**תרגיל 1.9**

1. אנו יודעים שחלק מימות השנה הם “ימים מיוחדים” ומספר האנשים שטסים בימים אלה שונה משמעותית מימי שיגרה. כיצד תייצג את המידע הזה ב- data.frame? מה יהיו העמודות ה-keys שיאפשרו שידוך המידע הזה לנתוני ה-flights? כיצד תחבר זאת לטבלת הנתונים?
2. הוסף לכל יעד נסיעה (origin and destination) את המיקומים שלו (lat and long) מתוך טבלת ה- airports. שים לב שעליך לשדך פעמיים את אותם הנתונים. מה תעשה עם שמות המשתנים?
3. האם יש קשר בין גיל המטוס לבין האיחורים שלו?
4. מה זה אומר עבור טיסה שיהיה לה ערך חסר ב- tailnum? מה משותף בין הערך החסר של ה tailnum לבין מטוסים אשר ה tailnum שלהם לא קיים בטבלת ה planes?
5. סנן טיסות עבור מטוסים שטסו 100 פעמים לפחות.
6. מצא את 48 השעות בהן היו האיחורים הגרועים ביותר. הצלב נתונים אלה עם טבלת הנתונים weather. מה מצאת?

1. I would add a table of special dates, similar to the following table.

special\_days <- **tribble**(

~year, ~month, ~day, ~holiday,

2013, 01, 01, "New Years Day",

2013, 07, 04, "Independence Day",

2013, 11, 29, "Thanksgiving Day",

2013, 12, 25, "Christmas Day"

)

The primary key of the table would be the (year, month, day) columns. The (year, month, day) columns could be used to join special\_days with other tables.

2. You can perform one join after another. If duplicate variables are found, by default, dplyr will distinguish the two by adding .x, and .y to the ends of the variable names to solve naming conflicts.

airport\_locations <- airports %>%

**select**(faa, lat, lon)

flights %>%

**select**(year:day, hour, origin, dest) %>%

**left\_join**(

airport\_locations,

by = **c**("origin" = "faa")

) %>%

**left\_join**(

airport\_locations,

by = **c**("dest" = "faa")

)

The suffix argument overrides this default behavior. Since is always good practice to have clear variable names, I will use the suffixes "\_dest" and "\_origin" to specify whether the column refers to the destination or origin airport.

airport\_locations <- airports %>%

**select**(faa, lat, lon)

flights %>%

**select**(year:day, hour, origin, dest) %>%

**left\_join**(

airport\_locations,

by = **c**("origin" = "faa")

) %>%

**left\_join**(

airport\_locations,

by = **c**("dest" = "faa"),

suffix = **c**("\_origin", "\_dest")

*# existing lat and lon variables in tibble gain the \_origin suffix*

*# new lat and lon variables are given \_dest suffix*

)

3. The question does not specify whether the relationship is with departure delay or arrival delay. I will look at both.

To compare the age of the plane to flights delay, I merge flights with the planes, which contains a variable plane\_year, with the year in which the plane was built. To look at the relationship between plane age and departure delay, I will calculate the average arrival and departure delay for each age of a flight. Since there are few planes older than 25 years, so I truncate age at 25 years.

plane\_cohorts <- **inner\_join**(flights,

**select**(planes, tailnum, plane\_year = year),

by = "tailnum"

) %>%

**mutate**(age = year - plane\_year) %>%

**filter**(!**is.na**(age)) %>%

**mutate**(age = **if\_else**(age > 25, 25L, age)) %>%

**group\_by**(age) %>%

**summarise**(

dep\_delay\_mean = **mean**(dep\_delay, na.rm = TRUE),

dep\_delay\_sd = **sd**(dep\_delay, na.rm = TRUE),

arr\_delay\_mean = **mean**(arr\_delay, na.rm = TRUE),

arr\_delay\_sd = **sd**(arr\_delay, na.rm = TRUE),

n\_arr\_delay = **sum**(!**is.na**(arr\_delay)),

n\_dep\_delay = **sum**(!**is.na**(dep\_delay))

)

