Bootcamp registration data: Gathering & cleaning

About

This page documents the process of gathering and cleaning data provided by Bootcamp 2026 registrants. It is supplementary material for the Quarto II workshop.

Set up

You might put any code that you need to run before you gather and clean your data here.

```
library(googlesheets4)
library(readr)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

Gather Google Forms data

```
googlesheets4::gs4_auth(email = Sys.getenv("GOOGLE_PSU"))

registered <- googlesheets4::read_sheet(Sys.getenv("BOOTCAMP_2026_REG_ID"),
                                        sheet = "Form Responses 1")
```

```
v Reading from "Open Scholarship Bootcamp 2026: Registration (Responses)".

v Range ''Form Responses 1''.
```

Inspect

It's not a bad idea to peek at the data you downloaded.

```
dim(registered)
```

```
[1] 94  8
```

```
names(registered)
```

```
[1] "Timestamp"
[2] "Email Address"
[3] "Which days of the bootcamp will you attend?"
[4] "What is your name?"
[5] "What is your department or unit?"
[6] "What is your current position?"
[7] "Any comments?"
[8] "drop-out"
```

Save raw

If you have space, it's a good idea to store date/time-stamped versions of the file you download.

```
save_path <- "data/raw_data"
fn <- "bootcamp-2026-registrations"
now_str <- format(Sys.time(), format = "%Y-%m-%d-%H%M%S")
file_path <- file.path(save_path, paste0(fn, "-", now_str, ".csv"))

readr::write_csv(registered, file = file_path)
```

And even confirm that you saved it.

```
list.files(save_path, pattern = ".csv")
```

```
 [1] "bootcamp-2026-registrations-2026-05-07-102539.csv"
 [2] "bootcamp-2026-registrations-2026-05-07-110716.csv"
 [3] "bootcamp-2026-registrations-2026-05-07-113513.csv"
 [4] "bootcamp-2026-registrations-2026-05-07-113947.csv"
 [5] "bootcamp-2026-registrations-2026-05-07-114118.csv"
```

```
 [6] "bootcamp-2026-registrations-2026-05-07-114333.csv"
 [7] "bootcamp-2026-registrations-2026-05-07-115132.csv"
 [8] "bootcamp-2026-registrations-2026-05-07-115137.csv"
 [9] "bootcamp-2026-registrations-2026-05-07-115143.csv"
[10] "bootcamp-2026-registrations-2026-05-07-123418.csv"
[11] "bootcamp-2026-registrations-2026-05-07-123421.csv"
[12] "bootcamp-2026-registrations-2026-05-07-123424.csv"
[13] "bootcamp-2026-registrations-2026-05-07-124523.csv"
[14] "bootcamp-2026-registrations-2026-05-07-125512.csv"
[15] "bootcamp-2026-registrations-2026-05-07-125835.csv"
[16] "bootcamp-2026-registrations-2026-05-07-133245.csv"
[17] "bootcamp-2026-registrations-2026-05-07-133429.csv"
[18] "bootcamp-2026-registrations-2026-05-07-133444.csv"
[19] "bootcamp-2026-registrations-2026-05-12-114332.csv"
[20] "bootcamp-2026-registrations-2026-05-12-115820.csv"
[21] "bootcamp-2026-silly-demo.csv"
[22] "bootcamp-2026.csv"
```

Clean & save Google Forms data

Import raw

In order to keep my raw data file "raw", I prefer to re-import the saved CSV. If I've saved time/date stamped versions, I need to get the most recent.

```
fl <- list.files(path = "data/raw_data", pattern = "registrations",
                 full.names = TRUE)
last_f_index <- length(fl)
latest_fn <- fl[last_f_index]
latest_fn
```

```
[1] "data/raw_data/bootcamp-2026-registrations-2026-05-12-115820.csv"
```

```
df <- readr::read_csv(file = latest_fn, show_col_types = FALSE)
dim(df)
```

```
[1] 94  8
```

Clean names

```
names(df)
```

```
[1] "Timestamp"
[2] "Email Address"
[3] "Which days of the bootcamp will you attend?"
[4] "What is your name?"
[5] "What is your department or unit?"
[6] "What is your current position?"
[7] "Any comments?"
[8] "drop-out"
```

The {dplyr} package `rename()` function helps rename the variables. The syntax is `new_name = old_name`.

