ECO394D Probability and Statistics Homework 2

Sibo Ding

Summer 2023

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

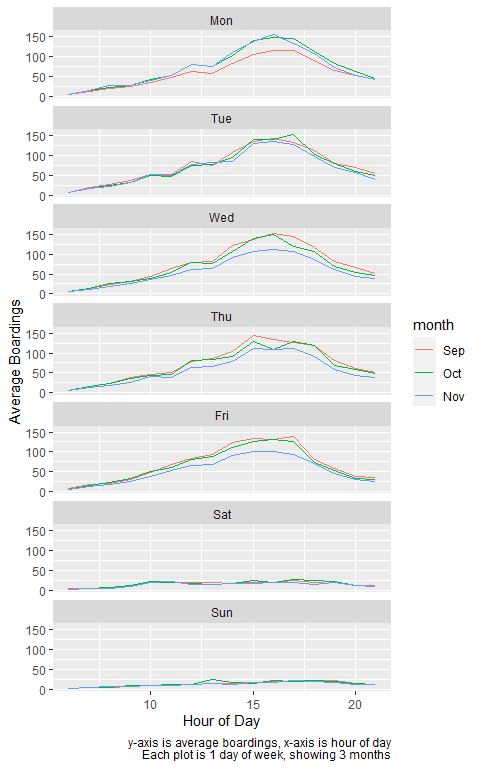
library(ggplot2)

## Problem 1

### 1

metro <- read.csv("capmetro\_UT.csv")  
  
# Recode the categorical variables in sensible, rather than alphabetical, order  
metro <- metro |> mutate(  
 day\_of\_week = factor(day\_of\_week, levels = c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")),  
 month = factor(month, levels = c("Sep", "Oct", "Nov")))  
  
metro |> group\_by(month, day\_of\_week, hour\_of\_day) |>  
 summarize(avg\_boarding = mean(boarding)) |>  
 ggplot(aes(hour\_of\_day, avg\_boarding, col = month)) +  
 geom\_line() +  
 facet\_wrap(~ day\_of\_week, nrow = 7) +  
 labs(x = "Hour of Day",  
 y = "Average Boardings",  
 caption = "y-axis is average boardings, x-axis is hour of day  
 Each plot is 1 day of week, showing 3 months")

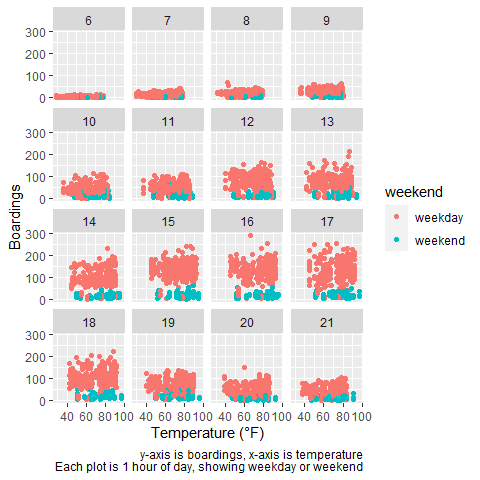
## `summarise()` has grouped output by 'month', 'day\_of\_week'. You can override  
## using the `.groups` argument.



The hour of peak boardings changes from day to day, as workdays have peaks but weekends do not.  
UT students have fewer class on Monday in September.  
Similarly, UT students have fewer class on Wed/Thur/Fri in November.

### 2

metro |> ggplot(aes(temperature, boarding, col = weekend)) +  
 geom\_point() +  
 facet\_wrap(~ hour\_of\_day) +  
 labs(x = "Temperature (°F)",  
 y = "Boardings",  
 caption = "y-axis is boardings, x-axis is temperature  
 Each plot is 1 hour of day, showing weekday or weekend")



Holding hour of day and weekend status constant, temperature does not have a noticeable effect on the number of UT students riding the bus, as different temperatures have similar boardings (data is somewhat rectangular).

