```
install.packages("tidyverse")
install.packages("nycflights13")
library(tidyverse)
library(nycflights13)
options(repr.plot.width=5, repr.plot.height=4)
```

```
Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)

Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)

Warning message in system("timedatectl", intern = TRUE):
"running command 'timedatectl' had status 1"
── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
✔ ggplot2 3.4.0      ✔ purrr   1.0.1
✔ tibble  3.1.8      ✔ dplyr   1.0.10
✔ tidyr   1.2.1      ✔ stringr 1.4.1
✔ readr   2.1.3      ✔ forcats 0.5.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
```

STATS 306

Homework 3: Advanced `dplyr` and tidy data

For each problem, enter the R code in the cell marked "YOUR SOLUTION HERE".

Problem 1: Why so delayed? (4 points)

The following code adds a variable `week` to `flights`, such that `week==1` for the first seven days of the year, `week==2` for days 8-14, etc. (In the second half of the semester we will learn how to work with times and date data using the `lubridate` package.)

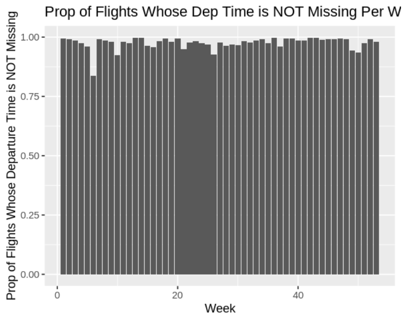
```
flights_week = mutate(flights, week=lubridate::week(time_hour))
```

(a) Make a bar plot of the proportion of flights each week whose actual departure time is NOT missing. The x-axis of your plot should contain the weeks of the year, ranging from 1 to 52, and the y-axis of your plot should be a number between 0 and 1 showing the decimal proportion of flights that have a departure time. What sort of plot geometry (line, bar, point, histogram, etc.) do you think is appropriate for this kind of plot? Does anything about this plot jump out at you? What and why? *1 point*

```
# Your solution here

flights_week %>%
  mutate(notMissing = !is.na(dep_time)) %>%
  group_by(week) %>%
  summarise(prop = mean(notMissing)) %>%
  ggplot(aes(x = week, y = prop)) +
  geom_bar(stat = "identity") +
  labs(title = "Prop of Flights Whose Dep Time is NOT Missing Per Week", x = "Week", y="Prop of Flights Whose Departure Time is NOT Missing")
```

```
#A bar plot would be the most appropriate for this kind of data as we
#want to compare values among different groups (in this case weeks). A reason
#why a bar plot is more useful than a histogram in this case is that histograms
#have continuous values on the x axis when we have discrete values (weeks can't
#be decimals). Something that jumps out at me on this graph is that week 6 has a
#significantly lower prop of flights who departure time is not missing compared
#to all other weeks in the year.
```



(b) For the week with the highest fraction of missing departure times, generate a table which shows the proportion of missing departure times for each day of that week. Your table should have columns `year`, `month`, `day`, and `prop_miss_dep_time`. Sort your table in chronological order and store it in a variable called `table1b`. *1 point*

```
# Your solution here

table1b <- flights_week %>%
  mutate(missingDep = is.na(dep_time))  %>%
  filter(week ==6)  %>%
  group_by(year, month, day) %>%
  summarise(prop_miss_dep_time = mean(missingDep)) %>%
  arrange(prop_miss_dep_time) %>%
  print
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
# A tibble: 7 × 4
# Groups:   year, month [1]
   year month   day prop_miss_dep_time
  <int> <int> <int>              <dbl>
1  2013     2     7            0.00429
2  2013     2     6            0.00888
3  2013     2     5            0.0179
4  2013     2    10            0.0314
5  2013     2    11            0.0786
6  2013     2     8            0.508
7  2013     2     9            0.575
```

