Lab 4 - IE 6200 - Sec 09 - Jordan Lian

Jordan Lian

10/25/2020

Problem Statement

My understanding of the assignments comes from the most recent chapters, where we combined the basic probability concepts to understand probability mass/density functions, continuous/cumulative distribution functions, and joint probability. I mainly used the packet to solve the tasks for the assignment, which is what I usually do to complete the lab assignments.

Output

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.4
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidyverse
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'ggplot2' was built under R version 4.0.4
```

```
## -- Conflicts ------------------------------------------------------------------------ tidyverse_conflicts()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.0.3
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.3
```

```
##
## Attaching package: 'lubridate'
```

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

```r
library(dplyr)
```

Task 1

Import the Bluebikes dataset provided along with the assignment in R and save it as a data frame.

```
## 'data.frame':    16854 obs. of  9 variables:
##  $ X                 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ starttime         : chr  "2018-06-01 00:14:10" "2018-06-01 00:17:59" "2018-06-01 00:55:10" "2018-0
##  $ stoptime          : chr  "2018-06-01 01:24:20" "2018-06-01 01:24:10" "2018-06-01 01:02:56" "2018-0
##  $ start.station.name: chr  "Back Bay" "Back Bay" "CSP" "CSP" ...
##  $ end.station.name  : chr  "Back Bay" "Back Bay" "MIT at Mass Ave / Amherst St" "Beacon St at Tappar
##  $ month             : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ date              : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ hour              : int  0 0 0 1 2 5 5 6 6 6 ...
##  $ dayofweek         : int  6 6 6 6 6 6 6 6 6 6 ...
```

Task 2

Using the data, compute the following statistics for the number of bikes that were picked up or dropped off (either one or both if needed) from any station of your choosing (except for the ones analyzed in class) between 10:00 AM and 11:00 AM

1. Probability Mass Function (PMF)
2. Continuous Distribution Function (CDF)
3. Expected Value
4. Joint Probability
5. Correlation Coefficient

Create any visualizations that you feel appropriate to convey your findings. (Example: heatmap for joint probability)

PMF and CDF

I chose to look at bikes picked up at the stop at South End Library – Tremont St at W Newton St between 10AM – 12PM. In this case, X is a random variable that represents the number of bikes picked up at South End Library – Tremont St at W Newton St. The assignment said between 10AM and 11AM, but when I generated the table, there was 1 row with PMF = 1, and CDF = 1. That is not useful for data, so I extended the time frame by 1 hour to generate at least 2 rows of data, where we got $P(X = 1)$ and $P(X = 3)$ along with $P(X < 1)$ and $P(X < 3)$. Below is a bar graph showing the number of days versus the count.

```r
# Filter out data
req_col <- select(bikes_df, start.station.name, month, date, hour)
req_row <- dplyr::filter(req_col,
                         start.station.name == 'South End Library - Tremont St at W Newton St'
                         & hour >= 10 & hour <= 11)

# Group by and summarize, pt 1
daily <- group_by(req_row, month, date, hour)
daily <- summarise(daily, count=n())
```

```
## `summarise()` regrouping output by 'month', 'date' (override with `.groups` argument)
```

```r
# Group by and summarize, pt 2
days <- group_by(daily, count)
days <- summarise(days, num_days = n())
```

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

```r
# Get new columns
pickup_pmf <- round(days$num_days/sum(days$num_days), 3)
pickup_cdf <- round(cumsum(pickup_pmf), 3)

# Final PMF and CDF Table
South_End_freq <- cbind(days, pickup_pmf = pickup_pmf, pickup_cdf = pickup_cdf)
South_End_freq
```

```
##   count num_days pickup_pmf pickup_cdf
## 1     1       31      0.969      0.969
## 2     3        1      0.031      1.000
```

Expected Value

