Homework 3 Solutions

Stats 32: Introduction to R for Undergraduates

Due: Thursday, April 25, 2024, 11:59 am PT

```
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 error
```

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.

All visualizations should have an appropriate title and axis labels.

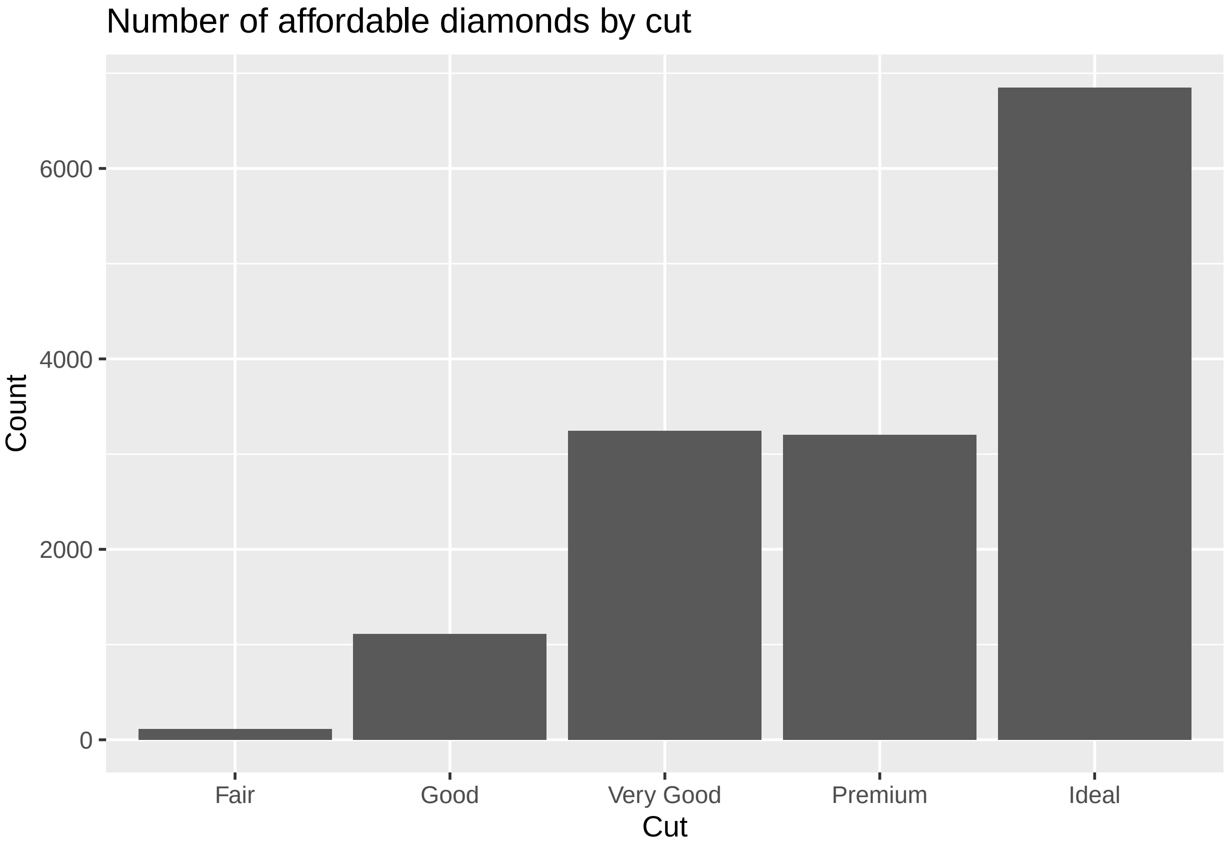
Problem 1: More diamonds

This problem pertains to the `diamonds` tibble, built in to `ggplot2`.

(a) (3 points) Ashley is interested in buying a diamond for their fiance. They have a tight budget of no more than $1,000, and are most concerned with the cut of the diamond. Create an appropriate visualization, using a geom covered in lecture, that shows the number of diamonds in this price range that have each `cut`.