(c) 2 days in `table1b` should jump out at you. What you're discovering from the data is the [North American Blizzard of 2013](). Many flights were cancelled due to extreme weather conditions. Identify the proportion of cancelled flights out of LaGuardia Airport (LGA) during the days that jumped out at you for each airline carrier in descending order. *1 point*

```
# Your solution here

#The two dates that jump out at me are Feb 9th 2013 and Feb 8th 2013. As those days
#have the highest proportion od missing depature times during week 6.

flights_week %>%
  filter((year == 2013 & month == 2 & day == 8) | (year == 2013 & month == 2 & day == 9) ) %>%
  filter(origin == "LGA") %>%
  mutate(missingDep = is.na(dep_time)) %>%
  group_by(carrier) %>%
  summarise(meanMissingDep = mean(missingDep)) %>%
  arrange(desc(meanMissingDep))  %>%
  print
```

```
# A tibble: 12 × 2
   carrier meanMissingDep
   <chr>            <dbl>
 1 YV               1
 2 9E               0.667
 3 DL               0.612
 4 MQ               0.6
 5 UA               0.562
 6 US               0.559
 7 FL               0.524
 8 B6               0.5
 9 EV               0.5
10 F9               0.5
11 WN               0.444
12 AA               0.431
```

(d) In your own words, summarize your findings from the previous exercises. Most importantly, comment on which airlines were the most and least cautious in terms of flight cancellations. Can you think of any reason why this might be? *1 point*

On Feb 8th to 9th during the North American Blizzard of 2013, carrier YV had the highest proportion of cancelled flights (in fact 100% of YV flights were cancelled), making it the most cautious during the blizzard. In contrary, AA had the lowest proportion of cancelled flights (at 43.1%), making it the least cautious during the blizzard. A reason to explain this is perhaps YV is a small carrier and thus does not have the best planes or extra safety features that could fly in hazardous conditions like bigger carriers like AA.

Problem 2: Graduate school admissions (4 points)

This problem studies a built-in dataset called `UCBAdmissions`. It contains graduate school admissions data from 1973 for six departments at UC Berkeley:

```
help(UCBAdmissions)
```

```
data(UCBAdmissions)
ucb <- as_tibble(UCBAdmissions) %>% print
```

```
# A tibble: 24 × 4
   Admit    Gender Dept      n
   <chr>    <chr>  <chr> <dbl>
```

```
 1 Admitted Male   A       512
 2 Rejected Male   A       313
 3 Admitted Female A        89
 4 Rejected Female A        19
 5 Admitted Male   B       353
 6 Rejected Male   B       207
 7 Admitted Female B        17
 8 Rejected Female B         8
 9 Admitted Male   C       120
10 Rejected Male   C       205
# … with 14 more rows
```

(For privacy reasons the names of the departments have been changed to `A`, `B`, ... , `F`.)

(a) Using the tool we learned for summarizing and manipulating tidy data, create a summary table from `ucb` which shows the acceptance rate by gender. Your table should have 5 columns: `Department`, `Gender`, `Admitted`, `Rejected`, and `Proportion Admitted`. Store it in a variable called `table3a`. *1 point*

Department	Gender	Admitted	Rejected	Proportion_Admitted
A	Female	-	-	-
A	Male	-	-	-
B	Female	-	-	-
B	Male	-	-	-
C	Female	-	-	-
C	Male	-	-	-
D	Female	-	-	-
D	Male	-	-	-
E	Female	-	-	-
E	Male	-	-	-
F	Female	-	-	-
F	Male	-	-	-

(A few entries have been provided for you; your job is to write code that will produce the complete table with no blanks.)

```
# Your solution here

 table3a <- ucb %>%
   rename(Department = Dept)  %>%
   group_by(Department, Gender) %>%
   mutate(totalAppsPerGenderDept = sum(n)) %>%
   ungroup %>%
   pivot_wider(names_from = Admit, values_from = n) %>%
   mutate(Proportion_Admitted = Admitted/totalAppsPerGenderDept) %>%
   select(Department, Gender, Admitted, Rejected, Proportion_Admitted) %>%
   print
```

```
# A tibble: 12 × 5
   Department Gender Admitted Rejected Proportion_Admitted
   <chr>      <chr>     <dbl>    <dbl>               <dbl>
 1 A          Male        512      313               0.621
 2 A          Female       89       19               0.824
 3 B          Male        353      207               0.630
 4 B          Female       17        8               0.68
```

```
 5 C          Male        120       205               0.369
 6 C          Female      202       391               0.341
 7 D          Male        138       279               0.331
 8 D          Female      131       244               0.349
 9 E          Male         53       138               0.277
10 E          Female       94       299               0.239
11 F          Male         22       351               0.0590
12 F          Female       24       317               0.0704
```

