Assignment4

Kyle Bambling

2023-11-02

Question 1

A common task is to take a set of data that has multiple categorical variables and create a table of the number of cases for each combination. An introductory statistics textbook contains a dataset summarizing student surveys from several sections of an intro class. The two variables of interest for us are `Gender` and `Year` which are the students gender and year in college.

a) Download the dataset and correctly order the `Year` variable using the following:

```
Survey <- read.csv('https://www.lock5stat.com/datasets3e/StudentSurvey.csv', na.strings=c('',' ')) %>%
  mutate( Year = factor(Year, levels = c( 'FirstYear', 'Sophomore','Junior','Senior')))
head(Survey)
```

```
##        Year Sex Smoke   Award HigherSAT Exercise TV Height Weight Siblings
## 1    Senior   M    No Olympic      Math       10  1     71    180        4
## 2 Sophomore   F   Yes Academy      Math        4  7     66    120        2
## 3 FirstYear   M    No   Nobel      Math       14  5     72    208        2
## 4    Junior   M    No   Nobel      Math        3  1     63    110        1
## 5 Sophomore   F    No   Nobel    Verbal        3  3     65    150        1
## 6 Sophomore   F    No   Nobel    Verbal        5  4     65    114        2
##   BirthOrder VerbalSAT MathSAT  SAT  GPA Pulse Piercings
## 1          4       540     670 1210 3.13    54         0
## 2          2       520     630 1150 2.50    66         3
## 3          1       550     560 1110 2.55   130         0
## 4          1       490     630 1120 3.10    78         0
## 5          1       720     450 1170 2.70    40         6
## 6          2       600     550 1150 3.20    80         4
```

b) Using some combination of `dplyr` functions, produce a data set with eight rows that contains the number of responses for each gender:year combination. Make sure your table orders the `Year` variable in the correct order of `First Year`, `Sophmore`, `Junior`, and then `Senior`.

```
survey.2 <- Survey %>%
  count(Year, Sex) %>%
  filter(!is.na(Year))
survey.2
```

```
##        Year Sex  n
## 1 FirstYear   F 43
## 2 FirstYear   M 51
## 3 Sophomore   F 96
## 4 Sophomore   M 99
## 5    Junior   F 18
## 6    Junior   M 17
## 7    Senior   F 10
```

```
## 8    Senior   M 26
```

c) Using `tidyr` commands, produce a table of the number of responses

```
pivot_wider(survey.2, names_from = Year, values_from = n)
```

```
## # A tibble: 2 x 5
##   Sex   FirstYear Sophomore Junior Senior
##   <chr>     <int>     <int>  <int>  <int>
## 1 F            43        96     18     10
## 2 M            51        99     17     26
```