Answer: We first filter out to diamonds of price no more than $1,000. Within these diamonds we are trying to visualize the distribution of a single categorical variable: cut. Thus we create a bar plot using `geom_bar()`:

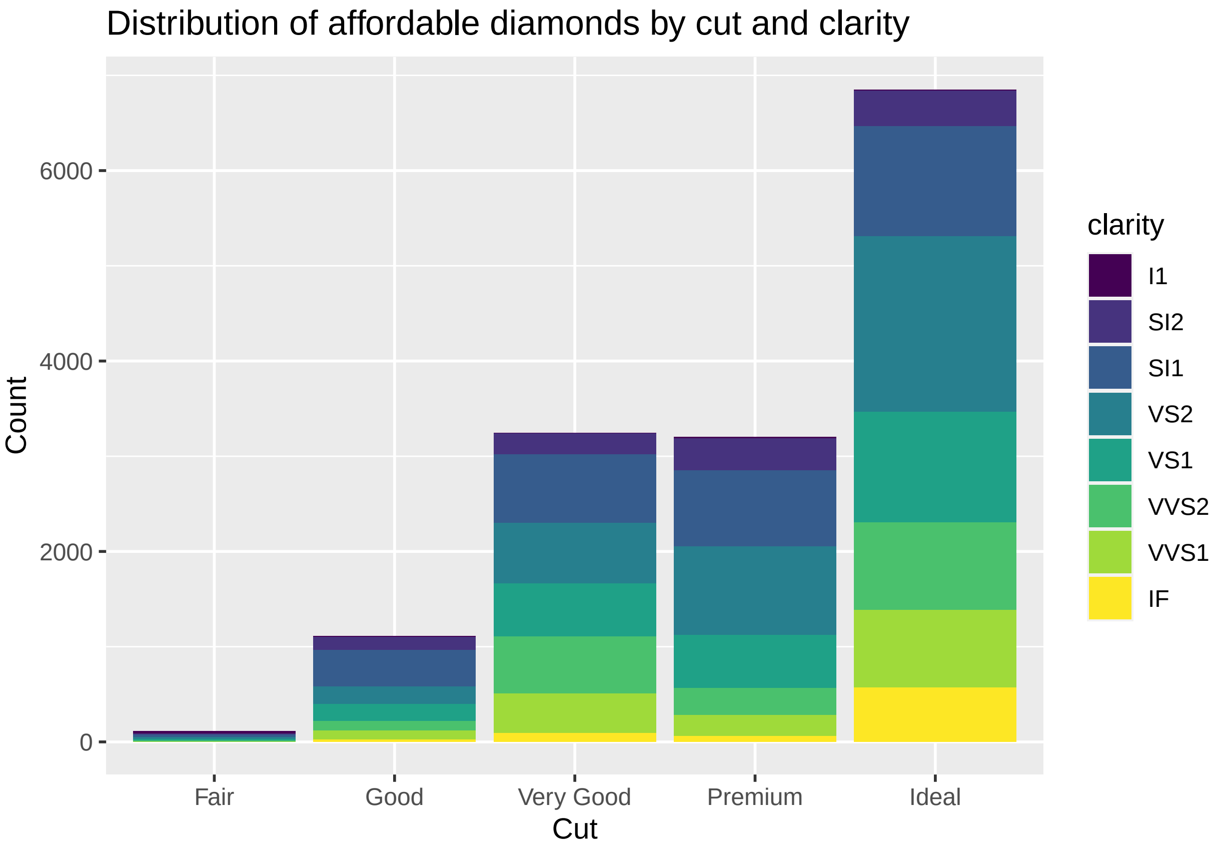
```
diamonds %>%
  filter(price <= 1000) %>%
  ggplot(aes(x=cut)) +
  geom_bar() +
  ylab("Count") +
  xlab("Cut") +
  ggtitle("Number of affordable diamonds by cut")
```



(b) (3 points) Now, enhance your visualization in part (a) to show the distribution of `clarity` within each `cut` (in addition to the overall distribution of `cut`). We are again only considering the diamonds within Ashley's budget.

