Instructions: This assignment will give you a little more practice using what you have learned about data manipulation using the dplyr package, *specifically joining data sets.* It will also give you a chance to review previous concepts and pick up some new code!

For each problem copy and paste your R code from the R script file window into this Word document. Please use a color other than black for your R code. I do not need the code’s output, unless otherwise specified, but don’t forget to answer any additional questions presented in the problems.

You will upload this Word document to Bb when finished.

**Begin by downloading** the *fueleconomy* and *maps* packages. Then loadin the*tidyverse*, *nycflights13, Lahman, babynames, fueleconomy,* and *maps* libraries.

install.packages("fueleconomy")

install.packages("maps")

library(tidyverse)

library(nycflights13)

library(Lahman)

library(babynames)

library(fueleconomy)

library(maps)

**Next, read 13.1 through 13.4** fromChapter 13: Relational Data in the R for Data Science book. As you read you will do the exercises specified below. I have provided notes for some problems to help you or to clarify the problem. Please read these carefully.

**Section 13.2.1**

* **#1** ~Hint: Drawing the routes requires the latitude and longitude of the origin and the destination airports of each flight.

You would need to combine the flights and airports tables. From flights you would need the tailnum to make sure you have individual planes. You would also need to have the origin and dest variables so you could match the faa variable from airports. Also, from airports you would need to use that latitude and longitude of each airport. With these variables, you should be able to draw routes for each plane with the data provided

* **#3**

If weather contained data from all US airports and not just New York airports (origin), it would gain an additional relation with flights through the dest variable. Then we would be able to look at weather of the flights destination and not just its origin location.

**Section 13.3.1**

* **#2 parts 1, 2, and 4**

~Note: You are looking for a variable or set of variables that allow(s) you to uniquely identify observations. For each part verify that you have found the correct answer using *count().* See the book’s examples for this.

~BUT keep this in mind while you work – In section 13.6, the book states that when you join tables you should “start by identifying the primary key in each table. ***You should usually do this based on your understanding of the data, not empirically by looking for a combination of variables that give a unique identifier.*** If you just look for variables without thinking about what they mean, you might get (un)lucky and find a combination that’s unique in your current data but the relationship might not be true in general.”

Part 1: Lahman::Batting

playerID,yearID,stint,teamID,lgID – there is no singular variable that allows you to uniquely

identify observations

code:

Batting %>%

count(playerID,yearID,stint,teamID,lgID) %>%

filter(n>1)

output:

[1] playerID yearID stint teamID lgID n

<0 rows> (or 0-length row.names)

Part 2: babynames::babynames

name,year,sex - there is no singular variable that allows you to uniquely identify observations

code:

babynames %>%

count(name,year,sex) %>%

filter(n>1)

output:

# A tibble: 0 x 4

# ... with 4 variables: name <chr>, year <dbl>, sex <chr>, n <int>

Part 4: fueleconmony::vehicles

id

code:

vehicles %>%

count(id) %>%

filter(n>1)

output:

# A tibble: 0 x 2

# ... with 2 variables: id <dbl>, n <int>

* Refer to **#2 part 5** – Explain why there is no primary key. Then add a surrogate key to the data set using the code below. Explain the purpose of adding the surrogate key.

*Diamonds2 <- mutate(diamonds, id = row\_number())*

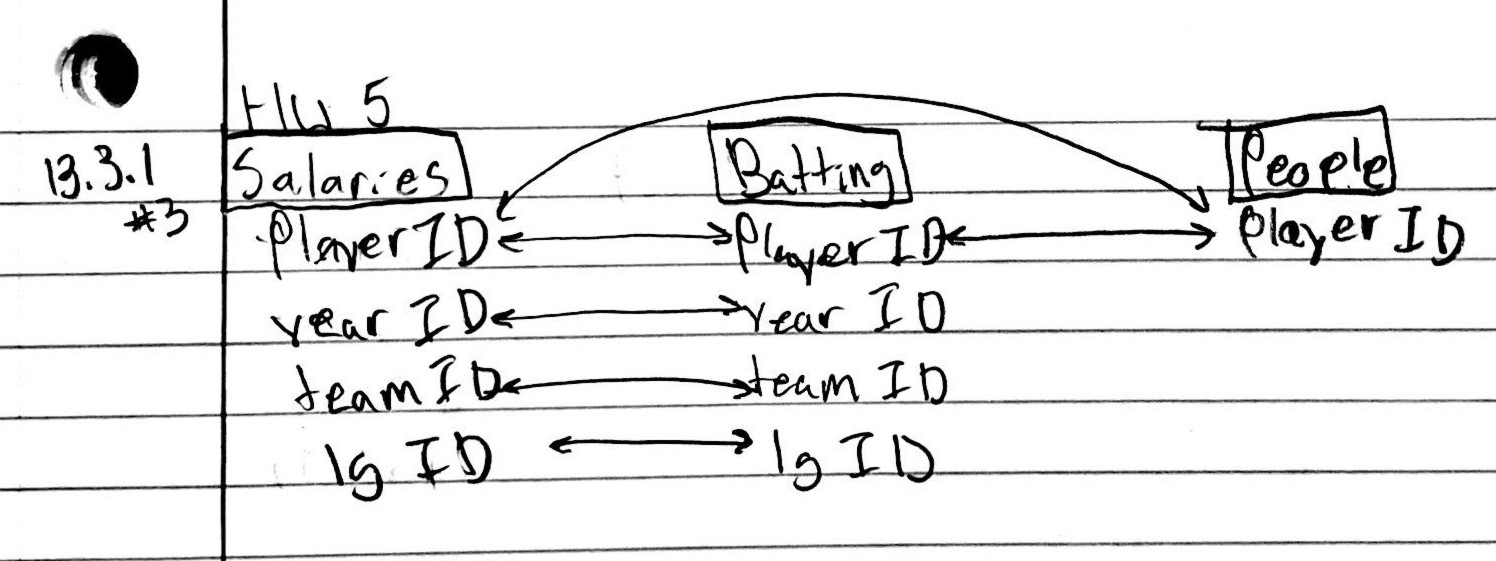
*Diamonds2*

Each row is an observation of a specific diamond’s attributes, but no combination of variables reliably identifies it. For example, there could be two diamonds that have the same attributes and there would be no real way to tell them apart. The purpose of the surrogate key is so that each diamond has their own id to be able to tell individual diamonds apart.