```r
# shortcut
South_End_freq_shortcut <- bikes_df %>%
  select(start.station.name, month, date, hour) %>%
  dplyr::filter(start.station.name == 'South End Library - Tremont St at W Newton St'
                & hour >= 10 & hour <= 11) %>%
  group_by(month, date, hour) %>%
  summarise(count = n()) %>%
  group_by(count) %>%
  summarise(num_days = n()) %>%
  mutate(pickup_pmf = num_days/sum(num_days)) %>%
  mutate(pickup_cdf = cumsum(pickup_pmf))
```

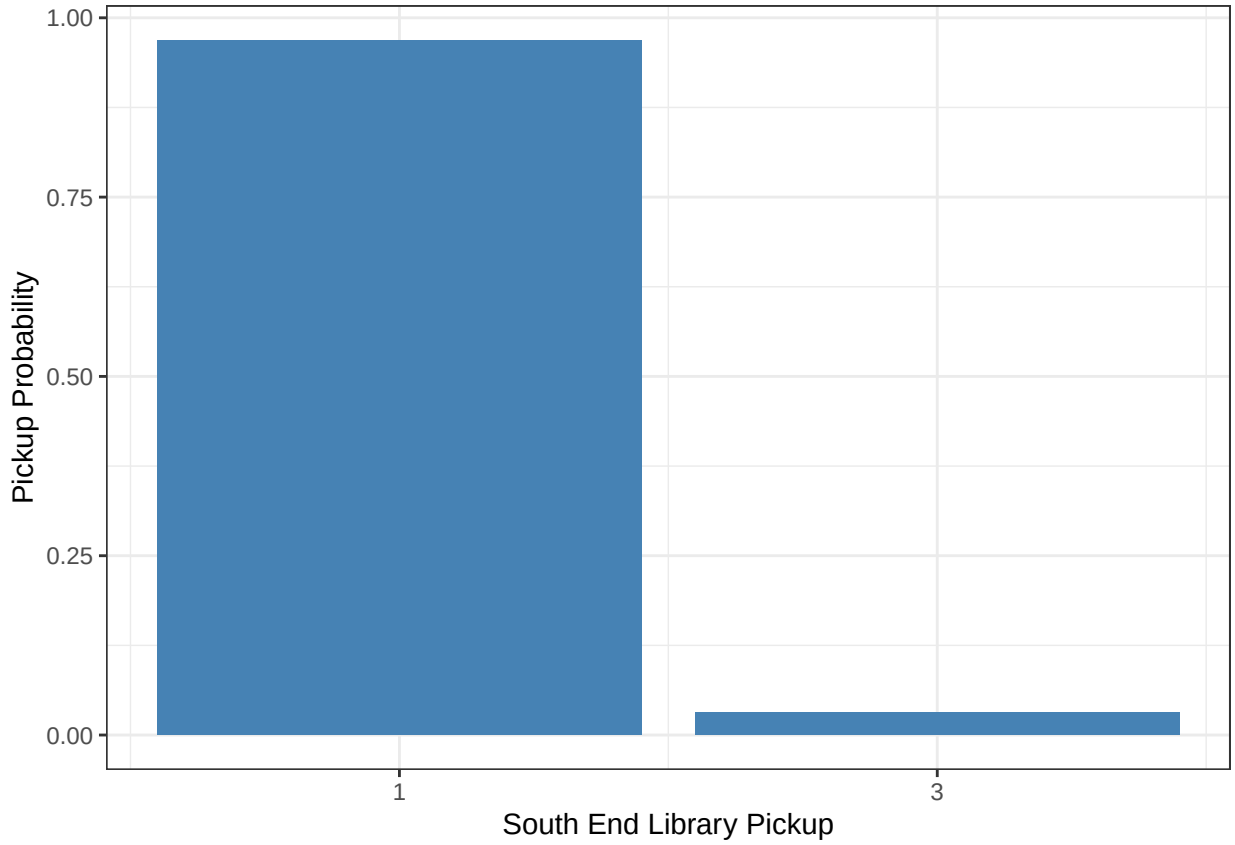
```
## `summarise()` regrouping output by 'month', 'date' (override with `.groups` argument)
```

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

