

PSYCH 10 Final Project · Fall 2026

Week 6 · M5 · A closer look at your data

Full name (in CAPITAL LETTERS) _____ **SUNet ID** (the part before @stanford.edu) _____ **Section** (e.g., TUES 9:30) _____

Please write legibly so the teaching team can log your attendance: if we can't read your name, we can't give you credit.

Please hand this sheet in to your TA at the end of section (each student should hand in their own copy). A copy is on the course website (psych10.github.io).

Deliverables

Each group should submit two files on Gradescope (the link is on Canvas): the group's `M5-closer-look.Rmd`, from the `milestones/` folder, and the knit PDF. One member of the group should upload both files and add the other members to the submission. By Sunday, your group's shared folder should also hold the updated `data/data-dictionary.md`, `R/prepare-data.R`, and `R/checks.R`, and the processed data file in `data/processed/`.

Steps

Your group will do most of these steps in section and finish the rest together before Sunday. Steps 5 through 7, which prepare the data, are where most groups spend the most time. The `.Rmd` has a place for every step on this sheet.

1. **Get the latest version of your folder.** Download your group's shared folder from Google Drive, so that you have everyone's latest changes, and download `M5-closer-look.Rmd` from the course website into the `milestones/` folder. Decide who will upload the changed files at the end of section.
2. **Copy in your two questions.** Copy your two questions from M4 into the `.Rmd`, along with any changes your group made in response to your TA's suggestions.
3. **Write the data dictionary.** In `data/data-dictionary.md`, fill in one row for every variable your two questions use: its name, its type (quantitative or categorical), its units or levels, **how it was measured** (the survey question, the instrument, or the record it came from), its allowed range, and the code the raw file uses for missing values. Please take "how measured" from the documentation that came with the data (a README, a codebook, the dataset's download page, or the help page for an R package dataset, such as `?gapminder`), not from an AI tool and not from memory.
4. **Look at each variable on its own.** Make a histogram of each quantitative variable and a bar graph of each categorical variable. Under each plot, write a sentence or two on the range that holds most of the values and on anything odd: a value that cannot be right (an age of 999), a missing-value code such as `-99`, a spike at a single value, or a group with only a few cases.
5. **Prepare the data in a script.** The data dictionary from step 3 describes the form each variable must have for your analysis. The script `R/prepare-data.R` turns the raw file into that form, one change at a time. Open the script. In block 1, replace `your-file.csv` with the name of your raw file. Make every other change in blocks 2 and 3, and please do not edit the raw file by hand or in Excel. For each column, compare the raw data with its dictionary row, and add code for every difference. The usual changes:
 - **Columns.** Keep only the columns your two questions need. Rename any column whose name does not describe its contents (`q12` to `smoker`).
 - **Missing values.** Replace the missing-value codes you recorded in the dictionary (`-99`, `999`, an empty cell) with `NA`.
 - **Values.** Recode values into the dictionary's form: number codes into labels (`1` into `"yes"` and `2` into `"no"`, following the codebook), several small groups into one larger group, one unit into another (inches into centimeters), or a new column computed from existing columns (age from birth year).
 - **Rows.** If one row is not one case, reshape the data (a table with one column per year becomes one row per country and year).
 - **Dropped cases.** Drop a case only for a reason you state in the `.Rmd`, such as an impossible value or a case outside the population named in your claim, and never because the relationship looks stronger after the drop. For each rule, have the script print the number of rows it removed, as in the template's example.
 - **A variable you do not have.** If one of your questions needs a variable that is not in your dataset, do not look for another dataset. Instead, describe in the `.Rmd` the variable you would want: what it would measure, its units or levels, and how it would be measured (the survey question, the instrument, or the record it would come from). Question 1 and the M6 model use only the variables already in your dataset.

In the `.Rmd`, list every change in the order the script makes them, with the reason for each. You may use an AI tool for this code; see the section “Using AI tools on this milestone”.

