Problem Set 3

Statistics 506, Fall 2017

Due Tuesday November 14, 2017 via Canvas

Course Homepage (https://jbhender.github.io/Stats506/)

Instructions

- Use R markdown to create PDF files (one per question) containing your answers to each of the questions posed below. Embed your R code either by setting “echo = TRUE” or by using fenced code blocks “````”.
- Your code should be clearly written and it should be possible to assess it by reading it. Use appropriate variable names and comments. Your style will be graded using the style rubric (./StyleRubric.html).
- All work for this problem set should be done using R. Do not do any data input/output or pre/post processing of results using tools other than R.
- Some of these exercises will require you to use commands or techniques that were not covered in class or in the course notes. You can use the web as needed to identify appropriate approaches. Questions are welcome at all times, but please make an attempt to locate relevant information yourself first.

Problems

1. Repeat question 1 from problem set 2 (also question 2 from problem set 1), using `data.table` . You only need to display tables and do not need to display a graph. For each part, summarize the key steps in plain English (i.e. without R code) using 3-5 bullet points.

---

2. Use the `NYCflights14` data from here (https://raw.githubusercontent.com/wiki/arunsrinivasan/flights/NYCflights14/flights14.csv) to answer the following questions. Each part can and should be answered using a single (perhaps chained) `data.table` expression prior to any requested plotting.
   a. Compute the average departure delay per flight for all carriers by month. Then, graph your results as a spaghetti plot showing the time trends for each carrier.
   b. Compute the 90^{th} percentile of arrival delays by carrier, origin, and destination. For each origin airport, produce a heat map to display the data.
   c. For each origin airport, compute the average departure delay for each of the following time windows:
      - 0:00 - 11:59
      - 12:00 - 17:59
      - 18:00 - 23:59.
   d. Within each flight, center and scale the air time by the mean. Next, bin the departure delays into fligths that left early or on time, flights delayed by less than 15 minutes, and flights that left more than 15 minutes late. For each bin of departure delays, compute a 95% confidence

interval for the mean relative air time.

3. In this question you will use web-scraping to supplement the `NYCflights14` data with distance between all airports. To get you started, I have written a script (./AirportCodesWebScrape.R) using the R package `rvest` (https://blog.rstudio.com/2014/11/24/rvest-easy-web-scraping-with-r/) to scrape the distances between all destination airports in the `NYCflights14` data from this site.

    a. Using my script as a model, write a script to scrape the distances between the three origin airports and also from each origin airport to each destination airport. I suggest you test your script using just the three origin airports, then include the destination airports once you have it working. Your script should not find distances between destination airports as these are provided here (./AirportCodeDists.RData).

    b. Combine the distances from part a, with the distances provided between all destination airports. Reshape these data into a 112 by 112 pairwise distance matrix.

    c. Use multidimensional scaling to produce a two-dimensional map for these 112 airports.

4. In this question, you will combine the airport distance data from question 3 with the `NYCflights14` data to build visualizations. Where possible, use `data.tables` for aggregation and data manipulation.

    a. Determine the number of flights per week from each origin airport to each destination airport among all carriers.

    b. Display the data from part “a” as network graph using the coordinate system from question 3, part “c”. Your display should show airports as nodes and have (directed) edges from each origin airport to each destination airport. The thickness of the edges should be proportional to the number of weekly flights found in part “a”. You may wish to use one of the following approaches to construct your plot:

        - `ggplot` using `geom_segment()`
        - base R plotting using `arrows()`
        - DiagrammeR (http://rich-iannone.github.io/DiagrammeR/docs.html)
        - igraph (http://igraph.org/redirect.html).

    c. Repeat part “a” separately for each carrier. Then compute pairwise distances between carriers based on the frequency of flights between airports. Use MDS to create a 2-dimensional map of the carriers. Briefly discuss your findings.

    d. Compute the average weekly number of flights for each carrier between all origin and destination airports. Repeat the visualization from part “c” after normalizing the frequency data to control for the average weekly number of flights by each carrier. Briefly discuss your findings and contrast with what you found in part “c”.