## Problem 2

### Part A

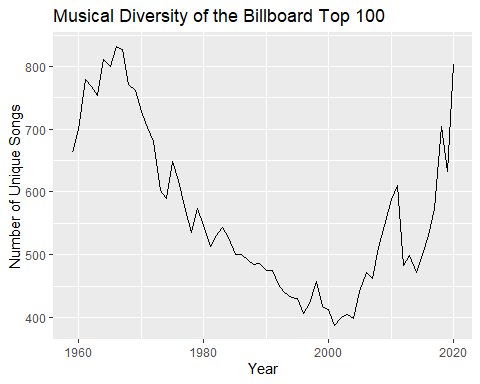
bb <- read.csv("billboard.csv")  
  
# 10 songs with highest total number of weeks  
bb |> group\_by(performer, song) |> count() |> arrange(desc(n)) |> head(10)

## # A tibble: 10 × 3  
## # Groups: performer, song [10]  
## performer song n  
## <chr> <chr> <int>  
## 1 Imagine Dragons Radioactive 87  
## 2 AWOLNATION Sail 79  
## 3 Jason Mraz I'm Yours 76  
## 4 The Weeknd Blinding Lights 76  
## 5 LeAnn Rimes How Do I Live 69  
## 6 LMFAO Featuring Lauren Bennett & GoonRock Party Rock Anthem 68  
## 7 OneRepublic Counting Stars 68  
## 8 Adele Rolling In The Deep 65  
## 9 Jewel Foolish Games/You Were Meant… 65  
## 10 Carrie Underwood Before He Cheats 64

10 songs with highest total number of weeks on Billboard Top 100.

### Part B

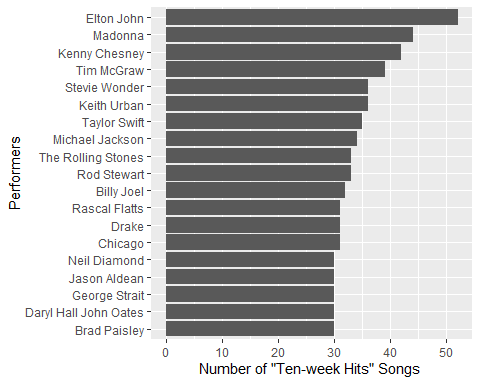
# The number of unique songs in each year  
bb |> filter(!year %in% c(1958, 2021)) |>  
 distinct(year, song\_id) |>  
 group\_by(year) |>  
 count() |>  
 ggplot(aes(year, n)) +  
 geom\_line() +  
 xlab("Year") +  
 ylab("Number of Unique Songs") +  
 ggtitle("Musical Diversity of the Billboard Top 100")



The line graph shows the number of unique songs each year.  
The number of unique songs decreased from 1970s to 2000s, increased from 2000s to 2020s.

### Part C

bb |> group\_by(performer, song) |>  
 count() |>  
 filter(n >= 10) |> # Songs on Billboard for at least 10 weeks  
 group\_by(performer) |>  
 count() |>  
 filter(n >= 30) |> # Performers who have at least 30 "10-week" songs  
 # Reorder performers based on "n"  
 ggplot(aes(forcats::fct\_reorder(performer, n), n)) +  
 geom\_col() +  
 coord\_flip() +  
 xlab("Performers") +  
 ylab("Number of \"Ten-week Hits\" Songs ")



The bar plot shows performers since 1958 who have at least 30 songs that are “ten-week hits”, with corresponding number of songs shown in descending order.

## Problem 3

### A

creat <- read.csv("creatinine.csv")  
  
lin <- lm(creatclear ~ age, data = creat)  
  
coef(lin)[1] + coef(lin)[2] \* 55

## (Intercept)   
## 113.723

### B

coef(lin)[2]

## age   
## -0.6198159

If age increases by 1 year, creatinine clearance rate decrease by 0.62 on average.

### C

# Prediction for 40-year-old  
coef(lin)[1] + coef(lin)[2] \* 40

## (Intercept)   
## 123.0203

# Prediction for 60-year-old  
coef(lin)[1] + coef(lin)[2] \* 60

## (Intercept)   
## 110.624

The 40-year-old is healthier, as it far exceeds the prediction of regression for its age.

## Problem 4

### Part A

### Part B

dice\_sum <- gtools::permutations(6, 2, repeats.allowed = TRUE) |>  
 rowSums()  
  
# P(Sum is odd)  
sum(dice\_sum %% 2 == 1) / length(dice\_sum)

## [1] 0.5

# P(Sum is less than 7)  
sum(dice\_sum < 7) / length(dice\_sum)

## [1] 0.4166667

# P(Sum is less than 7 | Sum is odd)  
sum(dice\_sum %% 2 == 1 & dice\_sum < 7) / sum(dice\_sum %% 2 == 1)

## [1] 0.3333333

These two events are not independent, as .