```
df_clean <- df |>
  dplyr::rename(
    timestamp = "Timestamp",
    attend_days = "Which days of the bootcamp will you attend?",
    name = "What is your name?",
    psu_email = "Email Address",
    dept = "What is your department or unit?",
    position = "What is your current position?",
    comments = "Any comments?",
    dropped_out = "drop-out"
  )
```

Clean values

The registration dataset asked respondents to provide their department. We then add a new `college` variable based on that department. Because the department values are so varied, we have to normalize them on a case-by-case basis. The extremely inelegant solution to this is below.

First, we recode the `dept` variable.

```
df_clean <- df_clean |>
  dplyr::mutate(
    dept = dplyr::recode(
```

```
      dept,
      `Clinical Psychology` = "Psychology",
      `Psychology (Cognitive)` = "Psychology",
      `Psychology / SSRI` = "Psychology",
      `Psychology (Developmental)` = "Psychology",
      `Department of Psychology` = "Psychology",
      `Cognitive Psychology` = "Psychology",
      `Psychology, Developmental` = "Psychology",
      `Developmental Psychology (CAT Lab)` = "Psychology",
      `Developmental Psychology` = "Psychology",
      `Psych` = "Psychology",
      `College of Liberal Arts (psychology)` = "Psychology",

      `English language` = "English",
      `english` = "English",
      `English Language Teaching` = "English",
      `English Department` = "English",

      `Languages` = "Global Languages & Literatures",
      `Languages and Literature` = "Global Languages & Literatures",
      `Department of Foreign Languages` = "Global Languages & Literatures",

      `Linguistics` = "Applied Linguistics",

      `Department of Sociology and Criminology` = "Sociology & Criminology",

      `Communication Arts and Sciences` = "Communication Arts & Sciences",

      `Communication Sciences and Disorders` = "Communication Sciences & Disorders",
      `CSD` = "Communication Sciences & Disorders",

      `Human Development and Family Studies & Social Data Analytics` = "HDFS",
      `Human Development and Family Studies` = "HDFS",
      `Human Development and Family Studies (HDFS)` = "HDFS",
      `Department of Human Development and Family Studies` = "HDFS",
      `Human Development and Family Sciences` = "HDFS",
      `HDFS/DEMO` = "HDFS",

      `bbh` = "BBH",
      `Biobehavioral Health` = "BBH",
      `Biobehavioural Health` = "BBH",
      `Biobehavioural  Health` = "BBH",
```

```
      `Biobehavioral health` = "BBH",

      `RPTM` = "Recreation, Park, & Tourism Management",
      `Recreation, Park and Tourism Management` = "Recreation, Park, & Tourism Management",
      `Sociology and Social Data Analytics` = "Sociology",
      `Spanish Italian and portuguese` = "Spanish, Italian, & Portuguese",
      `Spanish, Italian, and Portuguese Department` = "Spanish, Italian, & Portuguese",
      `Spanish Italian and Portuguese` = "Spanish, Italian, & Portuguese",
      `Spanish, Italian, and Portuguese` = "Spanish, Italian, & Portuguese",

      `French and Francophone Studies` = "French & Francophone Studies",

      `DEMOG` = "Demography",

      `Germanic & Slavic Languages & Literatures` = "German & Slavic Languages",
      `Germanic and Slavic Languages and Literatures` = "German & Slavic Languages",

      `Nutrition` = "Nutritional Sciences",
      `Department of Nutritional Sciences` = "Nutritional Sciences",
      `Nurition` = "Nutritional Sciences",

      `College of IST` = "IST",

      `Statistics Department` = "Statistics",
      `Department of Statistics` = "Statistics",
      `stat` = "Statistics",
      `statistic` = "Statistics",

      `Math` = "Mathematics",
      `Astronomy and Astrophysics` = "Astronomy & Astrophysics",

      `SHS` = "Student Health Svcs",

      `Department of Chemical Engineering` = "Chemical Engineering",

      `ESM` = "Engineering Science & Mechanics",
      `Engineering Science` = "Engineering Science & Mechanics",
      `Engineering Science and Mechanics` = "Engineering Science & Mechanics",

