**Solution to Homework 6**

**Sample #1**

#Load in the necessary libraries

library(tidyverse)

library(nycflights13)

library(dplyr)

library(ggplot2)

library(maps)

# Rename the data sets

airlines <- airlines

flights <- flights

airports <- airports

planes <- planes

weather <- weather

**# PART 1**

**# QUESTION 1:** Compute the average delay by destination, then join on the airports data frame so you can show the spatial distribution of delays.

# Calculate the average delay by destination

avg\_delay <- flights %>%

group\_by(dest) %>%

summarize(avg\_delay = mean(arr\_delay, na.rm = TRUE))

# Join with the airports data to get locations

delay\_airports <- airports %>%

semi\_join(flights, c("faa" = "dest")) %>%

left\_join(avg\_delay, by = c("faa" = "dest"))

# Plotting the map with average delay represented by color and size

ggplot(data = delay\_airports, aes(x = lon, y = lat)) +

borders("state") + # This adds state borders

geom\_point(aes(size = avg\_delay, color = avg\_delay)) + # Solid points sized and colored by delay

scale\_size(range = c(1, 3), name = "Average delay (min)") +

scale\_color\_viridis\_c(option = "C") + # A color scale

coord\_quickmap() + # This sets the aspect ratio correctly for the map

theme\_minimal() + # Minimal theme

labs(title = "Average Flight Delays by Destination", x = "Longitude", y = "Latitude") +

theme(legend.position = "bottom") # Move legend to the bottom

# QUESTION 2: Is there a relationship between the age of a plane and its delays?

# Perform an inner join on flights and planes datasets to combine them based on the 'tailnum' column.

# This will allow us to associate the year the plane was made ('plane\_year') with each flight.

plane\_cohorts <- inner\_join(flights,

select(planes, tailnum, plane\_year = year), # Select only the tailnum and year columns from planes, renaming year to plane\_year

by = "tailnum" # Specify the joining column

) %>%

# Calculate the age of the plane by subtracting the plane\_year from the year of the flight

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

# Remove rows where the age is NA (i.e., missing data)

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

# Cap the age at 25 years, changing any age value over 25 to 25

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

# Group the data by the age of the plane

group\_by(age) %>%

# Calculate summary statistics for departure and arrival delays for each age group

summarise(

dep\_delay\_mean = mean(dep\_delay, na.rm = TRUE), # Mean departure delay, excluding NA values

dep\_delay\_sd = sd(dep\_delay, na.rm = TRUE), # Standard deviation of departure delay, excluding NA values

arr\_delay\_mean = mean(arr\_delay, na.rm = TRUE), # Mean arrival delay, excluding NA values

arr\_delay\_sd = sd(arr\_delay, na.rm = TRUE), # Standard deviation of arrival delay, excluding NA values

n\_arr\_delay = sum(!is.na(arr\_delay)), # Count of non-NA arrival delay values

n\_dep\_delay = sum(!is.na(dep\_delay)) # Count of non-NA departure delay values

)

# Visualizing departure delays by age of plane

# Departure Delays

ggplot(plane\_cohorts, aes(x = age, y = dep\_delay\_mean)) + # Plot the mean departure delay against the age of the plane

geom\_point() + # Use points to represent each age group

scale\_x\_continuous("Age of plane (years)", breaks = seq(0, 30, by = 10)) + # Set the x-axis scale for age of plane

scale\_y\_continuous("Mean Departure Delay (minutes)") # Set the y-axis scale for mean departure delay

# Visualizing arrival delays by age of plane

# Arrival Delays

ggplot(plane\_cohorts, aes(x = age, y = arr\_delay\_mean)) + # Plot the mean arrival delay against the age of the plane

geom\_point() + # Use points to represent each age group

scale\_x\_continuous("Age of Plane (years)", breaks = seq(0, 30, by = 10)) + # Set the x-axis scale for age of plane

scale\_y\_continuous("Mean Arrival Delay (minutes)") # Set the y-axis scale for mean arrival delay