(b) In STATS 250 you learned how to test for differences in proportions between two populations. Apply this to part (a) `table3a`. Was the overall proportion of men admitted statistically different from that of women? Perform an appropriate test and interpret your findings. What do these result suggest about admissions practices at UC Berkeley in the early 1970s? *1 point*

(Hint: use the `prop.test()` function.)

```
help(UCBAdmissions)
```

```
# You solution here

table3a %>%
  group_by(Gender)  %>%
  summarise(totalAdmitted = sum(Admitted), totalPopulation = sum(Admitted + Rejected))
```

A tibble: 2 × 3

Gender	totalAdmitted	totalPopulation
<chr>	<dbl>	<dbl>
Female	557	1835
Male	1198	2691

```
prop.test(x = c(557,1198), n=c(1835,2691), alternative = "two.sided")
```

```
	2-sample test for equality of proportions with continuity correction

data:  c(557, 1198) out of c(1835, 2691)
X-squared = 91.61, df = 1, p-value < 2.2e-16
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.1703022 -0.1129887
sample estimates:
   prop 1    prop 2 
0.3035422 0.4451877 
```

Since the p value is less that 5%, there is good evidence to believe that there is no difference in proportion of students admitted between the two populations (females vs males). This suggest that UC Berkley admissions in 1970's tried to accept the same proportion of males and females.

(c) Reproduce the table from Problem 1, but now stratify by department. Compute the male and female acceptance proportion for each department separately. *1 point*

Your resulting table should look like:

Dept	Female_Admitted	Female_Rejected	Male_Admitted	Male_Rejected	Male_Proportion_Admitted	Female_Proportion_Admitted
A	89	-	-	-	-	-
B	-	-	353	-	-	-
C	-	391	-	-	-	-
D	-	-	-	-	0.33093525	-
E	-	-	-	138	-	-
F	-	-	-	-	-	0.07038123

(Again, a few table entries have been provided to help you check your work, and it is your job to provide code that computes the entire table automatically.)

```
# Your solution here

table3a %>%
  select(-Proportion_Admitted) %>%
  rename(Dept = Department) %>%
  pivot_wider(names_from = "Gender", values_from  = c("Admitted", "Rejected"), names_glue = "{Gender}_{.value}") %>%
  group_by(Dept) %>%
  mutate(Male_Proportion_Admitted = Male_Admitted/(Male_Admitted+Male_Rejected), Female_Proportion_Admitted =  Female_Admitted/(Female_Admitted+Female_Rejected) )%>%
  select(Dept, Female_Admitted, Female_Rejected, Male_Admitted, Male_Rejected, Male_Proportion_Admitted, Female_Proportion_Admitted)
```

A grouped_df: 6 × 7

Dept	Female_Admitted	Female_Rejected	Male_Admitted	Male_Rejected	Male_Proportion_Admitted	Female_Proportion_Admitted
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
A	89	19	512	313	0.62060606	0.82407407
B	17	8	353	207	0.63035714	0.68000000
C	202	391	120	205	0.36923077	0.34064081
D	131	244	138	279	0.33093525	0.34933333
E	94	299	53	138	0.27748691	0.23918575
F	24	317	22	351	0.05898123	0.07038123

(d) Do the department-level findings in part (c) agree or disagree with what you concluded in part (b)? Which departments agree with your conclusion in part (b) and which disagree? **Explain with numerical evidence for full credit.** *1 point*

```
# Your solution here

#The department-level findings disagree with what I concluded in part b.
#In part c above we see that dept A has a higher proportion
#of acceptance among females than males. Specifically,
#department A had a female proportion admitted rate of 82% compared to their
#male counterparts of 62%. The remaining departments do differ slighlty between
#males and females (and differ in which gender they favor), however is not as significant as
#department A's difference.
```