```
South_End_freq_shortcut
```

```
## # A tibble: 2 x 4
##   count num_days pickup_pmf pickup_cdf
##   <int>    <int>      <dbl>      <dbl>
## 1     1       31     0.969       0.969
## 2     3        1     0.0312      1
```

```
# South End Bar Graph
South_End_visual <- ggplot(South_End_freq_shortcut, aes(count, pickup_pmf)) +
  geom_bar(stat="identity", fill="steelblue") +
  theme_bw() +
  labs(x = 'Bikes Picked Up', y = 'Pickup Probability') +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_continuous("South End Library Pickup",
                     labels = as.character(South_End_freq_shortcut$count),
                     breaks = South_End_freq_shortcut$count)
South_End_visual
```




```
# EV
weighted.mean(South_End_freq_shortcut$count, South_End_freq_shortcut$pickup_pmf)
```

```
## [1] 1.0625
```

Joint Probability

I needed another random variable to evaluate joint probability. We already have X, so for Y, I decided to make Y a random variable that represents the number of bikes dropped off at the Andrew T Stop – Dorchester Ave at Dexter St between 10AM – 12PM. First, I generated a PMF/CDF table for just Y, then I combined X and Y to create the joint frequency and subsequently the joint probability table. Lastly, I created the final frequency table to create the heatmap for joint probability.

```
Andrew_T_freq <- bikes_df %>%
  select(end.station.name, month, date, hour) %>%
  dplyr::filter(end.station.name == 'Andrew T Stop - Dorchester Ave at Dexter St'
                & hour >= 10 & hour <= 11) %>%
  group_by(month, date, hour) %>%
  summarise(count = n()) %>%
  group_by(count) %>%
  summarise(num_days = n()) %>%
  mutate(pickup_pmf = num_days/sum(num_days)) %>%
  mutate(pickup_cdf = cumsum(pickup_pmf))
```

```
## `summarise()` regrouping output by 'month', 'date' (override with `.groups` argument)
```

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

```
Andrew_T_freq
```

```
## # A tibble: 2 x 4
##   count num_days pickup_pmf pickup_cdf
##   <int>    <int>      <dbl>      <dbl>
## 1     1        1        0.5        0.5
## 2     2        1        0.5        1
```

```
# Use outer()
joint_freq <- outer(South_End_freq_shortcut$num_days, Andrew_T_freq$num_days, FUN = "+")
rownames(joint_freq) <- South_End_freq_shortcut$count
colnames(joint_freq) <- Andrew_T_freq$count
joint_freq
```

```
##    1  2
## 1 32 32
## 3  2  2
```

```
# Get joint probability values
joint_prob <- round(joint_freq/sum(joint_freq), 3)
joint_prob
```

```
##       1     2
## 1 0.471 0.471
## 3 0.029 0.029
```

```
joint_df <- melt(joint_freq)
colnames(joint_df) <- c('South_End_Pickup', 'Andrew_T_Dropoff', 'Frequency')
joint_df
```

```
##   South_End_Pickup Andrew_T_Dropoff Frequency
## 1                1                1        32
## 2                3                1         2
## 3                1                2        32
## 4                3                2         2
```

Correlation Coefficient

```
# Scatter Plot
ggplot(data = joint_df, aes(x=South_End_Pickup, y=Andrew_T_Dropoff)) +
  geom_point(aes(size = Frequency, color = Frequency)) +
  labs(x = 'South End Pickup', y = 'Andrew T Stop Dropoff') +
  scale_x_continuous("South End Pickup",
                     labels = as.character(joint_df$South_End_freq_shortcut),
                     breaks = joint_df$South_End_freq_shortcut) +
  scale_y_continuous("Andrew T Stop Dropoff",
                     labels = as.character(joint_df$Andrew_T_freq),
                     breaks = joint_df$Andrew_T_freq)
```

This heatmap has 4 points, thus showing the lack of data in this dataset given the constraints. It is clear that a lot more bikes are picked up at South End than bikes dropped off at the Andrew T Stop.