# QUESTION 3: Filter flights to only show flights with planes that have flown at least 100 flights.

# Count the number of flights for each plane

plane\_flights <- flights %>%

group\_by(tailnum) %>%

summarise(n\_flights = n()) %>%

filter(n\_flights >= 100)

# Filter flights for planes that have flown at least 100 flights

flights\_filtered <- flights %>%

filter(tailnum %in% plane\_flights$tailnum)

# Printing the result

print(flights\_filtered)

# PART 2

# Load in lubridate

library(lubridate)

# QUESTION 4: Compare airtime with the duration between departure and arrival.

# Define a function to create date-time objects from the flight times

make\_datetime\_100 <- function(year, month, day, time) {

make\_datetime(year, month, day, time %/% 100, time %% 100)

}

# Apply the function to create date-time variables for departure and arrival

flights\_dt <- flights %>%

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

mutate(

dep\_datetime = make\_datetime\_100(year, month, day, dep\_time),

arr\_datetime = make\_datetime\_100(year, month, day, arr\_time),

sched\_dep\_datetime = make\_datetime\_100(year, month, day, sched\_dep\_time),

sched\_arr\_datetime = make\_datetime\_100(year, month, day, sched\_arr\_time),

airtime\_minutes = air\_time, # air\_time is already in minutes

# Calculate the duration between the actual departure and arrival

duration\_minutes = as.numeric(difftime(arr\_datetime, dep\_datetime, units = "mins"))

) %>%

# will only plot flights with reasonable duration and airtime to avoid clutter

filter(duration\_minutes > 0, duration\_minutes < 500, airtime\_minutes < 500)

# Use ggplot2 to create an enhanced scatter plot

ggplot(flights\_dt, aes(x = airtime\_minutes, y = duration\_minutes)) +

geom\_jitter(alpha = 0.4, color = "dodgerblue", size = 1.5) + # Use geom\_jitter for better visibility of overlapping points

geom\_smooth(method = "lm", se = FALSE, color = "darkorange", size = 1) + # Add a fitted line without the confidence interval

theme\_minimal() + # Use a minimal theme

theme(plot.title = element\_text(hjust = 0.5)) + # Center the plot title

labs(

title = "Comparison of Scheduled Airtime and Actual Flight Duration",

x = "Scheduled Airtime (minutes)",

y = "Actual Flight Duration (minutes)",

caption = "Data from nycflights13"

)

# QUESTION 5: On what day of the week should you leave if you want to minimize the chance of a delay?

# Calculate average delay by day of the week

day\_of\_week\_delay <- flights %>%

mutate(day\_of\_week = wday(time\_hour, label = TRUE)) %>%

group\_by(day\_of\_week) %>%

summarise(avg\_delay = mean(arr\_delay, na.rm = TRUE)) %>%

arrange(avg\_delay)

# Plotting the average delay by day of the week

ggplot(day\_of\_week\_delay, aes(x = day\_of\_week, y = avg\_delay)) +

geom\_col(fill = "steelblue") + # Use bars to display the average delay

labs(title = "Average Delay by Day of the Week", x = "Day of the Week", y = "Average Delay (minutes)")

**Sample #2**

library(tidyverse)

library(nycflights13)

library(lubridate)

nyc <- nycflights13::flights

#..........................................................

#Part 1: Relational data

#Here you will start by reproducing some of the steps in Sec 13.1 – 13.5 of the

#textbook

#.........................................................

#QUESTION 1: Compute the average delay by destination, then join on the airports data

#frame so you can show the spatial distribution of delays

# Calculate Average Delay by Destination

average\_delays <- flights %>%

group\_by(dest) %>%

summarize(average\_delay = mean(dep\_delay, na.rm = TRUE))

#Join with Airports Data

airport\_delays <- airports %>%

semi\_join(flights, by = c("faa" = "dest")) %>%

left\_join(average\_delays, by = c("faa" = "dest"))

#Plotting

ggplot(airport\_delays, aes(x = lon, y = lat, color = average\_delay)) +

borders("state") +

geom\_point(alpha = 0.7) +

coord\_quickmap() +

labs(title = "Average Flight Delays by US Airport",

x = "Longitude",

y = "Latitude",

color = "Average Delay") +

theme\_minimal()

#..................................................................