Answer: Since we want to visualize the distribution of the categorical variable `clarity` within `cut`, we create a stacked bar chart, with the bar fill color corresponding to `clarity`:

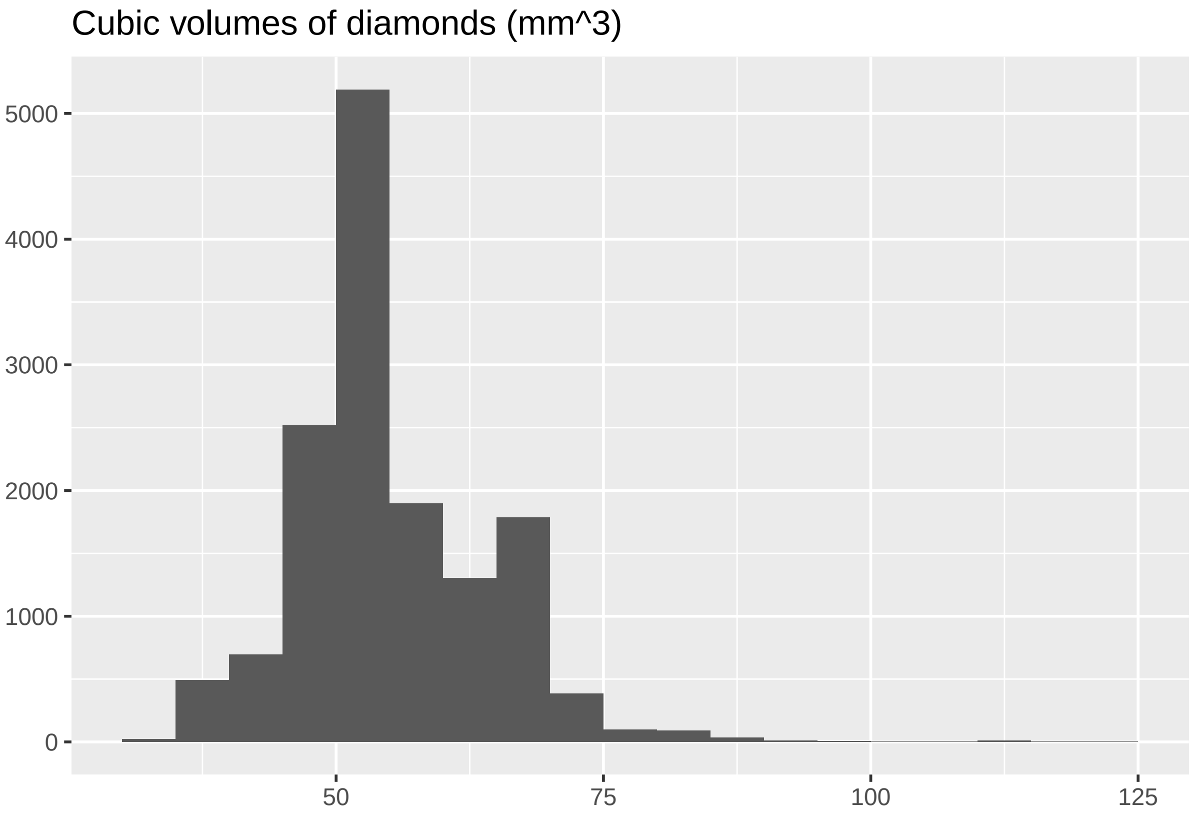
```
diamonds %>%
  filter(price <= 1000) %>%
  ggplot(aes(x=cut)) +
  geom_bar(aes(fill=clarity)) +
  ylab("Count") +
  xlab("Cut") +
  ggtitle("Distribution of affordable diamonds by cut and clarity")
```



(c) (4 points) Define the cubic volume of a diamond as the product of its length, width, and depth. Visualize the distribution of cubic volumes among diamonds cheap enough for Ashley. Is the distribution right-skewed, left-skewed, or neither? Explain.

Answer: From the help page, the column `x` corresponds to length, `y` is width, and `z` is height, all in mm. Cubic volumes are numeric so we can use either a histogram or a boxplot. We see a longer right tail in the histogram so the distribution is right-skewed, though only somewhat.