### Part C

### Part D

Bayes’ Theorem

### Part E

Bayes’ Theorem

## Problem 5

home <- read.csv("epl\_2018\_19\_home.csv")  
away <- read.csv("epl\_2018\_19\_away.csv")  
  
# GF: goals for; GA: goals against; GP: games played  
avg\_goal\_by\_team <- mean(home$GF + away$GF)  
avg\_goal\_home <- mean(home$GF) / mean(home$GP)  
avg\_goal\_away <- mean(away$GF) / mean(away$GP)

### 1

liv\_attack <- (  
 home |> filter(Team == "Liverpool") |> select(GF) +  
 away |> filter(Team == "Liverpool") |> select(GF)) / avg\_goal\_by\_team  
  
tot\_defense <- (  
 home |> filter(Team == "Tottenham") |> select(GA) +  
 away |> filter(Team == "Tottenham") |> select(GA)) / avg\_goal\_by\_team  
  
lambda\_liv <- avg\_goal\_home \* liv\_attack \* tot\_defense  
  
tot\_attack <- (  
 home |> filter(Team == "Tottenham") |> select(GF) +  
 away |> filter(Team == "Tottenham") |> select(GF)) / avg\_goal\_by\_team  
  
liv\_defense <- (  
 home |> filter(Team == "Liverpool") |> select(GA) +  
 away |> filter(Team == "Liverpool") |> select(GA)) / avg\_goal\_by\_team  
  
lambda\_tot <- avg\_goal\_away \* tot\_attack \* liv\_defense  
  
# Monte Carlo simulations of Poisson distribution  
set.seed(42)  
n\_sim <- 100000  
liv <- rpois(n\_sim, as.numeric(lambda\_liv))  
tot <- rpois(n\_sim, as.numeric(lambda\_tot))  
  
# Liverpool win  
sum(liv > tot) / n\_sim

## [1] 0.67117

# Draw  
sum(liv == tot) / n\_sim

## [1] 0.21034

# Liverpool lose  
sum(liv < tot) / n\_sim

## [1] 0.11849

Question: I want to predict the probabilities of win/lose/draw results between Liverpool (home) and Tottenham (away).

Approach: Monte Carlo simulation  
p.m.f. of Poisson distribution:   
Take Liverpool for example, I estimate its based on seasonal scores. Then I simulate every number of Liverpool’s goals based on the probability calculated by the p.m.f. of Poisson distribution. I do the same thing for Tottenham, assuming teams’ scores are independent. Liverpool win if Liverpool’s goals are more than Tottenham’s in a certain pair , vice versa.

Results: Liverpool have 67% probability to win, 21% probability to draw, 12% probability to lose.

Conclusion: I predict Liverpool have 67% probability to win, using Monte Carlo simulations on Poisson distribution.

### 2

mac\_attack <- (  
 home |> filter(Team == "Manchester City") |> select(GF) +  
 away |> filter(Team == "Manchester City") |> select(GF)) / avg\_goal\_by\_team  
  
ars\_defense <- (  
 home |> filter(Team == "Arsenal") |> select(GA) +  
 away |> filter(Team == "Arsenal") |> select(GA)) / avg\_goal\_by\_team  
  
lambda\_mac <- avg\_goal\_home \* mac\_attack \* ars\_defense  
  
ars\_attack <- (  
 home |> filter(Team == "Arsenal") |> select(GF) +  
 away |> filter(Team == "Arsenal") |> select(GF)) / avg\_goal\_by\_team  
  
mac\_defense <- (  
 home |> filter(Team == "Manchester City") |> select(GA) +  
 away |> filter(Team == "Manchester City") |> select(GA)) / avg\_goal\_by\_team  
  
lambda\_ars <- avg\_goal\_away \* ars\_attack \* mac\_defense  
  
# Monte Carlo simulations of Poisson distribution  
set.seed(42)  
n\_sim <- 100000  
mac <- rpois(n\_sim, as.numeric(lambda\_mac))  
ars <- rpois(n\_sim, as.numeric(lambda\_ars))  
  
# Manchester City win  
sum(mac > ars) / n\_sim

## [1] 0.77992

# Draw  
sum(mac == ars) / n\_sim

## [1] 0.1411

# Manchester City lose  
sum(mac < ars) / n\_sim

## [1] 0.07898

Question, Approach, and Conclusion are similar to previous question.

Results: Manchester City have 78% probability to win, 14% probability to draw, 8% probability to lose.