LAB-4

NAME: JYOTHI K C

REGNO:19BDS0144

COURSECODE: CSE3046

SLOT: L21+L22

Use “nycflights13" data set- library ("nycflights13")

**Exercises:**

1)Find all flights that

a. Had an arrival delay of two or more hours

**CODE:**

install.packages("tidyverse")

library("nycflights13")

library("tidyverse")

filter(flights,arr\_delay >= 120)

**OUTPUT:**

**Graphical user interface

Description automatically generated with medium confidence**

b. Flew to Houston (IAH or HOU)

CODE:

filter(flights, dest == "IAH" | dest == "HOU")  
OUTPUT:

**Graphical user interface, text, application

Description automatically generated**

c. Were operated by United, American, or Delta

**CODE**:  
select(flights,carrier)

filter(flights,carrier %in% c("AA","DL","UA"))

**OUTPUT**:

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

d. Departed in summer (July, August, and September)

**CODE:**  
colnames(flights)

str(flights)

filter(flights,month %in% c(7,8,9))  
**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

e. Arrived more than two hours late, but didn’t leave late

**CODE:**

filter(flights,arr\_delay>=120 & dep\_delay<=0)  
**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

f. Were delayed by at least an hour, but made up over 30 minutes in flight.

**CODE:**  
filter(flights, dep\_delay >= 60, dep\_delay - arr\_delay > 30)  
**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

g. Departed between midnight and 6 am (inclusive)

**CODE:**

filter(flights, dep\_time %% 2400 <= 600)  
**OUTPUT:**  
Graphical user interface, application

Description automatically generated

2) Another useful dplyr filtering helper is between(). What does it do? Can you use it to simplify the code needed to answer the previous challenges?

The between() is a shortcut equivalent for x >= left & x <= right, where left and right are scalar limits.

**CODE:**

filter(flights, between(month, 7, 9))  
**OUTPUT:**

Graphical user interface, text, application, email

Description automatically generated

3) How many flights have a missing dep\_time? What other variables are missing? What might these rows represent?Find the rows of flights with a missing departure time (dep\_time) using the is.na() function.

**CODE**:

filter(flights, is.na(dep\_time))

summary(flights)  
**OUTPUT**:

Graphical user interface, text, application, email

Description automatically generated

4) How could you use arrange() to sort all missing values to the start? (Hint: use is.na()).

**CODE:**

arrange(flights, dep\_time) %>% tail()

arrange(flights, desc(is.na(dep\_time)), dep\_time)  
**OUTPUT:**

Graphical user interface, text, application, email

Description automatically generated

5)Sort flights to find the most delayed flights. Find the flights that left earliest. Hint-Find the most delayed flights by sorting the table by departure delay, dep\_delay, in descending order.

CODE:

arrange(flights, desc(dep\_delay))

arrange(flights, dep\_delay  
**OUTPUT:**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

6)Sort flights to find the fastest flights.

**CODE:**

head(arrange(flights, air\_time))

head(arrange(flights, desc(distance / air\_time)))  
**OUTPUT:**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

7)select dep\_time, dep\_delay, arr\_time, and arr\_delay from flights.

CODE:

select(flights, starts\_with("dep\_"), starts\_with("arr\_"))  
**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

8)Currently dep\_time and sched\_dep\_time are convenient to look at, but hard to compute with because they’re not really continuous numbers. Convert them to a more convenient representation of number of minutes since midnight. (hint: Add new variables with mutate()

**CODE:**

flights\_times <- mutate(flights,dep\_time\_mins = (dep\_time %/% 100 \* 60 + dep\_time %% 100)%% 1440,sched\_dep\_time\_mins = (sched\_dep\_time %/% 100 \* 60 +

sched\_dep\_time %% 100) %% 1440)

select(flights\_times, dep\_time, dep\_time\_mins, sched\_dep\_time,sched\_dep\_time\_mins)  
**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

9)For each destination, compute the total minutes of delay. For each flight, compute the proportion of the total delay for its destination.

CODE:

flights %>%

filter(arr\_delay > 0) %>%

group\_by(dest, origin, carrier, flight) %>%

summarise(arr\_delay = sum(arr\_delay)) %>%

group\_by(dest) %>%

mutate(

arr\_delay\_prop = arr\_delay / sum(arr\_delay)

) %>%

arrange(dest, desc(arr\_delay\_prop)) %>%

select(carrier, flight, origin, dest, arr\_delay\_prop)  
**OUTPUT:**

Graphical user interface, text, application, email

Description automatically generated