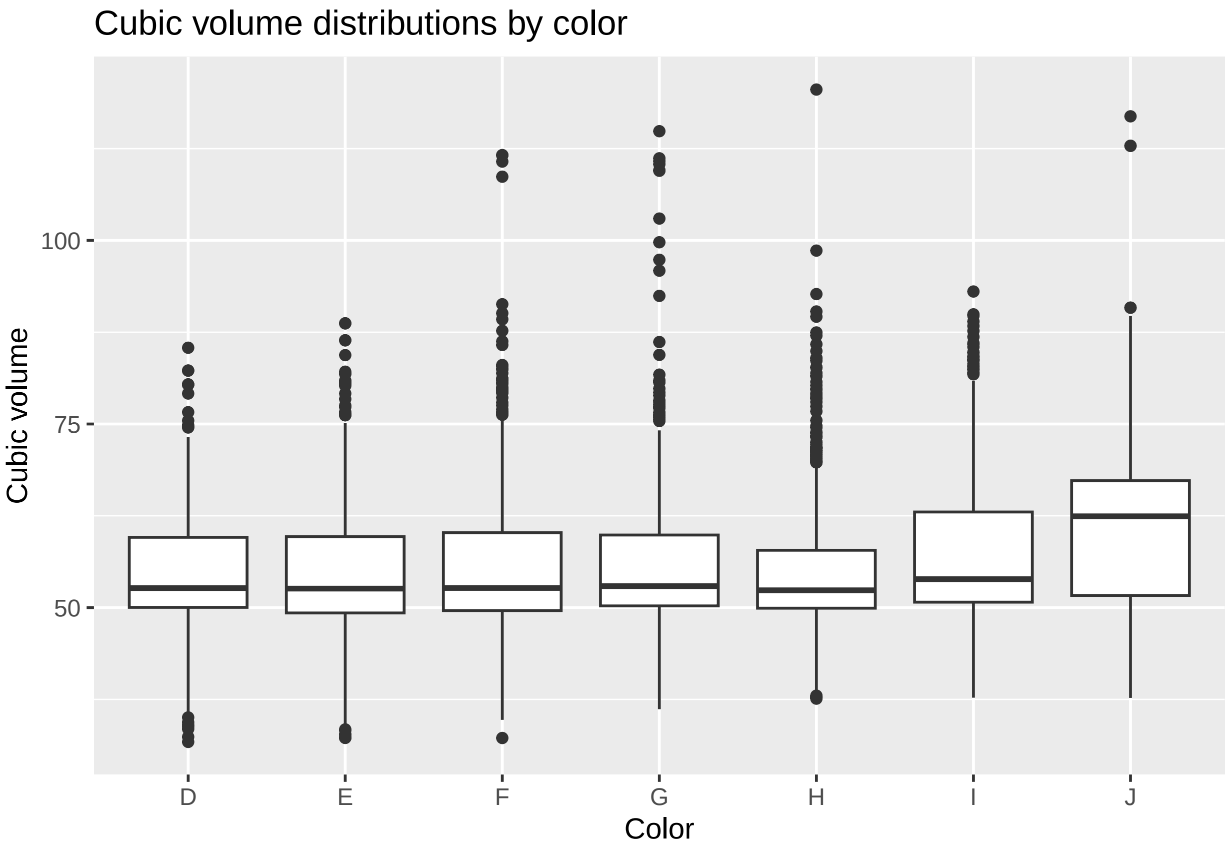
```
diamonds %>%
  filter(price <= 1000) %>%
  mutate(cubic_volume=x*y*z) %>%
  ggplot(aes(x=cubic_volume)) +
  geom_histogram(boundary=0, binwidth=5) +
  ylab("") +
  xlab("") +
  ggtitle("Cubic volumes of diamonds (mm^3)")
```



(d) (3 points) Create an appropriate visualization to determine whether certain diamond colors tend to have greater cubic volumes (among those in Ashley's budget)?. Do they?

