Assignment #5

MSAN 593

***DUE**: Tuesday, August 16th , 23:45*

Be sure to upload an`*.Rmd` file named `asn5.Rmd` as well as `asn5.pdf` with all required graphs to Canvas by the due date and time. A short write-up about any data cleaning is required. Late submissions will receive a grade of zero. Your `*.Rmd` files will be run on local machines by graders. If your file does not run, you will automatically lose 30% of the grade. If you resubmit your corrected homework by the last class of the module, it will be graded out of the remaining 70%. Failure to resubmit will result in a grade of zero. Assume that the data file(s) being read into your `*.Rmd` files are in the current local directory, e.g., `read.csv('myFile.csv')` will work. Do **not** hard code a specific directory structure.

Question 1

You will be working on anonymized data from hotel from `hotelData.csv`. The data contains the following variables:

- `room`: room name
- `date`: date
- `area`: area of `room` in square feet
- `event_min`: the number of minutes an event scheduled in `room` on `date`
- `max_min`: total number of available minutes in which to schedule an event in `room` on `date`
- `year`: year
- `month`: month
- `day`: day

For all of the following graphing exercises, you are expected to use `ggplot()` and to label the axes titles, tick marks and graph titles exactly as the sample graphs (including units and size).

Task 1

Clean and validate the data and briefly explain what steps you took to clean and validate the data.

Task 2

Create a line graph plotting the room utilization rate in percent (y) against day of year from 1 to 365 (x), horizontally faceted by year (six facets). This will generate 21 graphs, one for each room.

Task 3

Create a line graph plotting the room utilization rate in percent (y) against day of year (x), horizontally faceted by weekday (Monday to Sunday, seven facets). This will generate 21 graphs, one for each room.

Task 4

Create a line graph plotting the **mean** room utilization rate in percent (y), against day of year from 1 to 365 (x), faceted horizontally by either weekday or weekend (two facets). Furthermore, include a semi-opaque area indicating the min and max utilization for any given day. This will generate 21 graphs, one for each room.

Task 5

Create a line graph plotting the **mean** room utilization rate in percent (y), against day of year from 1 to 365 (x). Furthermore, include a semi-opaque area indicating the min and max utilization for any given day. This will generate 21 graphs, one for each room.

Task 6

For each day of the week (Monday - Sunday), run `kmeans()` on the mean utilizations of each room for each day of the week, with $k \in \mathbb{K} = \{2, \ldots, 10\}$. For each $k \in \mathbb{K}$, create a horizontal Cleveland dot plot plotting room name (y) on mean utilization in percent (x), grouping clusters by color. This will generate Cleveland dot plots per day of the week, and 63 in total. For each weekday, you should also create a summary line and dot plot plotting total within SS (y) on number of clusters (x). This will generate an additional seven plots.

Task 7

Using the `corrplot` package, create a correlation matrix for mean utilizations for each room for each year. Include the option where insignificant correlations are x-ed out (see example), resulting in six correlation matrices.

Task 8

Using the `corrplot` package, create a correlation matrix for mean utilizations for each room for each day of the week. Include the option where insignificant correlations are x-ed out (see example), resulting in seven correlation matrices.

Task 9

Using the `corrplot` package, create a correlation matrix for mean room utilizations. Include the option where insignificant correlations are x-ed out. This will result in a single correlation matrix.

n.b.

- No legends are necessary on any of the graphs, but you should have descriptive titles, axis titles and axis tick marks. Font sizes for the aforementioned should also be sufficiently large to be comfortably legible.
- Examples of all graphs are on Slack and should be used as a template.
- All graphs are expected to look identical to sample graphs provided.
- Use `ggsave()` to save all graphics in `pdf`. Any grainy graphics will be heavily penalized.
- Once all graphs have been created, merge them all into a single `pdf`.
- You **will** be graded on code clarity, efficiency, and etiquette
  - Using five lines of code when one will do will result in loss of marks
  - Using a `for()` loop when an `apply()` will work will result in loss of marks
  - You should employ piping in your code (from the `maggritr` package) where appropriate. Although your entire code need not be piped, it should be used at least a couple of times to demonstrate your knowledge of piping.

Question 2

2.1 Using a single line graph, generate the probability density functions of the beta distribution where $\{\alpha, \beta\} \in \{\{0.5, 0.5\}, \{5, 1\}, \{1, 3\}, \{2, 2\}, \{2, 5\}\}$. The pdf of each each $\{\alpha, \beta\}$ tuple should be differentiated using color, and a legend should be included.

2.2 Using a single line graph, generate the cumulative distribution functions of the beta distribution where $\{\alpha, \beta\} \in \{\{0.5, 0.5\}, \{5, 1\}, \{1, 3\}, \{2, 2\}, \{2, 5\}\}$. The pdf of each each $\{\alpha, \beta\}$ tuple should be differentiated using color, and a legend should be included.

2.3 Combining output from 2.1 and 2.2, create a single graphical output with all pdfs and cdfs. They will all be line graphs and will be faceted using two columns and five rows. The title of each facet column will be `pdf` and `cdf`. The titles of the row facets, located on the right hand side of the graph, will have the $\{\alpha, \beta\}$ tuple values, i.e., $\{\alpha, \beta\} = \{x, y\}$, where x and y are the tuple values. The titles **must** have the correct greek symbols α and β, and the tuple values must be generated dynamically based on their values (not hard coded). The output will be a single graphical object with 10 sub-graphs in two columns and five rows.

Question 3

The file `redfinData.txt` contains unstructured data about the housing market. Each row contains a monthly observation for either the national market or a specific market, e.g., San Francisco. The data contains the following specific information:

- `Average Sale To List YoY`
- `Average Sale To List MoM`
- `Average Sale To List`
- `Days on Market YoY`
- `Days on Market MoM`
- `Days on Market`
- `Inventory YoY`
- `Inventory MoM`
- `Inventory`
- `New Listings YoY`
- `New Listings MoM`
- `New Listings`
- `Homes Sold YoY`
- `Homes Sold MoM`
- `Homes Sold`
- `Median Sale Price YoY`
- `Median Sale Price MoM`
- `Median Sale Price`

Create a data frame where each of the above variables is a column, along with `year` and `region`. The data in your data frame should be the mean for a given region/year combination. E.g., there should exist a single row for `region == San Francisco` and `year == 2014`, with values for each of the above variables equal to the mean of the all the monthly values for that year. Another row should exist for `region == San Francisco` and `year == 2015` and so on, for each unique region/year pairing. The final deliverable is the data frame itself.