statistical description summarizes the data in a way that permits interpretation.

The mean

The average, or mean, is a measure of typical performance; it summarizes all the scores and produces a single number which represents the most typical value. The basic formula for the mean of a set of scores is:

$$M = \frac{\Sigma X}{n}$$

In this equation, X refers to all the scores in the group, and n is the number of scores in the group. The symbol Σ instructs you to sum all the scores. A simple way of saying the formula in words is: Add up all the scores in the group and divide by the number of scores in that group.

Standard deviation

However, there is always variability in the scores in a group. The mean is a *central* value, but some scores fall below it and others above it. Therefore, researchers also need to describe the amount of variability in scores. This puts the mean in context, describing just how representative of all the scores it is. If there is high variability, scores are spread widely and the mean is relatively unrepresentative; if there is low variability, scores are clustered tightly and the mean is relatively representative.

A mathematical way of describing the amount of variability in a group of scores is to calculate the deviation of each score from the mean, square the deviations, and then sum the squared deviations. This quantity is called Sum of Squares (SS). One mathematical formula is:

$$SS = \Sigma(X - M)^2$$

Dividing SS by the number of scores in the group minus 1 produces a quantity called variance, which is represented by the symbol s^2. Variance is the average squared deviation. (Remember that to calculate an average, you add a set of scores and divide by n. Here we add a set of deviations and divide by n–1. We use n–1, rather than just n, because it is a necessary statistical adjustment to account for the fact that samples tend to underestimate variability.)

$$s^2 = \frac{\Sigma(X - M)^2}{n - 1}$$

Taking the square root of the variance produces another quantity, called standard deviation. It is represented mathematically by the symbol s, but in psychology papers you will most often see it represented by the letters SD.

$$SD = \sqrt{\frac{\Sigma(X - M)^2}{n - 1}}$$

While variance is the average squared deviation, SD is the average deviation in the original units (i.e. not squared). This is the most intuitive way to convey how much scores typically varied about the mean.

Correlation

When you measure two variables and wish to know if scores on one measure are related to scores on the other, you calculate the correlation coefficient. This quantifies the extent to which changes on one measure are related to changes on the other. For example, if higher scores on measure X are associated with higher scores on measure Y, there is a positive correlation. If higher scores on measure X are associated with lower scores on measure Y, there is a negative correlation. No correlation means that scores on X are unrelated to scores on Y.

Calculating the correlation coefficient

To calculate the correlation between two variables, you must first calculate the Sum Product, SP. The mathematical formula is:

$$SP = (X - M_X)(Y - M_Y)$$

Notice that $X - M_X$ and $Y - M_Y$ are deviation scores, just like we calculated for the standard deviation. Here we have two variables, X and Y, so the equation is telling us to calculate the deviation of each score from its respective mean. We then multiply each deviation for variable X by its counterpart deviation from variable Y. These are the "products," meaning multiplied deviation scores. Finally, the tells us to add up all those products, giving the "sum of products," SP.

Once we have calculated SP, the correlation coefficient, symbolized by r is calculated using the following equation:

$$r = \frac{SP}{\sqrt{SS_X SS_Y}}$$

Here, SS_X and SS_Y are the Sums of Squares for each variable. Multiplying them and taking the square root gets us a measure of the variability in X and Y separately. The numerator, SP, represents the covariability of X and Y. So the equation results in covariability as a proportion of all variability. It can range from -1, meaning a perfect negative correlation, to 0, meaning no correlation at all, to $+1$, meaning a perfect positive correlation.

Effect size for correlation

The correlation coefficient is a measure of effect size. It's absolute value can range from 0 to 1.

You may see some "rules of thumb" about interpreting the "effect size" of correlations in psychology. Cohen (1977) proposed that correlations of less than around ± 0.30 should be considered weak; around ± 0.30 to ± 0.70 considered moderate; and greater than around ± 0.70 considered large.

However, more recent researchers have proposed more nuanced and empirically-grounded interpretations. Funder and Ozer (2019) proposed the following:

r	Description
0.05	Very small for the explanation of single events but potentially consequential in longer run
0.10	Still small at the level of single events but potentially more ultimately consequential
0.20	Medium effect of some explanatory and practical use even in the short run
0.30	Large effect that is potentially powerful in both the short and the long run
0.40	A very large effect in the context of psychological research; likely to be a gross overestimate

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. Clearly articulates the problem, purpose, methods, results, and interpretation. There is evidence of critical thought, and the work goes beyond the assignment requirements in terms of analysis or presentation, demonstrating a sophisticated understanding of the concepts and techniques used. The presentation/report is free of errors and is clearly and professionally executed
A	90-97	Excellent work. Clearly articulates the problem, purpose, methods, and results. There is evidence of critical thought, demonstrating a strong understanding of the concepts and techniques used. The presentation/report is virtually free of errors and is clearly and professionally executed
B	80-89	Above average work. Articulates the problem, purpose, methods, and results. There is some evidence of critical thought. Demonstrates a good understanding of the concepts and techniques used. There are minor errors or lack of clarity in some aspects, but the presentation/report is generally clear and professional.

Grade	Point range	Description
C	70-79	Satisfactory work. Articulates the problem, purpose, methods, and results but may lack clarity or detail. There is minimal evidence of critical thought. Demonstrates an acceptable understanding of the concepts and techniques used. There are noticeable errors, and the presentation/writing could be improved.
D	60-69	Below average work. Does not clearly articulate the problem, purpose, methods, results, or interpretation. There is little to no evidence of critical thought. Demonstrates a minimal understanding of the concepts and techniques used. There are significant errors, and the presentation/writing is unclear.
F	<60	Unsatisfactory work. Does not articulate the problem, purpose, methods, results, or interpretation. There is no evidence of critical thought. Demonstrates a lack of understanding of the concepts and techniques used. There are many errors, and the presentation/writing is poor.