Answer: We seek to understand the distribution of cubic volume broken down by `color`. This calls for side-by-side boxplots:

```r
diamonds %>%
  filter(price <= 1000) %>%
  mutate(cubic_volume=x*y*z) %>%
  ggplot(aes(x=color, y=cubic_volume)) +
  geom_boxplot() +
  xlab("Color") +
  ylab("Cubic volume") +
  ggtitle("Cubic volume distributions by color")
```

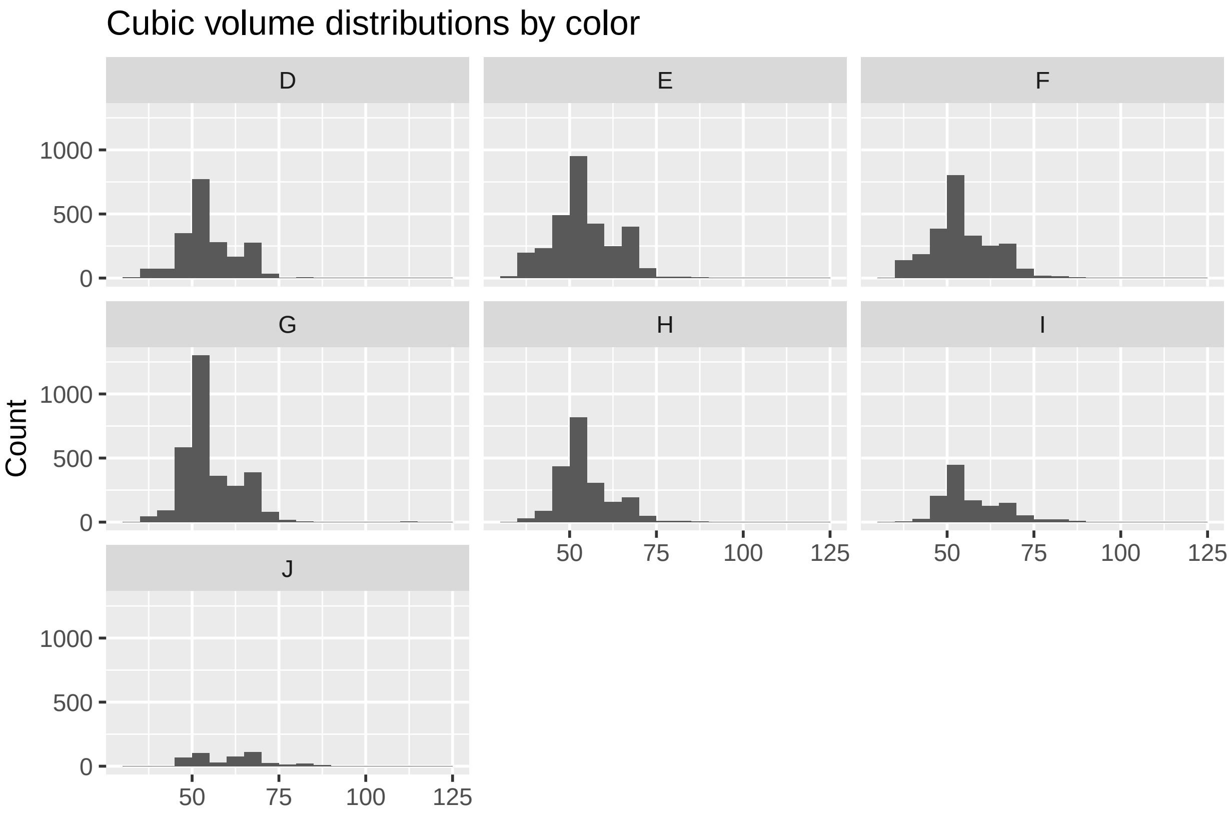


J colored diamonds seem to have somewhat higher cubic volumes, but in general there are not large differences across colors.

(e) (4 points) Now repeat part (d), but showing a histogram of cubic volumes for each color (and using faceting). What is one advantage and one disadvantage of using these histograms, as opposed to the side-by-side boxplots?

Answer:

```
diamonds %>%
  filter(price <= 1000) %>%
  mutate(cubic_volume=x*y*z) %>%
  ggplot(aes(x=cubic_volume)) +
  geom_histogram(boundary=0, binwidth=5) +
  facet_wrap(~color) +
  ggtitle("Cubic volume distributions by color") +
  xlab("") +
  ylab("Count")
```



The histograms convey a more detailed breakdown of each distribution than a side-by-side boxplot would. However, they take up more space than side-by-side boxplots, and also do not clearly show outliers.

Problem 2: More droughts

Let's revisit `droughts.csv` from Homework 2.

(a) (3 points) Create an appropriate visualization to show the percentage land area of Santa Clara County under severe or worse drought for each week in 2020 (recall the definition of severe or worse in Homework 2). Make sure your vertical axis goes from 0 to 100%.