Question 2

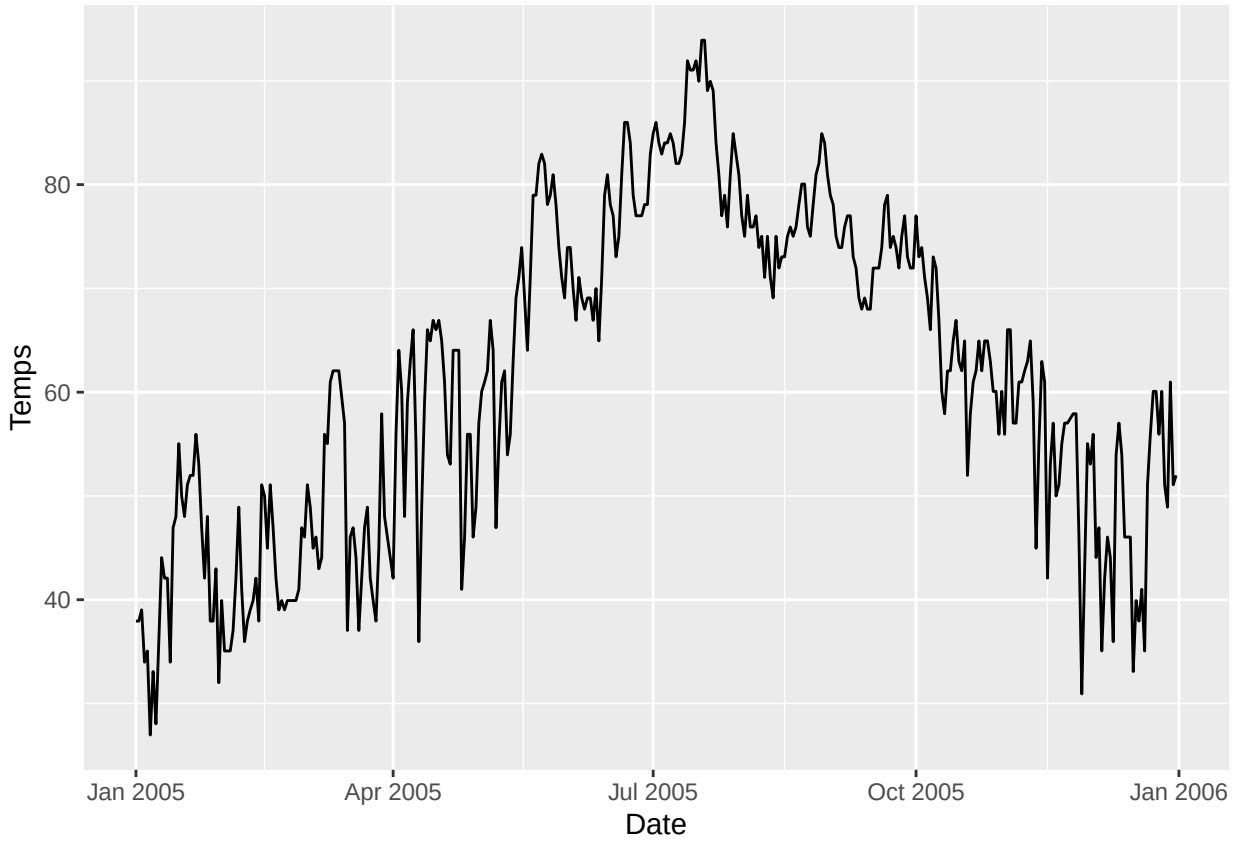
From the book website, there is a .csv file of the daily maximum temperature in Flagstaff at the Pulliam Airport.

```
airport <- read.csv('https://raw.githubusercontent.com/dereksonderegger/444/master/data-raw/FlagMaxTemp
```

a) Create a line graph that gives the daily maximum temperature for 2005.

```
airport.2005 <- airport %>% filter(Year == 2005)
airport.2005.long <- pivot_longer(airport.2005, X1:X31, names_to = 'Days', values_to = 'Temps')
airport.2005.long <- airport.2005.long %>%
  mutate( Days = str_replace(Days, pattern='X', replacement='') ) %>%
  drop_na() %>%
  mutate( Date = make_date(year=Year, month=Month, day=Days))

airport.2005.long %>% ggplot( aes(x=Date, y=Temps) ) +
  geom_line()
```