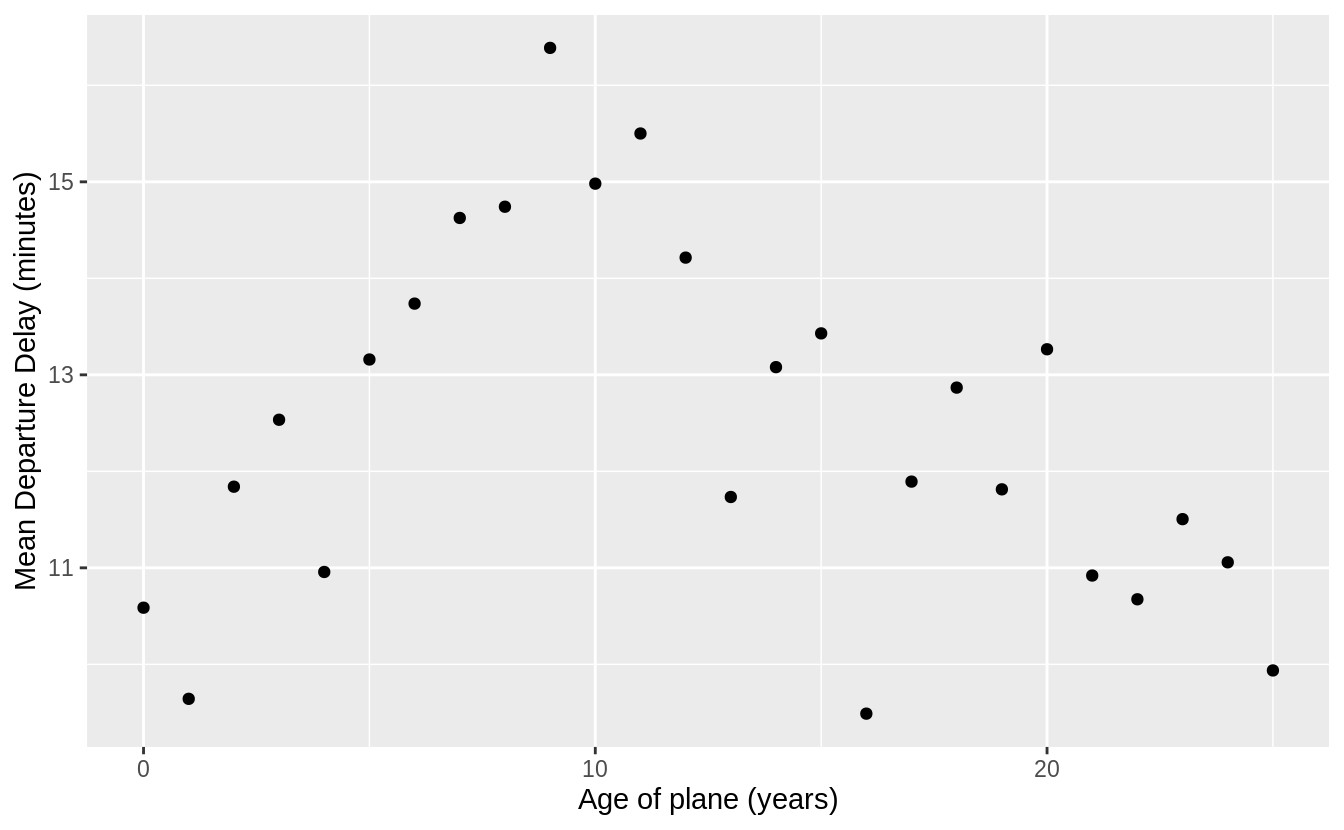
I will look for a relationship between departure delay and age by plotting age against the average departure delay. The average departure delay is increasing for planes with ages up until 10 years. After that the departure delay decreases or levels off. The decrease in departure delay could be because older planes with many mechanical issues are removed from service or because air lines schedule these planes with enough time so that mechanical issues do not delay them.