Note: Run the lines below to read in `droughts.csv` (add a line to change the working directory, if needed) and create an additional column in the tibble called `Date` with the date in "date format" (rather than numeric). Use this `Date` column in your visualization.

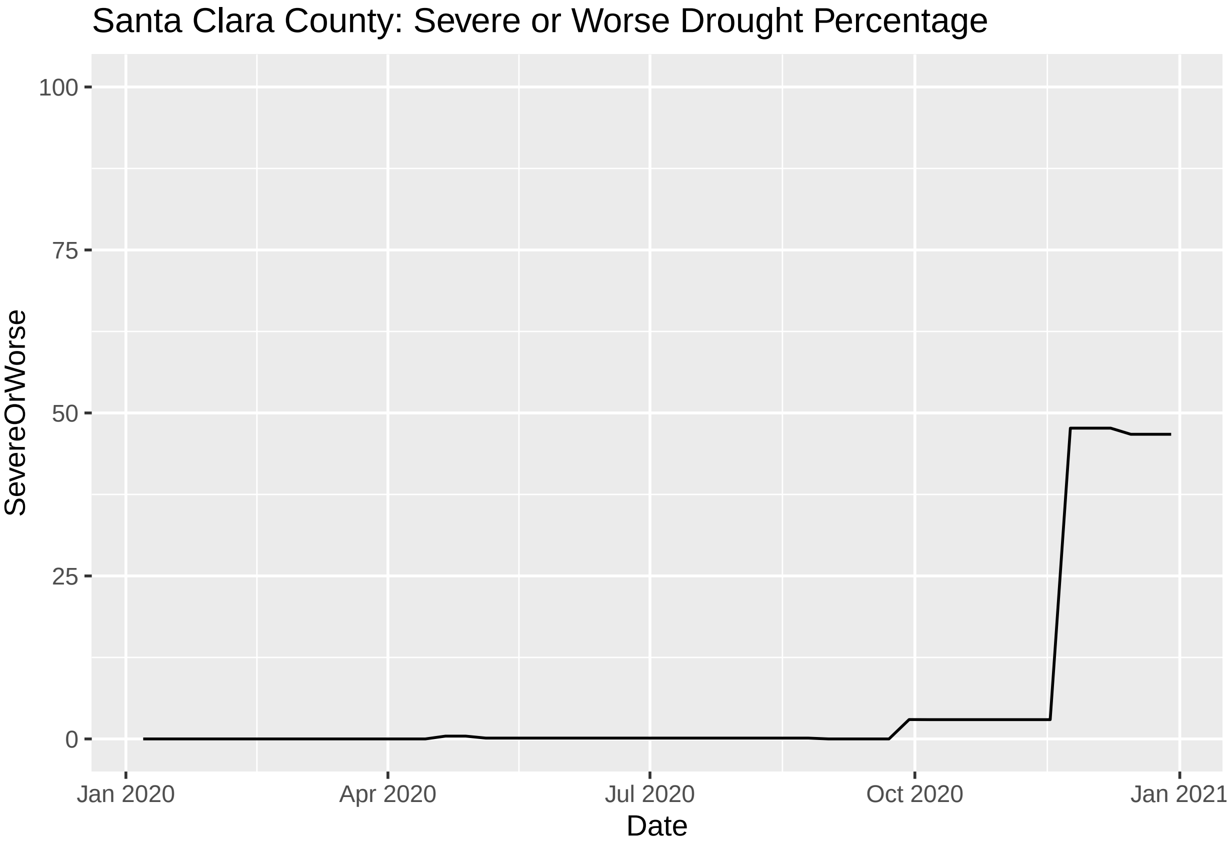
```r
# Read in the droughts.csv file and add a date column
droughts <- read_csv("droughts.csv")
```

```
## 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.
```

```
droughts$Date <- as.Date(as.character(droughts$MapDate), format="%Y%m%d")
```