6. **Save the processed data.** The last lines of `R/prepare-data.R` write the cleaned data to `processed-data.csv` in `data/processed/`. Run the whole script from the `.Rmd` with `source(here("R", "prepare-data.R"))` and check that the file is in `data/processed/`. From this step on, every code chunk should read from `data/processed/`, not from `data/raw/`, and every later milestone starts from this file.
7. **Write three checks.** In `R/checks.R`, write at least three statements that you believe are true of the cleaned data, each as a `stopifnot()` line: for example, the number of rows, the allowed range of a quantitative variable, or the full set of levels for a categorical variable. `prepare-data.R` runs this file just before it saves the processed data, so a failing check stops the save; please rerun `prepare-data.R` after you add your checks. If a check fails, find out why, and say in the `.Rmd` which one was mistaken: your belief about the data, or your code.
8. **Compute summary statistics.** For each variable in your two questions, run `favstats()` if it is quantitative or `tally()` if it is categorical. For question 1, if the explanatory variable is categorical, also summarize the outcome within each group, for example `favstats(LifeExp ~ continent, data = processed)`. Under each table, write one sentence that names the typical value and the spread (for a quantitative variable) or the largest and smallest groups (for a categorical variable); for the summary by group, name the gap between the group means in the units of the outcome.
9. **Plot question 1.** Make a plot of the outcome against the explanatory variable, with the outcome on the y-axis. Choose the kind of plot from the variables’ types: a scatter plot for two quantitative variables, or a box plot or a jitter plot by group for a categorical explanatory variable. Then describe the plot in two or three sentences: the direction of the relationship (whether the outcome rises or falls with the explanatory variable, or which group is higher), its rough size in the units of the outcome, and the spread of the values around that trend (how tightly the points sit around a line, or the amount of overlap between the boxes).
10. **Report what you have found so far for question 2.** If the other variable your second question needs is already in your data, make one plot or table that shows all three variables, for example by coloring the points by group or by giving each group its own panel (`gf_facet_grid()`), and say whether the question 1 relationship is the same within every group or differs between groups in its direction or size. If that variable is not in your dataset, say what you would expect such a plot to show if you had the variable you described in step 5.
11. **Write what you have learned so far.** In a few sentences, say what your plots and numbers suggest about each question, and which relationship your group wants to test in M6, when you will fit a model and measure the strength of that relationship.
12. **Knit to PDF in a fresh session** (Session → Restart R, then Knit → Knit to PDF). One member should submit the `.Rmd` and the PDF on Gradescope and add the other members to the submission.
13. **Upload your changed files to the shared folder:** the data dictionary, `R/prepare-data.R`, `R/checks.R`, the processed data file, and the M5 `.Rmd`.

Using AI tools on this milestone

The course encourages you to use AI tools on the project, and learning to use them well is one of the project’s goals. This week, use an AI tool to help you write the code in `prepare-data.R` (keeping and renaming columns, recoding, reshaping, and dropping cases) and the code for your plots. To get code you can use, give the tool your plan from M4, the `str()` output for your raw data, and a plain-language description of each change, one change at a time. For example: “In my data, `q12` is 1 for smokers and 2 for non-smokers. Write R code that renames it `smoker` and recodes it to “yes” and “no”.” Paste the code you get into `prepare-data.R`, run it, and then run `tally()` or `str()` on the new variable to confirm that the change matches your description. Every member of the group should be able to explain the purpose of each line in the script. Two parts of this milestone should come from your group alone: please take “how measured” in the data dictionary from the dataset’s documentation, and please write the descriptions of your plots and numbers in your own words.

How it’s graded (2 points)

	Points
The data dictionary: a row for every variable in your questions, with “how measured” from the documentation	0.3
A histogram or bar graph of each variable on its own, with a note on anything odd	0.2
The recoding in <code>prepare-data.R</code> : each change listed in the <code>.Rmd</code> with its reason, tied to your M4 plan	0.2
The processed data written to <code>data/processed/</code> by the script, and three passing checks in <code>checks.R</code>	0.3
Summary statistics for each variable (and by group for question 1, if it has groups), each with a sentence	0.3
The question 1 plot, of the right kind for the variables’ types, with a description of its direction, rough size, and spread	0.4
Question 2 so far: a plot or table, or the variable you would want and its expected pattern; what to test in M6	0.3