**ggplot**(plane\_cohorts, **aes**(x = age, y = dep\_delay\_mean)) +

**geom\_point**() +

**scale\_x\_continuous**("Age of plane (years)", breaks = **seq**(0, 30, by = 10)) +

**scale\_y\_continuous**("Mean Departure Delay (minutes)")



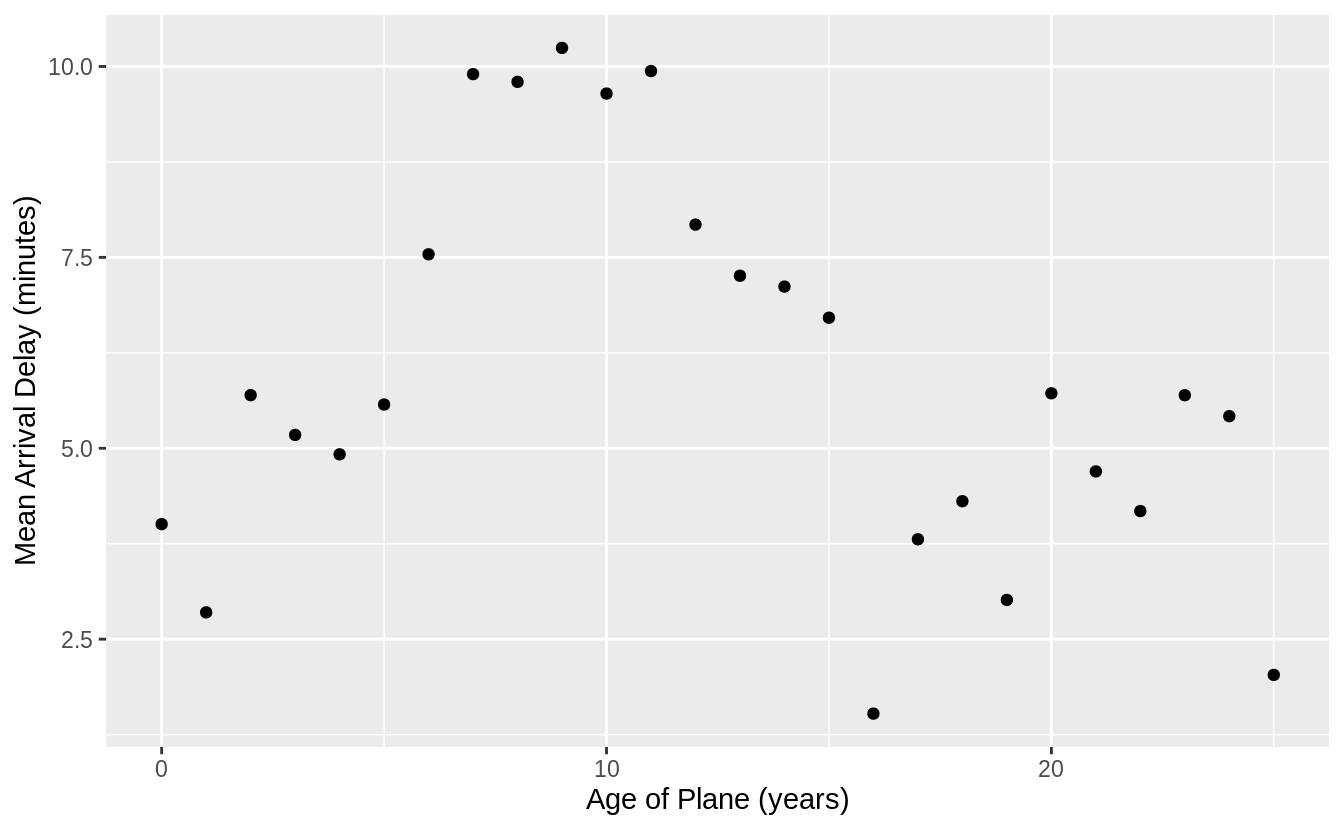
There is a similar relationship in arrival delays. Delays increase with the age of the plane until ten years, then it declines and flattens out.