Answer: Since we are trying to visualize the evolution of a quantitative variable SevereOrWorse over time, a line plot with geom_line() seems most appropriate.

```
droughts %>%
  mutate(Year=substr(MapDate, 1, 4)) %>%
  filter(County == "Santa Clara County" & Year == "2020") %>%
  mutate(SevereOrWorse=D2+D3+D4) %>%
  ggplot(aes(x=Date, y=SevereOrWorse)) +
  geom_line() +
  ylim(c(0, 100)) +
  ggtitle("Santa Clara County: Severe or Worse Drought Percentage")
```



(b) (6 points) Create a stacked bar chart showing the percentage land area in the Bay Area and Southern California under each of the 5 drought severity categories (D0 through D4) on December 29, 2020. You should have two vertical bars side by side: one for the Bay Area, and one for Southern California. Each bar should be sliced horizontally into 5 different colored regions, with each color corresponding to a particular drought category in D0 through D4. The total height of each bar should be the total percentage area of the county in one of those drought categories. Make sure that within each bar, D0 is the bottom-most portion, all the way up through D4 at the top, and use a color palette conveying increasing severity.

Note: The Bay Area is made up of 9 counties: Alameda County, Contra Costa County, Marin County, Napa County, San Francisco County, San Mateo County, Santa Clara County, Solano County, and Sonoma County. Southern California consists of 10 counties: Imperial County, Kern County, Los Angeles County, Orange County, Riverside County, San Bernardino County, San Diego County, Santa Barbara County, San Luis

Obispo County, and Ventura County.

Hint: Recall Problem 1(g) from Homework 2. You may want to use `pivot_longer()` (Lecture 2) so that you have a tibble containing 10 rows, one for each (region, drought category) pair, where region is either “Bay Area” or “Southern California.” Then read the help page for `geom_bar()` or `geom_col()` to learn what happens if multiple bars in a bar plot occupy the same `x` position.

Answer: Our solution strongly recalls the construction of a tibble containing the percent land area of the Bay Area in severe or worse drought from Problem 1(g) of Homework 2. There are some differences here: we want both Bay Area and Southern California, and want a result for each drought category D0 through D4.

```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
summary(counties$SQMI)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      1.99    427.04    618.25   1117.66    933.00 147811.16
```

```r
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")
so_cal_counties <- c("Imperial County", "Kern County", "Los Angeles County", "Orange County",
                     "Riverside County", "San Bernardino County", "San Diego County",
                     "Santa Barbara County", "San Luis Obispo County")
severe_percent <- droughts %>%
  filter(County %in% c(bay_area_counties, so_cal_counties), MapDate==20201229) %>%
  mutate(Region=ifelse(County %in% bay_area_counties, "Bay Area", "Southern California")) %>%
  left_join(counties, by="FIPS") %>%
  mutate(D0Area = 0.01*D0*SQMI,
         D1Area = 0.01*D1*SQMI,
         D2Area = 0.01*D2*SQMI,
         D3Area = 0.01*D3*SQMI,
         D4Area = 0.01*D4*SQMI) %>%
  group_by(Region, MapDate) %>%
  summarise(D0=100*sum(D0Area)/sum(SQMI),
            D1=100*sum(D1Area)/sum(SQMI),
            D2=100*sum(D2Area)/sum(SQMI),
            D3=100*sum(D3Area)/sum(SQMI),
            D4=100*sum(D4Area)/sum(SQMI)) %>%
  select(!MapDate)
```

