# Homework 2 Solutions

Stats 32: Introduction to R for Undergraduates

Due: Thursday, April 18, 2024, 11:59 am PT

## Instructions

You may (in fact, are encouraged to) use the Internet (including AI assistants, like ChatGPT) to search up any information to help you with this assignment, though you must cite any external (i.e. non-course related) resources that you use. Similarly, *after attempting this assignment by yourself*, you may collaborate with other students in the course, but you must each write your own code and acknowledge all students with whom you collaborated *for each problem* (you don't need to cite by subpart). However, you may not post on Internet forums (e.g. Stack Exchange) for help with this assignment; doing so is considered an Honor Code violation. You also may not copy verbatim any significant amount of code from the Internet (including AI assistants, like ChatGPT), even with citation. Feeding in the problems directly into AI assistants (or substantively paraphrased version) is also not permitted.

Please provide your code responses to each problem in the `.Rmd` file in the R code chunks directly below each subpart, inserting additional R code chunks if needed. Any text response can go right underneath the corresponding question.

On Gradescope, please submit a single `.pdf` file created by knitting the document with your responses. Problem 0 will provide guidance on how to do this.

Credit is given based on the approach and code, not necessarily the final answer.

---

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

## Problem 1

On Canvas, `droughts.csv` contains the percentage area of each county in California that was in each of 5 possible categories of drought — D0, D1, D2, D3, and D4 — and not in drought (None) at the end of each week in the years 2000 through 2020. Assume each observation is taken on the date given in `MapDate`.

(a) (3 points) Download and read in the `droughts.csv` file on Canvas, as a tibble. Use `dplyr::select()` to drop the columns `ValidStart` and `ValidEnd`, and store the result in a variable called `droughts`. Create a column called `SevereOrWorse` which corresponds to the percentage land area of the county in categories D2, D3, and D4 of drought.

Answer:

```r
droughts <- read_csv("droughts.csv") %>%
  select(!c(ValidStart, ValidEnd)) %>%
  mutate(SevereOrWorse = D2+D3+D4)
```

```
## Rows: 63568 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): FIPS, County, State
## dbl  (8): MapDate, None, D0, D1, D2, D3, D4, StatisticFormatID
## date (2): ValidStart, ValidEnd
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

(b) (3 points) Verify, using `group_by()` and `summarise()`, that for each `MapDate` and county pair, there is exactly one entry.

Answer: We group by `MapDate` and `County`, and then use the `n()` summary function with `summarise()` to show how many entries (i.e. rows in the tibble) there are for each `MapDate` and county pair.

```r
county_counts <- droughts %>%
  group_by(MapDate, County) %>%
  summarise(n_entries = n())
```

```
## `summarise()` has grouped output by 'MapDate'. You can override using the
## `.groups` argument.
```

We now verify no (`MapDate`, `County`) pair has a count greater than 1:

```r
max(county_counts$n_entries)
```

```
## [1] 1
```

(c) (2 points) For each variable in `droughts`, justify whether it is categorical or quantitative. It may help to look up what a FIPS code is.

Answer: `None`, `D0`, `D1`, `D2`, `D3`, `D4`, and `SevereOrWorse` are quantitative, since they can take on a continuous range of numeric values. `County` and `State` must be qualitative since they do not take numeric values. `FIPS` and `StatisticFormatID` is also qualitative; even though their values are numbers (and in fact coded as numeric), they are ID numbers with no sense of ordering. Finally, I would say `MapDate` is quantitative (as a date, which has a sense of numeric ordering) though you could reasonably argue it is qualitative (as there are only a fixed number of different days in the dataset).

(d) (4 points) What was the average percent area of Santa Clara County that was in severe or worse drought in 2020? Hint: You may want to create a new column called `Year` that specifies the year of the observation based on `MapDate`. You can do this with math (modular arithmetic or truncation), or by extracting the first 4 characters in `MapDate`. The question is asking you to average over all weeks in 2020.

Answer: One way to extract the year from `MapDate` is to subtract the remainder when `MapDate` is divided by 10000 from `MapDate` itself, and then dividing by 10000 to get the first 4 digits. After we do this we filter to Santa Clara County and Year 2020. Finally, we `summarise()` using `mean()` to get the average over all observations.

```r
droughts %>%
  mutate(Year = (MapDate - (MapDate %% 10000))/10000) %>%
  filter(County == "Santa Clara County", Year == 2020) %>%
  summarise(AvgSevereOrWorse = mean(SevereOrWorse))
```

```
## # A tibble: 1 x 1
##   AvgSevereOrWorse
##              <dbl>
## 1             5.96
```

On average in 2020, 5.96% of Santa Clara County was in severe or worse drought.

An alternative solution is to use substr to extract the year:

```r
droughts %>%
  mutate(Year = substr(MapDate, 1, 4)) %>%
  filter(County == "Santa Clara County", Year == 2020) %>%
  summarise(AvgSevereOrWorse = mean(SevereOrWorse))
```

```
## # A tibble: 1 x 1
##   AvgSevereOrWorse
##              <dbl>
## 1             5.96
```

(e) (2 points) Download and read in USA_counties.csv from Canvas and provide some numerical summary information (minimum, median, maximum) about the SQMI variable.

Answer:

```r
counties <- read_csv("USA_counties.csv")
```

```
## Rows: 3220 Columns: 61
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): NAME, STATE_NAME, STATE_FIPS, CNTY_FIPS, FIPS
## dbl (56): FID, OBJECTID, POPULATION, POP_SQMI, POP2010, POP10_SQMI, WHITE, B...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
counties %>%
  summarise(min_area=min(SQMI),
            median_area=median(SQMI),
            max_area=max(SQMI))
```