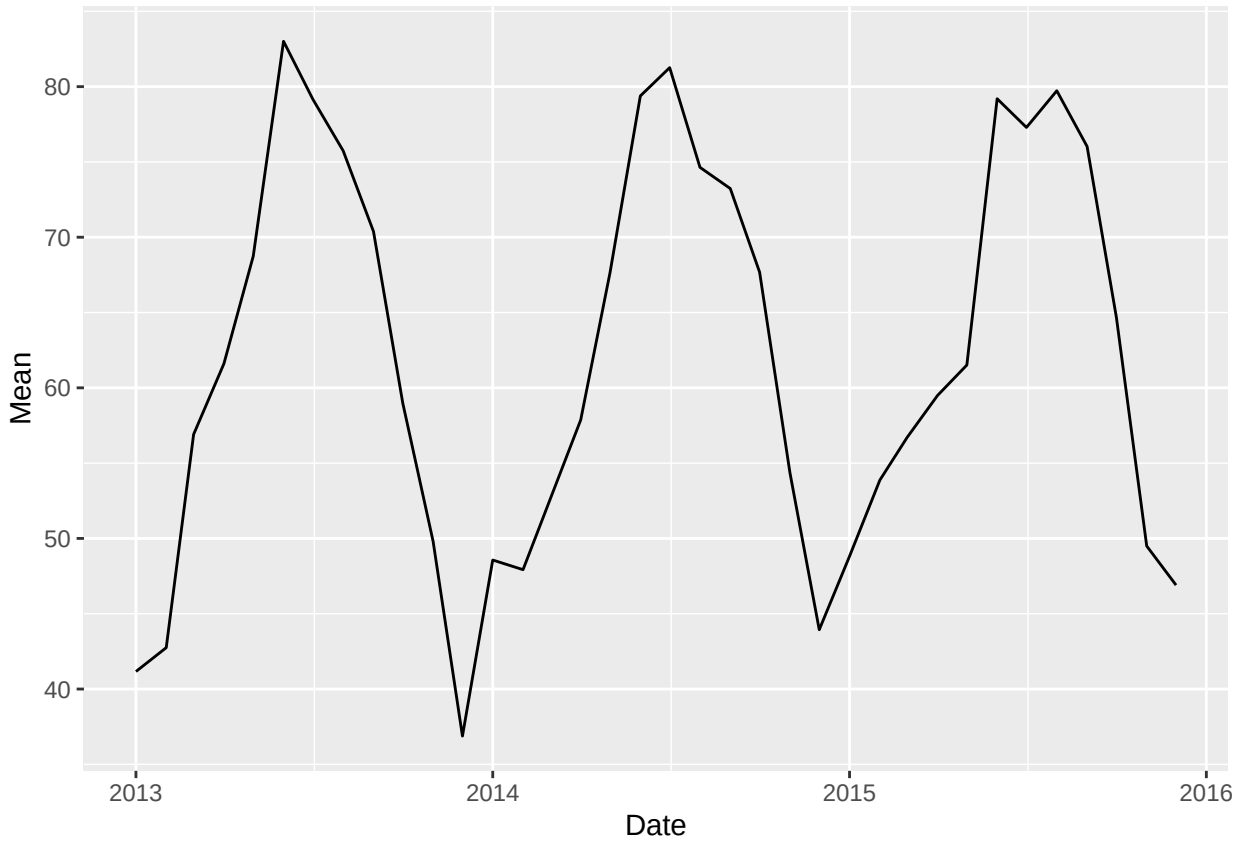


b) Create a line graph that gives the monthly average maximum temperature for 2013 - 2015.

```
airport.2013 <- airport %>% filter(2013 <= Year & Year <= 2015)
airport.2013.long <- pivot_longer(airport.2013, X1:X31, names_to = 'Days', values_to = 'Temps')
airport.2013.long <- airport.2013.long %>%
  mutate( Days = str_replace(Days, pattern='X', replacement='') ) %>%
  drop_na() %>%
  group_by(Year, Month) %>%
  summarise( Mean = mean(Temps)) %>%
  mutate( Date = make_date(year=Year, month=Month))
```

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

```
airport.2013.long %>%
  ggplot( aes(x=Date, y=Mean)) +
  geom_line()
```



Question 4

For this problem we will consider two simple data sets.

```
A <- tribble(
  ~Name, ~Car,
  'Alice', 'Ford F150',
  'Bob',   'Tesla Model III',
  'Charlie', 'VW Bug')

B <- tribble(
  ~First.Name, ~Pet,
```

```
  'Bob',  'Cat',
  'Charlie', 'Dog',
  'Alice', 'Rabbit')
```

a) Squish the data frames together to generate a data set with three rows and three columns. Do two ways: first using `cbind` and then using one of the `dplyr join` commands.

```
B <- rename( B, Name = First.Name )
ABbind <- cbind(A, B)

ABbind <- full_join(A,B)
```

```
## Joining with `by = join_by(Name)`
```

```
ABbind
```

```
## # A tibble: 3 x 3
##   Name    Car             Pet
##   <chr>   <chr>           <chr>
## 1 Alice   Ford F150       Rabbit
## 2 Bob     Tesla Model III Cat
## 3 Charlie VW Bug          Dog
```

b) It turns out that Alice also has a pet guinea pig. Add another row to the `B` data set. Do this using either the base function `rbind`, or either of the `dplyr` functions `add_row` or `bind_rows`.

```
B <- B %>% add_row( Name = 'Alice', Pet = 'Guinea Pig')
```

c) Squish the `A` and `B` data sets together to generate a data set with four rows and three columns. Do this two ways: first using `cbind` and then using one of the `dplyr join` commands. Which was easier to program? Which is more likely to have an error.

```
ABjoin <- full_join(A, B)
```

```
## Joining with `by = join_by(Name)`
```

```
ABjoin
```

```
## # A tibble: 4 x 3
##   Name    Car             Pet
##   <chr>   <chr>           <chr>
## 1 Alice   Ford F150       Rabbit
## 2 Alice   Ford F150       Guinea Pig
## 3 Bob     Tesla Model III Cat
## 4 Charlie VW Bug          Dog
```