**ggplot**(plane\_cohorts, **aes**(x = age, y = arr\_delay\_mean)) +

**geom\_point**() +

**scale\_x\_continuous**("Age of Plane (years)", breaks = **seq**(0, 30, by = 10)) +

**scale\_y\_continuous**("Mean Arrival Delay (minutes)")



4. Flights that have a missing tailnum all have missing values of arr\_time, meaning that the flight was canceled.

flights %>%

**filter**(**is.na**(tailnum), !**is.na**(arr\_time)) %>%

**nrow**()

Many of the tail numbers that don’t have a matching value in planes are registered to American Airlines (AA) or Envoy Airlines (MQ). The documentation for planes states:

"American Airways (AA) and Envoy Air (MQ) report fleet numbers rather than tail numbers so can’t be matched".

flights %>%

**anti\_join**(planes, by = "tailnum") %>%

**count**(carrier, sort = TRUE) %>%

**mutate**(p = n / **sum**(n))

However, not all tail numbers appearing in flights from these carriers are missing from the planes table. I don’t know how to reconcile this discrepancy.

flights %>%

**distinct**(carrier, tailnum) %>%

**left\_join**(planes, by = "tailnum") %>%

**group\_by**(carrier) %>%

**summarise**(total\_planes = **n**(),

not\_in\_planes = **sum**(**is.na**(model))) %>%

**mutate**(missing\_pct = not\_in\_planes / total\_planes) %>%

**arrange**(**desc**(missing\_pct))

5. First, I find all planes that have flown at least 100 flights. I need to filter flights that are missing a tail number otherwise all flights missing a tail number will be treated as a single plane.

planes\_gte100 <- flights %>%

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

**group\_by**(tailnum) %>%

**count**() %>%

**filter**(n >= 100)

Now, I will semi join the data frame of planes that have flown at least 100 flights to the data frame of flights to select the flights by those planes.

flights %>%

**semi\_join**(planes\_gte100, by = "tailnum")

This can also be answered with a grouped mutate.

flights %>%

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

**group\_by**(tailnum) %>%

**mutate**(n = **n**()) %>%

**filter**(n >= 100)

6. I will start by clarifying how I will be measuring the concepts in the question. There are three concepts that need to be defined more precisely.

1. What is meant by “delay”? I will use departure delay. Since the weather data only contains data for the New York City airports, and departure delays will be more sensitive to New York City weather conditions than arrival delays.
2. What is meant by “worst”? I define worst delay as the average departure delay per flight for flights *scheduled* to depart in that hour. For hour, I will use the scheduled departure time rather than the actual departure time. If planes are delayed due to weather conditions, the weather conditions during the scheduled time are more important than the actual departure time, at which point, the weather could have improved.
3. What is meant by “48 hours over the course of the year”? This could mean two days, a span of 48 contiguous hours, or 48 hours that are not necessarily contiguous hours. I will find 48 not-necessarily contiguous hours. That definition makes better use of the methods introduced in this section and chapter.
4. What is the unit of analysis? Although the question mentions only hours, I will use airport hours. The weather dataset has an observation for each airport for each hour. Since all the departure airports are in the vicinity of New York City, their weather should be similar, it will not be the same.

First, I need to find the 48 hours with the worst delays. I group flights by hour of scheduled departure time and calculate the average delay. Then I select the 48 observations (hours) with the highest average delay.

worst\_hours <- flights %>%

**mutate**(hour = sched\_dep\_time %/% 100) %>%

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

**summarise**(dep\_delay = **mean**(dep\_delay, na.rm = TRUE)) %>%

**ungroup**() %>%

**arrange**(**desc**(dep\_delay)) %>%

**slice**(1:48)

Then I can use semi\_join() to get the weather for these hours.

weather\_most\_delayed <- **semi\_join**(weather, worst\_hours,

by = **c**("origin", "year",

"month", "day", "hour"))