```
## # A tibble: 1 x 3
##   min_area median_area max_area
##      <dbl>       <dbl>    <dbl>
## 1     1.99        618.  147811.
```

(f) (4 points) Generate a tibble that contains the percentage land area of California in severe or worse drought for each MapDate. Order the tibble by MapDate, from most recent to least recent, and store it in a variable called CA_severe_percent. Then print out the first few rows of this tibble. Hint: You will need the information from SQMI in the previous part.

Answer: We begin by reading in and inspecting `USA_counties.csv`. Left joining `droughts` onto counties allows us to add information about the county areas to `droughts` (note we can join by FIPS code). Then we compute the area in severe or worse drought for each week and county, storing it in a column called `SevereOrWorseArea`. We group by the week (i.e. `MapDate`) before calling `summarise()` with `100*sum(SevereOrWorseArea)/sum(SQMI)` to get the percentage area of the entire state under severe or worse drought. Finally, we use `arrange()` to order the tibble in reverse order of `MapDate`.

```
CA_severe_percent <- droughts %>%
  left_join(counties, by="FIPS") %>%
  mutate(SevereOrWorseArea = 0.01*SevereOrWorse*SQMI) %>%
  group_by(MapDate) %>%
  summarise(PercentArea=100*sum(SevereOrWorseArea)/sum(SQMI)) %>%
  arrange(desc(MapDate))
head(CA_severe_percent)
```

```
## # A tibble: 6 x 2
##    MapDate PercentArea
##      <dbl>       <dbl>
## 1 20201229        74.4
## 2 20201222        74.4
## 3 20201215        74.4
## 4 20201208        66.8
## 5 20201201        48.2
## 6 20201124        48.2
```

(g) (2 points) Repeat part (f), but for the percentage land area of the 9 Bay Area counties (Marin County, Napa County, Sonoma County, Solano County, Alameda County, Contra Costa County, Santa Clara County, San Mateo County, and San Francisco County) in severe or worse drought for each `MapDate`. Store this tibble in a variable called `Bay_Area_severe_percent`.

Answer: We do the same as part (f) except we start by filtering out to only rows for counties in the Bay Area, using the `%in%` keyword from Lab 3.

```
bay_area_counties <- c("Marin County", "Napa County", "Sonoma County",
                       "Solano County", "Alameda County", "Contra Costa County",
                       "Santa Clara County", "San Mateo County", "San Francisco County")
Bay_Area_severe_percent <- droughts %>%
  filter(County %in% bay_area_counties) %>%
  left_join(counties, by="FIPS") %>%
  mutate(SevereOrWorseArea = 0.01*SevereOrWorse*SQMI) %>%
  group_by(MapDate) %>%
  summarise(PercentArea=100*sum(SevereOrWorseArea)/sum(SQMI)) %>%
  arrange(desc(MapDate))
head(Bay_Area_severe_percent)
```

```
## # A tibble: 6 x 2
##    MapDate PercentArea
##      <dbl>       <dbl>
## 1 20201229        88.5
## 2 20201222        88.5
## 3 20201215        88.5
## 4 20201208        88.7
## 5 20201201        88.7
## 6 20201124        88.7
```

(h) (4 points) In what proportion of weeks from 2000-2020 did the Bay Area have a higher percentage of its land area under severe or worse drought than California as a whole?

Answer: We first join `CA_severe_percent` to `Bay_Area_severe_percent` on `MapDate` so we have two columns, side by side, showing the percent area in all of CA and the percent area of the Bay Area in severe or worse drought. We use the optional `suffix` argument in `left_join()` to give the `PercentArea` columns more descriptive names `PercentAreaCA` and `PercentAreaBay` (since by default, the `left_join()` function gives them the names `PercentArea.x` and `PercentArea.y`).

```
CA_and_Bay <- CA_severe_percent %>%
  left_join(Bay_Area_severe_percent, by="MapDate", suffix=c("CA", "Bay"))
head(CA_and_Bay)
```

```
## # A tibble: 6 x 3
##    MapDate PercentAreaCA PercentAreaBay
##      <dbl>         <dbl>          <dbl>
## 1 20201229          74.4           88.5
## 2 20201222          74.4           88.5
## 3 20201215          74.4           88.5
## 4 20201208          66.8           88.7
## 5 20201201          48.2           88.7
## 6 20201124          48.2           88.7
```

Now we compare the number of rows where the `PercentAreaBay` value is greater than the `PercentAreaCA` value, or equivalently the number of rows where the `PercentAreaBay` value minus the `PercentAreaCA` value is greater than zero:

```
total_weeks <- nrow(CA_and_Bay)
bay_weeks <- nrow(CA_and_Bay %>%
                    mutate(CA_minus_Bay=PercentAreaBay-PercentAreaCA) %>%
                    filter(CA_minus_Bay > 0))
bay_weeks / total_weeks
```

```
## [1] 0.1979927
```

19.8% of weeks saw the Bay Area have a greater percentage area in severe or worse drought than CA as a whole.

(i) (1 point) Save `CA_severe_percent` from above as `CA_severe_percent.csv` in a folder where you won't lose it.

Answer:

```
path <- "/Users/harrisonli/Documents/iCloud_Documents/Stanford/Teaching/Stats 32/Spring 2024/CA_severe_p
write_csv(CA_severe_percent, path)
```