Problem 3: Popular Baby Names of the Decade (2 points)

Recall from lecture the `babynames` dataset that contains a lot of information about frequency of baby names over time.

```
install.packages("babynames")
library(babynames)
```

```
Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)
```

(a) Generate a table that has **decade** on the vertical axis, and the most popular male **and** female name of each decade. A decade will be definied by the years **_0 - _**9. So for example, 1880-1889 is a decade followed by 1890-1899, etc. *1 point*

Hint: The `cut()` function can be used to "discretize" a continuous variable by placing each continuous observation into a bin. For example:

```
head(babynames)
```

A tibble: 6 × 5

year	sex	name	n	prop
<dbl>	<chr>	<chr>	<int>	<dbl>
1880	F	Mary	7065	0.07238359
1880	F	Anna	2604	0.02667896
1880	F	Emma	2003	0.02052149
1880	F	Elizabeth	1939	0.01986579
1880	F	Minnie	1746	0.01788843
1880	F	Margaret	1578	0.01616720

```
v = 1:10  # vector of the numbers 1 through 10
cut(v, breaks=c(0, 5, 10))
```

(0,5] · (0,5] · (0,5] · (0,5] · (0,5] · (5,10] · (5,10] · (5,10] · (5,10] · (5,10]
▶ **Levels**:

converts the vector $v = (1, \ldots, 10)$ into a *factor* (discrete variable) that has two levels: $(0, 5]$ and $(5, 10]$.

```
library(data.table)
```

```
Attaching package: 'data.table'


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

    between, first, last


The following object is masked from 'package:purrr':
```

transpose

```
# Your solution here

df <- babynames %>%
  mutate(decade = year - year %% 10) %>%
  group_by(decade, sex) %>%
  summarise(max = max(n), name)

dt <- as.data.table(df, TRUE)
dt
```

```
`summarise()` has grouped output by 'decade', 'sex'. You can override using the
`.groups` argument.
```

A data.table: 1924665 × 5

rn	decade	sex	max	name
<chr>	<dbl>	<chr>	<int>	<chr>
1	1880	F	11754	Mary
2	1880	F	11754	Anna
3	1880	F	11754	Emma
4	1880	F	11754	Elizabeth
5	1880	F	11754	Minnie
6	1880	F	11754	Margaret
7	1880	F	11754	Ida
8	1880	F	11754	Alice
9	1880	F	11754	Bertha
10	1880	F	11754	Sarah
11	1880	F	11754	Annie
12	1880	F	11754	Clara
13	1880	F	11754	Ella
14	1880	F	11754	Florence
15	1880	F	11754	Cora
16	1880	F	11754	Martha
17	1880	F	11754	Laura
18	1880	F	11754	Nellie
19	1880	F	11754	Grace
20	1880	F	11754	Carrie
21	1880	F	11754	Maude
22	1880	F	11754	Mabel
23	1880	F	11754	Bessie
24	1880	F	11754	Jennie
25	1880	F	11754	Gertrude
26	1880	F	11754	Julia
27	1880	F	11754	Hattie
28	1880	F	11754	Edith
29	1880	F	11754	Mattie
30	1880	F	11754	Rose
⋮	⋮	⋮	⋮	⋮
1924636	2010	M	22117	Zaver

(b) Do any names appear more than once? Write code that converts the table from part (a) into a dataframe with all the names that show up more than once. **Manual answers will not receive credit. Your code should automatically convert the table to a new one showing the duplicated names.** *1 point*

```
# Your solution here

 df <- as.data.frame(dt)
 df %>%
   count(name) %>%
    filter(n>1)
```

A data.frame: 76467 × 2

name	n
<chr>	<int>
Aaban	10
Aabha	5
Aabid	2
Aabriella	5
Aadam	26
Aadan	11
Aadarsh	17
Aaden	18
Aadesh	4
Aadhav	11
Aadhavan	6
Aadhi	5
Aadhira	6