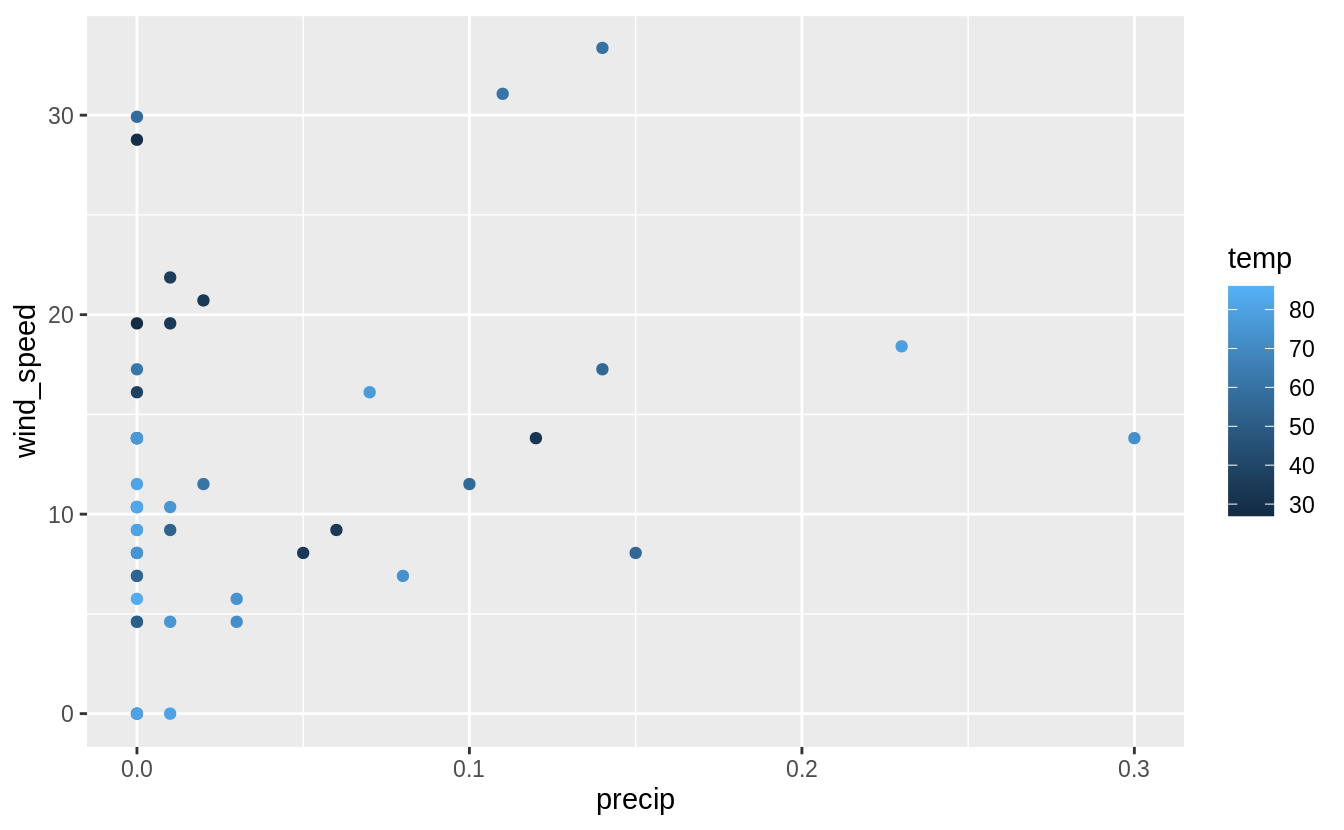
For weather, I’ll focus on precipitation, wind speed, and temperature. I will display these in both a table and a plot.  
Many of these observations have a higher than average wind speed (10 mph) or some precipitation. However, I would have expected the weather for the hours with the worst delays to be much worse.

**select**(weather\_most\_delayed, temp, wind\_speed, precip) %>%

**print**(n = 48)

**ggplot**(weather\_most\_delayed, **aes**(x = precip, y = wind\_speed, color = temp)) +

**geom\_point**()



It’s hard to say much more than that without using the tools from [Exploratory Data Analysis](https://r4ds.had.co.nz/exploratory-data-analysis.html#covariation) section to look for covariation between weather and flight delays using all flights. Implicitly in my informal analysis of trends in weather using only the 48 hours with the worst delays, I was comparing the weather in these hours to some belief I had about what constitutes “normal” or “good” weather. It would be better to actually use data to make that comparison.