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

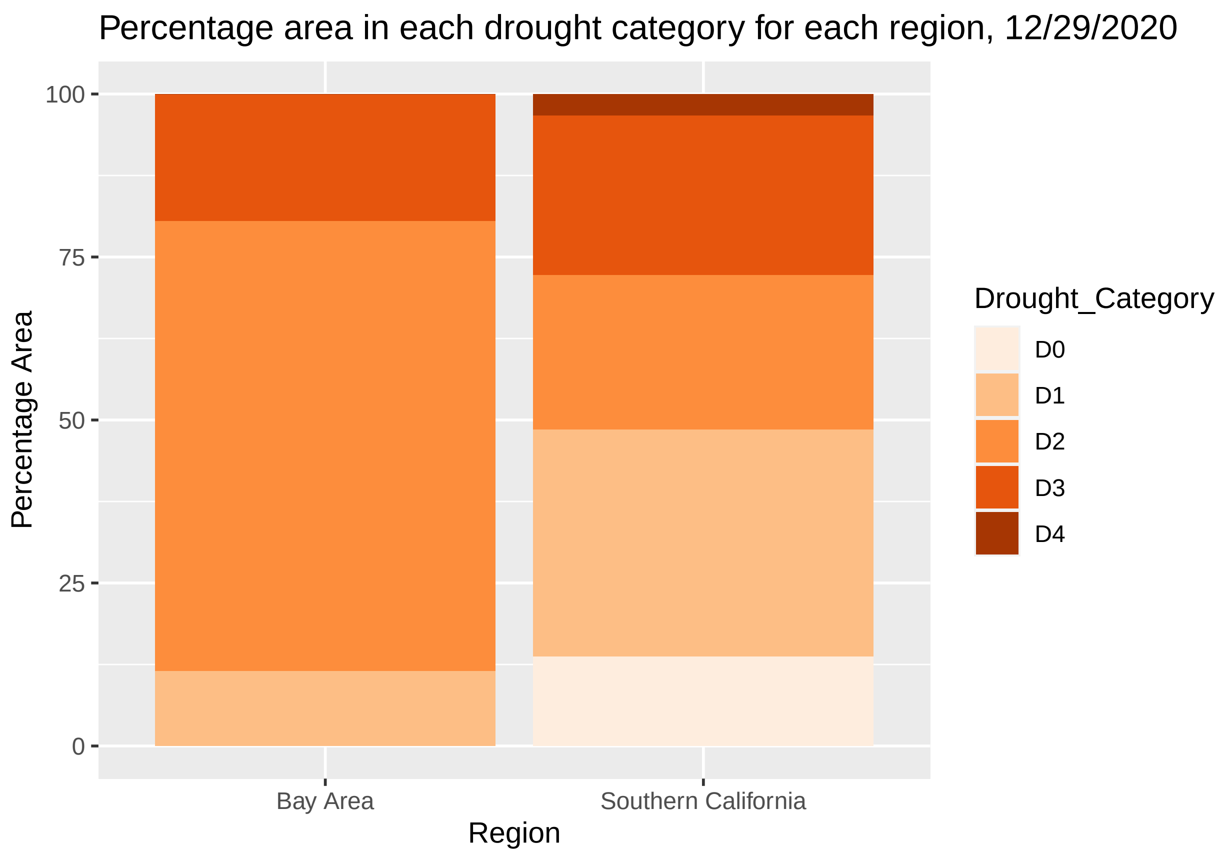
```
severe_percent
```

```
## # A tibble: 2 x 6
## # Groups:   Region [2]
##   Region                 D0    D1    D2    D3    D4
##   <chr>               <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Bay Area              0    11.5  69.0  19.5  0   
## 2 Southern California  13.8  34.8  23.7  24.5  3.28
```

Now as the hint suggests, we use `pivot_longer()` so that we have one row for each (region, drought category) pair, and the corresponding percent area in that category of drought for that region in a new column called `Percentage_Area`. Then we can create a stacked bar chart using `geom_col()` with `Percentage_Area` on the y-axis and `Region` on the x-axis. From the help page for `geom_col()`, we know that specifying `position="stack"` (which is enabled by default) means we'll have 5 stacked bars on top of each other for each region, since each region has 5 rows corresponding to it (one for each drought category). To differentiate among these bars, we assign a different fill by `Drought_Category`.

To put D0 on the bottom we specify `position_stack(reverse=TRUE)` in the position argument. Finally we use a sequential palette (Lab 5) to convey increasing severity.

```
severe_percent %>%
  pivot_longer(cols=c("D0", "D1", "D2", "D3", "D4"),
               names_to="Drought_Category",
               values_to="Percentage_Area") %>%
  ggplot(aes(x=Region, y=Percentage_Area)) +
  geom_col(aes(fill=Drought_Category), position=position_stack(reverse=TRUE)) +
  xlab("Region") +
  ylab("Percentage Area") +
  ggtitle("Percentage area in each drought category for each region, 12/29/2020") +
  scale_fill_brewer(type="seq", palette=7)
```



We see that southern California had more area in the worst categories of drought but also more area in the less severe categories of drought. Much of the Bay Area was in the middle, D2 category of drought.