      `EECS` = "Electrical Engineering & Comp Sci",

      `Department of Food Science` = "Food Science",
```

```
    `Libraries` = "University Libraries",
    `University libraries` = "University Libraries",

    `Ecosystem Science and Management` = "Ecosystem Science & Management",

    `PRC` = "Population Research Center",

    `TLT, PSU Libraries` = "University Libraries",

    `Business and Economics` = "Business & Economics",

    `EE` = "Electrical Engineering",

    `College of Medicine / Clinical and Translational Science Institute` = "CTSI",
    `College of Medicine, CTSI` = "CTSI",

    `Mechanical engineering,Penn state Harrisburg` = "Mechanical Engineering (Harrisburg)"

    `Smeal College of Business, Accounting` = "Accounting",

    `School of Science, Engineering, and Technology` = "Sci, Engr, & Tech",

    `institute for Computational and Data Sciences` = "ICDS",

    `Plant Pathology and environmental microbiology` = "Plant Pathology & Environmental Mi

    `Meteorology and Atmospheric Sciences` = "Meteorology & Atmospheric Sciences",
    `Department of Meteorology and Atmospheric Science` = "Meteorology & Atmospheric Scien

    `School of Labor and Employment Relations` = "School of Labor & Employment Relations",

    `PCD` = "Preservation Conservation & Digitization",

    `Vazquez Lab/Eberly College of Science/Bio` = "Biology",
    `Biology Department/ Vazquez lab` = "Biology",

    `WORKFORCE EDUCATION AND DEVELOPMENT` = "Workforce Education & Development",

    `Curriculum and Instructions` = "Curriculum & Instruction",

    `Kinesiology, HHD` = "Kinesiology",
```

```
      `BME` = "Biomedical Engineering",

      `Computer Science` = "Computer Science & Engineering",
      `Department of Computer Science and Engineering` = "Computer Science & Engineering",
      `College of Engineering/Computer Science` = "Computer Science & Engineering",

      `Ed Policy Studies` = "Education Policy Studies",

      `CHE` = "Chemical Engineering",

      `Agricultural and Biological Engineering` = "Agricultural & Biological Engineering",

      `Mechanical` = "Mechanical Engineering",

      `Health Administration` = "Health Policy & Administration",

      `CMPEN` = "Computer Engineering",

      `ESM/ REI` = "Engineering Science & Mechanics",

      `Civil Eng` = "Civil Engineering",

      `Meteorology and Atmospheric Science` = "Meteorology & Atmospheric Science",
      `Meteorology & Atmospheric Sciences` = "Meteorology & Atmospheric Science"
    )
  )
```

Then, we add (`mutate()`) a new `college` variable based on `dept`.

```
df_clean <- df_clean |>
  dplyr::mutate(
    college = recode_values(
      dept,
      "Agricultural & Biological Engineering" ~ "AgSci",
      "Ecosystem Science & Management" ~ "AgSci",
      "Entomology" ~ "AgSci",
      "Food Science" ~ "AgSci",
      "Plant Pathology & Environmental Microbiology" ~ "AgSci",
      "Plant Science" ~ "AgSci",

      "German & Slavic Languages" ~ "CLA",
      "Psychology" ~ "CLA",
```

```
    "Spanish, Italian, & Portuguese" ~ "CLA",

    "Anthropology" ~ "CLA",
    "Applied Linguistics" ~ "CLA",
    "Asian Studies" ~ "CLA",
    "C-SoDA" ~ "CLA",
    "Demography" ~ "CLA",
    "Communication Arts & Sciences" ~ "CLA",
    "Economics" ~ "CLA",
    "English" ~ "CLA",
    "French & Francophone Studies" ~ "CLA",
    "Global Languages & Literatures" ~ "CLA",
    "Office of Digital Pedagogies and Initiatives" ~ "CLA",
    "Political Science" ~ "CLA",
    "Sociology" ~ "CLA",
    "Sociology & Criminology" ~ "CLA",
    "School of Labor & Employment Relations" ~ "CLA",

    "Bellisario College of Communication" ~ "Comm",
    "Mass Communications" ~ "Comm",


    "Astronomy & Astrophysics" ~ "ECoS",
    "Biology" ~ "ECoS",
    "Chemistry" ~ "ECoS",
    "Integrative Science" ~ "ECoS",
    "Mathematics" ~ "ECoS",
    "Statistics" ~ "ECoS",