#QUESTION 2: Is there a relationship between the age of a plane and its delays?

#Write code to (how a plot that) support(s) your conclusions.

# The provided code for calculating the relationship between plane age and delays

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))

)

#> `summarise()` ungrouping output (override with `.groups` argument)

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)")

#QUESTION 3: Filter flights to only show flights with planes that have flown at least 100

#flights

# Count the number of flights for each plane

flight\_counts <- flights %>%

group\_by(tailnum) %>%

summarize(flight\_count = n())

# Filter for planes with at least 100 flights

frequent\_flying\_planes <- flight\_counts %>%

filter(flight\_count >= 100)

# Filter the original flights dataset to include only those flights

flights\_with\_frequent\_fliers <- flights %>%

semi\_join(frequent\_flying\_planes, by = "tailnum")

# Display the resulting tibble

flights\_with\_frequent\_fliers

#...........................................................................

**#Part 2: Dates and times**

#..........................................................................

#QUESTION 4: Compare airtime with the duration between departure and arrival.

#Write code and/or produce plots to explain your findings.

#(Hint: consider the location of the airport.)

# Function to create datetime objects from the given year, month, day, and time

make\_datetime\_100 <- function(year, month, day, time) {

make\_datetime(year, month, day, time %/% 100, time %% 100)

}

# Transforming flight times into datetime objects and calculating durations

flights\_dt <- flights %>%

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

mutate(

dep\_time = make\_datetime\_100(year, month, day, dep\_time),

arr\_time = make\_datetime\_100(year, month, day, arr\_time),

sched\_dep\_time = make\_datetime\_100(year, month, day, sched\_dep\_time),

sched\_arr\_time = make\_datetime\_100(year, month, day, sched\_arr\_time)

) %>%

select(origin, dest, ends\_with("delay"), ends\_with("time"))

# Calculating the difference between the flight duration and airtime

flights\_dt %>%

mutate(

flight\_duration = as.numeric(arr\_time - dep\_time),

air\_time\_mins = air\_time,

diff = flight\_duration - air\_time\_mins

) %>%

select(origin, dest, flight\_duration, air\_time\_mins, diff)

#.......................................................................

#QUESTION 5: On what day of the week should you leave if you want to minimize the

#chance of a delay? Write code and/or produce plots to explain your findings.

flights\_dt %>%

mutate(dow = wday(sched\_dep\_time)) %>%

group\_by(dow) %>%

summarise(

dep\_delay = mean(dep\_delay),

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

) %>%

print(n = Inf)

# Analyzing delays by day of week

delay\_summary<-flights\_dt %>%

mutate(wday = wday(dep\_time, label = TRUE)) %>%

group\_by(wday) %>%

summarize(ave\_dep\_delay = mean(dep\_delay, na.rm = TRUE))

# Plotting average departure delays by day of week

ggplot(delay\_summary, aes(x = wday, y = ave\_dep\_delay)) +

geom\_bar(stat = "identity", fill = "steelblue") +

labs(

title = "Average Departure Delays by Day of the Week",

x = "Day of the Week",

y = "Average Departure Delay (minutes)"

) +

theme\_minimal()

# Creating a summary of average arrival delays by day of week

arrival\_delay\_summary <- flights\_dt %>%

mutate(wday = wday(arr\_time, label = TRUE)) %>%

group\_by(wday) %>%

summarize(ave\_arr\_delay = mean(arr\_delay, na.rm = TRUE))

#plotting the average arrival delays by day of week

ggplot(arrival\_delay\_summary, aes(x = wday, y = ave\_arr\_delay)) +

geom\_bar(stat = "identity", fill = "coral") +

labs(

title = "Average Arrival Delays by Day of the Week",

x = "Day of the Week",

y = "Average Arrival Delay (minutes)"

) +

theme\_minimal()