**Bay Area Bike Rental Operation Research**

The dataset provided is from the second year of Bay Area Bike Share's operation and has been downloaded from Kaggle (<https://www.kaggle.com/datasets/benhamner/sf-bay-area-bike-share>). The dataset has been transformed to suit the purposes of this exercise.

The data is in this dataset represents data collected from 1/1/14 to 12/31/14. All files in the dataset are comma delimited and contain header rows.

\*\*\* Please note that this dataset is not the same as the one shared in the course page.

The files contained in this dataset include:

**station.csv** - Contains data that represents a station where users can pick up or return bikes.

**trips.csv** - Data about individual bike trips

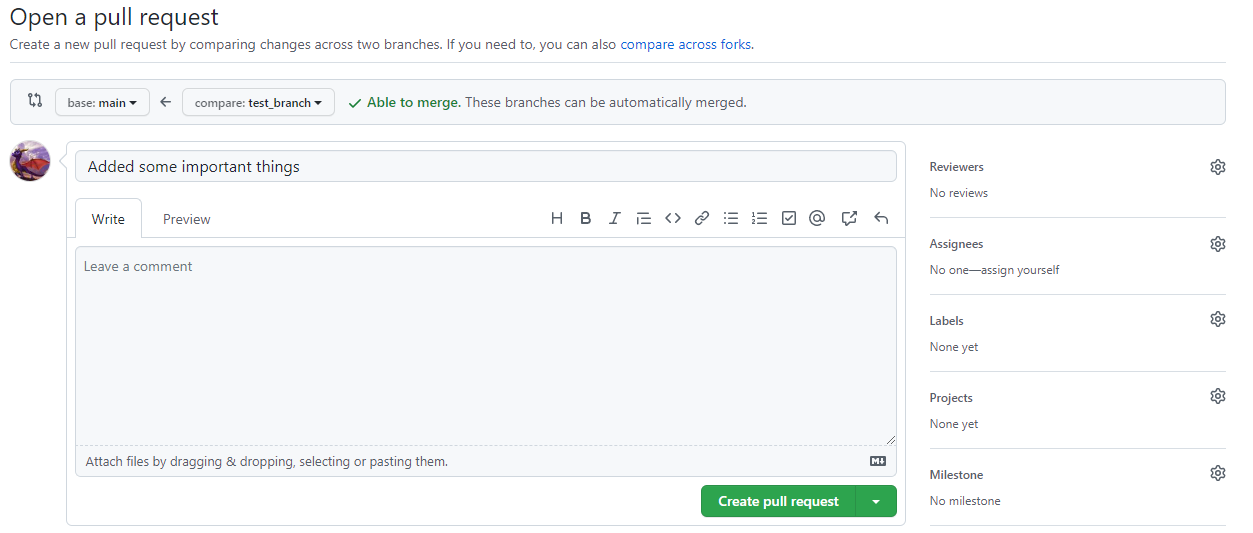
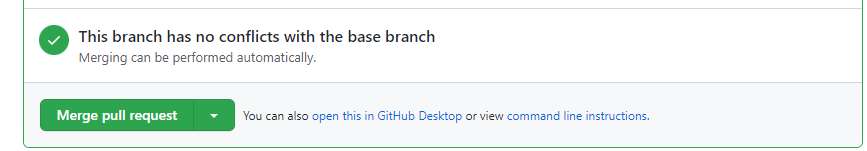
**weather.csv** - Data about the weather on a specific day for certain zip codes/cities.

You and your partner are working as data analysts with a team of data scientists at Bay Area Bike Operations. The data science team is interested in creating a predictive model that will predict the number of bikes leaving each station and the number of bikes being returned to each station in the next three days. This insight will help the maintenance team to better plan their bike/dock maintenance operations.

You meet with your team, discuss the project and decide on the following:

* The data analysts should run an exploratory data analysis (EDA) on the trip and weather data.
  + Exploratory Data Analysis or EDA is a statistical approach or technique for analyzing data sets in order to summarize their important and main characteristics generally by using some visual aids. The EDA approach can be used to gather knowledge about the main characteristics or features of the data, the variables and their relationships, finding out the important variables that can be used in our problem.
  + You can follow the steps at <https://blog.datascienceheroes.com/exploratory-data-analysis-in-r-intro/> and do a quick and dirty EDA using the funModeling package.
* Any trip with duration less than 2 minutes is likely a 'cancelled trip'. Find out the number of such trips, record the information for your report and then remove them from the dataset.
* Identify the outliers in the dataset (you have to decide which table/column), record them for your final report and then remove from the dataset.
* The data science team needs you to establish the highest volume hours on weekdays, so that they can build 'rush hours' into their model (lubridate package is your BFF here). You can choose what approach to take, but you have to find the hours of weekdays where the trip volume is highest. (e.g. you may try histograms)
* Determine the 10 most frequent starting stations and ending stations during the rush hours you established.
* Determine the 10 most frequent starting stations and ending stations during the weekends.
* Calculate the average utilization of bikes for each month (total time used/total time in month).
* The team assumes that weather conditions probably have an impact on the bike rental patterns, but they are not sure whether they should use temperature, weather events, visibility or other weather measurements available. Help them decide by creating a new dataset combining trip data with the weather data. (Note that the weather data is available for *each city and date*. Join your datasets accordingly). Create a correlation matrix for the new dataset using the cor() function from the corrplot package. Flag the highest correlations for the data science team.
* You team lead expects a Data Analysis Report with all your findings for the next meeting in two weeks.

**Tasks:**

* Prefer to use ‘tidyverse’ packages when appropriate for the purposes of this exercise.
* Start by creating a repo in the DHT22-UoT organization on GitHub.
* Take the time to meet briefly and distribute the tasks. It is always a good idea to use the README.md file (generated automatically when creating a repo if you select the option) to note the task assignments.
  + When making the task assignments, try to divide tasks so that they can be either individual functions or individual scripts. It is neither efficient nor advisable for two people to work on the same script file simultaneously unless they are practicing pair programming.
  + Decide on who will be responsible to merge pull requests. It is always easier to keep track of the work when only one person in the team merges the pull requests.
* Each group member should then clone the repo (with the task assignments) to their local computers.
* Each member should create a new branch (with a meaningful name) for each task.
  + Start coding on your task branch.
  + Repeat the pull-commit-push cycle as often as necessary. There should be no direct uploads of complete scripts, ever!
  + Once you are done with your task and pushed all your outstanding commits, go to GitHub. You will see a message box like this:
  + Click on the 'Compare & pull request' button.
  + On the 'Open a pull request' screen, put in a short description of what you have done; add any concerns/comments for your reviewer. Assign your partner as reviewer by clicking on the little gear icon next to Reviewers on the top right side. Then click Create pull request.
  + Your partner will receive an email to review your code. After the code review and possibly some additional commits answering the reviewer concerns, you are ready to merge. The party responsible can then go to the pull request page and click the 'Merge pull request' button.
  + While your reviewer is reviewing your first branch, you can create a new branch and start working on your new task.
  + At the end of the project, make sure that all the relevant/useful branches have been merged to main.
* You team lead expects a Data Analysis Report with all your findings for the next meeting in two weeks. You will create a small research report (preferably a Word file). Your report should include a brief description and summary of the dataset (using tables/plots from EDA), describe any pre-processing you applied to the dataset (e.g. number of excluded records, exclusion reason), the requirements from your team and your findings.
* Remember that you can save any plots in RStudio using the Export button. Save and embed select plots in your report as appropriate.
* Push your Report into your repo (preferably in an appropriately named directory).
* Submit your repo URL in Quercus.
* Sit back and enjoy a job well done!