    "College of Education" ~ "Education",
    "LPS/LDT" ~ "Education",
    "Workforce Education & Development" ~ "Education",
    "Curriculum & Instruction" ~ "Education",
    "Education Policy Studies" ~ "Education",

    "Meteorology & Atmospheric Science" ~ "EMS",

    "Acoustics" ~ "Engineering",
    "Biomedical Engineering" ~ "Engineering",
    "Chemical Engineering" ~ "Engineering",
    "Chemical/Biomedical Engineering" ~ "Engineering",
    "Civil Engineering" ~ "Engineering",
```

```
    "College of Engineering" ~ "Engineering",
    "Computer Engineering" ~ "Engineering",
    "Computer Science & Engineering" ~ "Engineering",
    "Electrical Engineering & Comp Sci" ~ "Engineering",
    "Electrical Engineering" ~ "Engineering",
    "Engineering" ~ "Engineering",
    "Engineering Science & Mechanics" ~ "Engineering",
    "Material Science and Engineering" ~ "Engineering",
    "Mechanical Engineering" ~ "Engineering",

    "College of Human and Health Development" ~ "HHD",
    "Communication Sciences & Disorders" ~ "HHD",
    "BBH" ~ "HHD",
    "HDFS" ~ "HHD",
    "Kinesiology" ~ "HHD",
    "Nutritional Sciences" ~ "HHD",
    "Recreation, Park, & Tourism Management" ~ "HHD",

    "HHD" ~ "HHD",
    "Health Policy & Administration" ~ "HHD",

    "Biotechnology" ~ "Huck",
    "Plant Biology" ~ "Huck",

    "ICDS" ~ "ICDS",

    "IST" ~ "IST",
    "Data Analytics" ~ "IST",
    "Cybersecurity" ~ "IST",
    "Information Science and Technology" ~ "IST",

    "Preservation Conservation & Digitization" ~ "Libraries",
    "Research Informatics and Publishing" ~ "Libraries",
    "University Libraries" ~ "Libraries",

    "Neuroscience" ~ "Medicine",
    "Medicine" ~ "Medicine",
    "CTSI" ~ "Medicine",

    "College of Nursing" ~ "Nursing",

    "EESI" ~ "EESI",
```

```
      "Psychology (Harrisburg)" ~ "PSU Harrisburg",
      "PSU Harrisburg" ~ "PSU Harrisburg",
      "Mechanical Engineering (Harrisburg)" ~ "PSU Harrisburg",
      "PSU Harrisburg" ~ "PSU Harrisburg",
      "Sci, Engr, & Tech" ~ "PSU Harrisburg",

      "Business & Economics" ~ "PSU Brandywine",

      "Schreyer Institute for Teaching Excellence" ~ "Old Main",

      "OVPR" ~ "OVPR",
      "ORP" ~ "OVPR",

      "Population Research Center" ~ "SSRI",

      "Accounting" ~ "Smeal",
      "Marketing" ~ "Smeal",

      "University of Kansas, Psychology" ~ "UKansas"

    ),
    .default = "Unknown",
    .missing = "Unknown"
  )
```

Save cleaned

Full dataset privately

A recommended practice is to save derived datasets in a separate directory from the raw data. We follow that practice here by saving this cleaned dataset to `data/derivatives`.

We could add a date/timestamp to this file, but don't here for time reasons.

```
private_path <- "data/derivatives"
fn <- "bootcamp-2026-registrations-cleaned.csv"
private_fn <- file.path(private_path, fn)
readr::write_csv(df_clean, private_fn)
```

Public dataset without private variables

A portion of this dataset is going to be publicly available, so we drop the variables that have identifiable or sensitive information. The `dplyr::select()` command selects the variables we want to keep. The `-c("var_name")` syntax means *drop* the variables in the list. We could also pick the variables we want to keep.

```
df_public <- df_clean |>
  dplyr::select(-c("name", "psu_email", "comments"))
```

Save the cleaned public file.

```
public_path <- "data_public"
fn <- "bootcamp-2026-registrations-public.csv"
public_fn <- file.path(public_path, fn)
readr::write_csv(df_public, public_fn)
```