<https://github.com/SThornewillvE/Udacity-Project---US-Bikeshare-Data-Analysis> solution one

<https://vickysuworld.blogspot.in/2017/10/2016-us-bike-share-activity-snapshot.html> solution 2

<https://github.com/sunsided/DAND-BayAreaBikeShare/blob/master/Bay_Area_Bike_Share_Analysis.ipynb> solution 3

<https://www.python.org/dev/peps/pep-0257/>

<https://www.datacamp.com/courses/introduction-to-data-visualization-with-python>

<https://www.hackerrank.com/domains/python?filters%5Bsubdomains%5D%5B%5D=py-introduction> try hackerank

*csv*, *pprint*, *datetime*, and *time library neeeded use numpy and pandas too*

<https://www.motivateco.com/use-our-data/>

Pprint <http://www.informit.com/articles/article.aspx?p=1719315&seqNum=9>

## 

|  |  |  |
| --- | --- | --- |
| **1** | **Import pandas** |  |
| 2 | Input functions city, month , week | I take input as in city / month or week or none |
|  | I have  City  Time period month day none  If time period == month  Ask which month  Then filter the data frame here and serve it  Ask which day  Filter the df and serve  If none  Give the full df  If time period ( month  Make three dfs and use it | **What i am going to do is make two dataframes one with month and another with days of the week.**  **I ll split date**  **And then i ll add month or days to it in another coulumn**  **I need three df s here df1 df2 and df 3** |
| 3 | I need to make two dataframes niow |  |
| **4** | **From city filter month or week** |  |
| **5** | **Feed the required df for rest of the things** |  |
| **6** | **Popular month** | **Mode of th edataframe** |
| **7** | **Popular day** |  |
| **8** | **Popular hour** |  |
| **9** | **Trip duration** |  |
| **x** | **Popular station** |  |
| **xi** | **Popular trip** |  |
| **12** | **user** |  |
| **13** | **gender** | **Washington doesnt have gender and birth year. And gender and birth year have NAN values** |
| **14** | **Birth year** |  |
| **15** | **Display data** |  |
| **16** | **statistics** | **Around 12 solutuions** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 

## **The Questions**

You will write code to answer the following questions about the bike share data:

* What is the most popular month for start time?
* What is the most popular day of week (Monday, Tuesday, etc.) for start time? *Hint: datetime.weekday()(documentation* [*here*](https://docs.python.org/3/library/datetime.html#datetime.date.weekday)*) may be helpful!*
* What is the most popular hour of day for start time?
* What is the total trip duration and average trip duration?
* What is the most popular start station and most popular end station?
* What is the most popular trip?
* What are the counts of each user type?
* What are the counts of gender?
* What are the earliest (i.e. oldest person), most recent (i.e. youngest person), and most popular birth years?
* Get stats
* **Add your own questions**
* **Display raw data if asked.**
* **Member Engagement**
* **Make sure that only trips with a max duration of 75 mins are plotted with bins 5 minutes wide. Plot the trips and see which ones are where**
* **Which city has the highest number of trips? Which city has the highest proportion of trips made by subscribers? Which city has the highest proportion of trips made by short-term customers?**
* ***"""*  
   *Takes as input a dictionary containing info about a single trip (datum) and*  
   *its origin city (city) and returns the trip duration in units of minutes.*  
    
   *Remember that Washington is in terms of milliseconds while Chicago and NYC*  
   *are in terms of seconds.*   
    
   *HINT: The csv module reads in all of the data as strings, including numeric*  
   *values. You will need a function to convert the strings into an appropriate*  
   *numeric type when making your transformations.*  
   *see https://docs.python.org/3/library/functions.html*  
   *"""***
* ***"""*  
   *This function reads in a file with trip data and reports the number of*  
   *trips made by subscribers, customers, and total overall.*  
   *"""***
* **We’ve found effective onboarding to be a key factor in driving usage and renewals: members who ride at least 5 times in their first month are 40% more likely to renew their memberships. To this end, we’ve employed techniques to engage and activate new members through a welcome package and onboarding email series. We continue to drive engagement throughout a member’s lifecycle with monthly newsletters, seasonal emails, and user-specific messaging.**
* [**https://www.motivateco.com/use-our-data/**](https://www.motivateco.com/use-our-data/) **more questions here**

|  |  |
| --- | --- |
| RITERIA | MEETS SPECIFICATIONS |
| Functionality of code | All code cells can be run without error. |
| Choice of data types and structures | Appropriate data types (e.g. strings, floats) and data structures (e.g. lists, dictionaries) are chosen to carry out the required analysis tasks. |
| Use of loops and conditional statements | Loops and conditional statements are used to process the data correctly. |
| Use of packages | Packages are used to carry out advanced tasks. |
| Use of functions | Functions are used to reduce repetitive code. |
| Use of good coding practices | Docstrings, comments, and variable names enable readability of the code. |