```
# Coefficient
South_End_Coeff <- bikes_df %>%
  select(start.station.name, month, date, hour) %>%
  dplyr::filter(start.station.name == 'South End Library - Tremont St at W Newton St'
                & hour >= 10 & hour <= 11) %>%
  group_by(month, date) %>%
  summarise(count = n())
```

```
## `summarise()` regrouping output by 'month' (override with `.groups` argument)
```

```
South_End_Coeff
```

```
## # A tibble: 30 x 3
## # Groups:   month [5]
##    month  date count
##    <int> <int> <int>
##  1     1     6     1
##  2     1     8     1
##  3     1    10     1
##  4     1    14     1
##  5     1    17     1
##  6     1    19     1
##  7     6     2     1
##  8     6     5     4
##  9     6    12     1
## 10     6    15     1
## # ... with 20 more rows
```

```
mod_South_End <- head(South_End_Coeff, 10)
mod_South_End
```

```
## # A tibble: 10 x 3
## # Groups:   month [2]
##    month  date count
##    <int> <int> <int>
##  1     1     6     1
##  2     1     8     1
##  3     1    10     1
##  4     1    14     1
##  5     1    17     1
##  6     1    19     1
##  7     6     2     1
##  8     6     5     4
##  9     6    12     1
## 10     6    15     1
```

```
# Group
Andrew_T_Coeff <- bikes_df %>%
  select(end.station.name, month, date, hour) %>%
```

```
  dplyr::filter(end.station.name == 'Andrew T Stop - Dorchester Ave at Dexter St'
                & hour >= 10 & hour <= 11) %>%
  group_by(month, date) %>%
  summarise(count = n())
```

```
## `summarise()` regrouping output by 'month' (override with `.groups` argument)
```

```
Andrew_T_Coeff
```

```
## # A tibble: 2 x 3
## # Groups:   month [1]
##   month  date count
##   <int> <int> <int>
## 1     6    23     1
## 2     6    29     2
```

```
# Manipulation
South_End_mat <- c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0)
Andrew_T_mat <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1)
Corr_Coeff <- cor(South_End_mat, Andrew_T_mat)
Corr_Coeff
```

```
## [1] -1
```

This was a bit flawed, as I ran into issues calculating the coefficient because of different dimension sizes. So, I only used the first 10 rows of the South_End_Coeff table to calculate the coefficient. I could not see the other 20 values, so I had to get rid of them in my calculations. I then created respective matrices for each coefficient to calculate the overall coefficient. I used time to assign a value of 0 or 1, and ordered them chronologically. The South_End matrix had values of 1 from [0, 10] of a matrix of length 12 because the 2 dates for the Andrew_T matrix were the lowest chronologically.

I got a correlation coefficient of -1, which shows an inverse relationship. This was shown well by the frequency scatterplot, where it showed how few bikes were dropped off at Andrew T Stop versus how many bikes were picked up at South End – Tremont St at W Newton St.

Task 3 (Bonus Points)

Search for datasets from reputed online data repositories (such as UCI Machine Learning Repository) and provide details about those for which similar statistics (PMF, CDF, Joint Probability) can be calculated.

Conclusion

I wasn't able to deduce much from this lab because the dataset was very minimal, and given the constraints assigned to us in the assignment, we could not generate a set of probability values that could tell us anything of real value. Regardless, I still learned a lot on how to calculate PMFs, CDFs, Joint Probabilities, and correlation coefficients, all while generating data visuals using ggplot2, which I know I need to work on. In the future, I would like to work with larger data sets so we can really see the impact of probability and statistics in real life.