diamonds2<-mutate(diamonds,id=row\_number())

diamonds2

* **#3** ~Just answer the first part: Draw a diagram illustrating the connections between the *Batting*, *People*, and *Salaries*tables in the *Lahman*package.



**Section 13.4.6**

* **#1** ~ Start by using the correct *dplyr* commands to compute the average delay by destination. Then join the result to the *airports* data set using *inner\_join().* Call this new joined data set *avg\_dest\_delays*. You can then modify the book’s code to create your map! [I’ve provided the correct code below.]

*avg\_dest\_delays %>%*

*ggplot(aes(lon, lat, colour = delay)) +*

*borders("state") +*

*geom\_point() +*

*coord\_quickmap()*

flights.1<-flights %>%

filter(!is.na(dep\_delay), !is.na(arr\_delay)) %>%

mutate(total\_delay=(dep\_delay+arr\_delay)) %>%

group\_by(dest) %>%

summarize(avg\_delay=mean(total\_delay))

avg\_dest\_delays<-airports %>%

inner\_join(flights.1,by=c("faa"="dest"))

avg\_dest\_delays %>%

ggplot(aes(lon, lat, colour = avg\_delay)) +

borders("state") +

geom\_point() +

coord\_quickmap()

Chart, scatter chart

Description automatically generated

* **#3** ~ In the *planes* data set, year represents the year that the plane was manufactured. In the *flights* data set, year represents the year of the flight. You can join the two data sets and then use these two pieces of information to figure out the age of the plane at the time of the flight. After you have this, you can create a graph (using ggplot2) to explore the relationship between delays and year of manufacture. Make sure to interpret the graph and answer the original question.

flights.2<-flights %>%

filter(!is.na(dep\_delay), !is.na(arr\_delay)) %>%

mutate(total\_delay=(dep\_delay+arr\_delay)) %>%

group\_by(tailnum) %>%

summarize(avg\_delay=mean(total\_delay))

age\_vs\_delays<-planes %>%

inner\_join(flights.2,by="tailnum") %>%

filter(!is.na(year)) %>%

mutate(age=(2013-year))

ggplot(age\_vs\_delays,mapping=aes(x=age,y=avg\_delay))+

geom\_point()+

geom\_smooth(se=F)

Chart, scatter chart

Description automatically generated

The graph above shows that there is little to no relationship between age of the plane and its delays. However, there are less planes in the older range, which may mean they retire the planes before they begin to show signs of aging or cause more delays. Ultimately, the graph paints the picture that the age of a plane tends not to affect the avg time of delays. So, there is little to no relation between the two variables based on this data.

* **#4 ~** Challenge! See if you can figure this one out. Maybe you can treat it similarly to #3? Be thorough in your data exploration and answer.

#all graphs were made with this code, with the exception of changing what x equals in the ggplot

flights.3<-flights %>%

filter(!is.na(dep\_delay), !is.na(arr\_delay)) %>%

mutate(total\_delay=(dep\_delay+arr\_delay)) %>%

group\_by(day, month, year, hour, origin) %>%

summarize(avg\_delay=mean(total\_delay))

delays\_vs\_weather<-weather %>%

inner\_join(flights.3)

ggplot(delays\_vs\_weather,mapping=aes(x=precip,y=avg\_delay))+

geom\_point()+

geom\_smooth(se=F)

Wind speed vs delays humidity vs delays

Chart, scatter chart

Description automatically generated Chart, scatter chart

Description automatically generated

Temperature vs delays Dewpoint vs delays

Chart, scatter chart

Description automatically generated Chart, scatter chart

Description automatically generated

Precipitation vs delays

Chart, scatter chart

Description automatically generated

Visibility vs delays

Chart

Description automatically generated

Based on the graphs above, I would say that low visibility and high precipitation have the greatest relation with delays. However, there are slight trends that say as dewpoint, humidity, and windspeed do increase there are more delays. The one variable that seemingly has no relation to an increase of delays is temperature. Ultimately, there are so many combinations of weather that could pose a threat to flights that it is hard to single any one out. There is likely to some combination of the factors above that has a strong relation to the average delay time.

**Extra Credit:** Read section 13.5 and then answer #2 from 13.5.1.

To receive the full amount of extra credit you must answer this question in two different ways:

1. Using a *semi\_join().* See their example!

flights\_100<-flights %>%

filter(!is.na(tailnum)) %>%

count(tailnum, sort=T) %>%

filter(n>=100)

ec.1<-flights %>%

semi\_join(flights\_100)

1. Using *group\_by()* with *mutate()* instead. No joins!

[Note: We’ve always used *group\_by()* with *summarise(),* so this is new!]

~Hint: You’ll need to filter out flights that are missing a tail number, otherwise they will all be treated as a single plane.

flights\_100<-flights %>%

filter(!is.na(tailnum)) %>%

count(tailnum, sort=T) %>%

filter(n>=100)

ec.2<-flights%>%

group\_by(tailnum) %>%

mutate(f.100=ifelse(tailnum %in% flights\_100$tailnum,1,0)) %>%

filter(f.100==1)