Script and Questions

|  |  |
| --- | --- |
| CRITERIA | MEETS SPECIFICATIONS |
| Solicit and handle raw user input | Raw input is solicited and handled correctly to guide the interactive question-answering experience; no errors are thrown when unexpected input is entered. |
| Use descriptive statistics to answer questions about the data | Descriptive statistics are correctly computed and used to answer the questions posed about the data. Raw data is displayed upon request by the user. |

### **Suggestions to Make Your Project Stand Out!**

* Change the structure of bikeshare.py to make the code more efficient or in better style.
* Ask and answer additional questions about the data beyond the questions already provided.
* Make the interactive experience wow-worthy! Add images, make it into a web app, etc. Make it your own!

|  |  |
| --- | --- |
| Print hello world |  |
| Import files and see data |  |
| Answer the nine questions |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Questions that need answers:

|  |  |
| --- | --- |
| 1 | 1. Would you like to see data for Chicago, New York, or Washington? |
| 2 | 1. Would you like to filter the data by month, day, or not at all? |
| 3 | 1. Which day of the week? Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, or Sunday? |
| 4 | 1. Which month? January, February, March, April, May, or June? |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

My end product should be

A sample end product would look something like this:

* **Prompt**: Hello! Let's explore some US bike share data! Would you like to see data for Chicago, New York, or Washington?
* **User input**: Chicago
* **Prompt**: Would you like to filter the data by month, day, or not at all? Type "none" for no time filter.
* **User input**: month
* **Prompt**: Which month? January, February, March, April, May, or June?
* **User input**: March
* *Statistics for Chicago in March are printed out*
* **Prompt**: Would you like to restart? Type 'yes' or 'no.'
* **User input**: no

<https://www.youtube.com/watch?v=tJkoTihIOUE> if needed.

I'd completed the original Bikeshare project but decided to re-submit the new Bikeshare in an attempt to gain feedback on the requirements so I could be more helpful. I wanted to offer a few lessons learned that might be beneficial prior to your own submission:

(1) Verify results of \*all\* the filter options. My code was correct for filter='none' but my count was off on one statistic (`popular\_station()`) if the filter was 'month', due to "if's" that should have been "elif's" that caused an error in my counts.

(2) The review feedback received didn't indicate the filter option associated with the incorrect output, only that the `popular\_station()` statistic was wrong, so you may not get much feedback if there is an error.

(3) It appears the reviewer tested with a full data file, rather than a test file.

(4) My approach was to load once, and loop once and count for each statistic. I didn't do any subsetting or sorting, so while those approaches may also be acceptable they were not required, looping through the data in this manner was apparently fine.

(5) My approach was to use numerical 'day of the week" (1, 2, 3...) as input, and not 'calendar day" (no checks for number of days in a month, etc). I used string day of the week for output (i.e., Saturday, Sunday, etc). Again, other interpretations may be acceptable too but 'day of the week' was acceptable in my case.

(6) I did not receive any feedback on the 'day" option, but I chose to include 2 approaches to the 'day' filter in my code by adding another filter option, one irrespective of the selected month ('day') and one for the day \_within\_ the selected month (which I called 'both', meaning filter by month \*and\* day) and I noted that in the comments of my submission. Unfortunately, I don't know which was the expected interpretation.

(7) The reviewer noted that all my stats computed correctly and that my "time trial for the stats beat our expected timing, amazing work!" I'm a bit skeptical of this feedback because it appears the reference timings provided were far too quick to be my code (unless they executed my code on a supercomputer), but it does appear they at least "looked" at the timings

***My result should be***

***Most popular month: June***

***Most popular day for start time: Tuesday***

***Most popular hour: 17***

***Total Duration:1458069892.0, Count:1551505, Avg Duration:939.7777590146342, Filter:none***

***Start Station: Streeter Dr & Grand Ave, Count:36686***

***End Station:Streeter Dr & Grand Ave, Count:39537, Filter:none***

***Popular trip: ('Lake Shore Dr & Monroe St', 'Streeter Dr & Grand Ave'), Count:4647, Filter:none***

***Subscribers:1234339, Customers:317162, Filter:none***

<https://udacitydatascience.slack.com/messages/C71DTS9CJ/details/> slack

###@@

## **Submitting the Project**

When you're ready, click on the "Submit Project" button to go to the project submission page. You can submit your files as a .zip archive or you can link to a GitHub repository containing your project files. If you go with GitHub, note that your submission will be a snapshot of the linked repository at the time of submission. It is recommended that you keep each of your projects in a separate repository to avoid any potential confusion: if a reviewer gets multiple folders representing multiple projects, there might be confusion regarding what project is to be evaluated.

It can take us up to a week to grade the project, but in most cases it is much faster. You will get an email once your submission has been reviewed. If you are having any problems submitting your project or wish to check on the status of your submission, please email us at dataanalyst-project@udacity.com. In the meantime, you should feel free to proceed with your learning journey by continuing on to the next module in the program.