Github Project: Git Commands Documentation Template

Programming for Data Science Nanodegree Program

You will use this template to copy and paste the git commands you used to complete all tasks on your local and remote git repository for this project. This file will serve as your submission for the GitHub project.

Instructions:

1. Make a copy of this Git Commands Documentation template on your Google Drive.
2. Complete the four sections in this document with the appropriate git commands.
3. Download this document as a PDF file.
4. Submit this on the Project Submission page within the Udacity Classroom.

1. Set Up Your Repository

The following are the steps you will take to create your git repository, add your python code, and post your files on GitHub.

Step 1. Create a GitHub profile (if you don't already have one).
Step 2. Fork a repository from Udacity's GitHub Project repository and provide a link to your forked GitHub repository here:

GitHub Repository Link
https://github.com/ahmednassar34/Programming-for-Data-Science-Nanodegree-Program

Step 3. Complete the tasks outlined in the table below and copy and paste your git commands into the "Git Commands" column. The first git command is partially filled out for you.

	Tasks	Git Commands
A.	Clone the GitHub repository to your local repository.	git clone https://github.com/ahmednassar34/Programming-for-Data-Science-Nanodegree-Program
B.	Move your bikeshare.py and data files into your local repository.	**No git command needed (you can use cp or a GUI)**
C.	Create a .gitignore file containing the name of your data file.	**No git command needed (you can use touch or a GUI)**
D.	List the file names associated with the data files you added to your .gitignore	**No git command needed (add the file names into your .gitignore file)**
E.	Check the status of your files to make sure your files are not being tracked	Git status
F.	Stage your changes.	Git add .

G.	Commit your changes with a descriptive message.	git commit -m "BikeShare Project"
H.	Push your commit to your remote repository.	git push origin master

2. Improve Documentation

Now you will be working in your local repository, on the BikeShare python file and the README.md file. You should repeat steps C through E three times to make at least three commits as you work on your documentation improvements.

	Tasks	**Git Commands**
A.	Create a branch named *documentation* on your local repository.	git branch documentation
B.	Switch to the *documentation* branch.	git checkout documentation
C.	Update your README.md file.	**No git command needed (edit the text in your README.md file)**
D.	Stage your changes.	git add README.md
E.	Commit your work with a descriptive message.	git commit -m "Changing README.md documentation"
F.	Push your commit to your remote repository branch.	git push origin documentation
G.	Switch back to the master branch.	git checkout master

3. Additional Changes to Documentation

In a real world situation, you or other members of your team would likely be making other changes to documentation on the documentation branch. To simulate this follow the tasks below.

	Tasks	**Git Commands**
A.	Switch to the *documentation* branch.	git checkout documentation
B.	Make at least 2 additional changes to the documentation - this might be additional changes to the README or changes to the document strings and line comments of the bikeshare file.	git diff diff --git a/README.md b/README.md index a3334b4..0389bff 100644 --- a/README.md +++ b/README.md @@ -1,2 +1,69 @@ # Programming-for-Data-Science-Nanodegree-Program Programming for Data Science Nanodegree Program Project + +#Bike Share Data + +Over the past decade, bicycle-sharing systems have been growing in number and popularity in cities across the world. Bicycle-sharing systems allow users to rent bicycles on a very short-term basis for a price. This allows people to borrow a bike from point A and return it at point B, though they can also return it to the same location if they'd like to just go for a ride. Regardless, each bike can serve several users per day. + +Thanks to the rise in information

		technologies, it is easy for a user of the system to access a dock within the system to unlock or return bicycles. These technologies also provide a wealth of data that can be used to explore how these bike-sharing systems are used. + +In this project, you will use data provided by Motivate, a bike share system provider for m: diff --git a/README.md b/README.md index a3334b4..0389bff 100644 --- a/README.md +++ b/README.md @@ -1,2 +1,69 @@ # Programming-for-Data-Science-Nanodegree-Program Programming for Data Science Nanodegree Program Project + +#Bike Share Data +

		+Over the past decade, bicycle-sharing systems have been growing in number and popularity in cities across the world. Bicycle-sharing systems allow users to rent bicycles on a very short-term basis for a price. This allows people to borrow a bike from point A and return it at point B, though they can also return it to the same location if they'd like to just go for a ride. Regardless, each bike can serve several users per day. + +Thanks to the rise in information technologies, it is easy for a user of the system to access a dock within the system to unlock or return bicycles. These technologies also provide a wealth of data that can be used to explore how these bike-sharing systems are used. + +In this project, you will use data provided by Motivate, a bike share system provider for :

		diff --git a/README.md b/README.md index a3334b4..0389bff 100644 --- a/README.md +++ b/README.md @@ -1,2 +1,69 @@ # Programming-for-Data-Science-Nanodegree-Program Programming for Data Science Nanodegree Program Project + +#Bike Share Data + +Over the past decade, bicycle-sharing systems have been growing in number and popularity in cities across the world. Bicycle-sharing systems allow users to rent bicycles on a very short-term basis for a price. This allows people to borrow a bike from point A and return it at point B, though they can also return it to the same location if they'd like to just go for a ride. Regardless, each bike can serve several users per day. + +Thanks to the rise in information technologies, it is easy for a user of the system to access a dock within the system to unlock or return bicycles. These technologies also provide a wealth of data that can be used to explore how these bike-sharing systems are used. + +In this project, you will use data provided by Motivate, a bike share system provider for m:...skipping... diff --git a/README.md b/README.md index a3334b4..0389bff 100644 --- a/README.md

		+++ b/README.md @@ -1,2 +1,69 @@ # Programming-for-Data-Science-Nanodegree-Program Programming for Data Science Nanodegree Program Project + +#Bike Share Data + +Over the past decade, bicycle-sharing systems have been growing in number and popularity in cities across the world. Bicycle-sharing systems allow users to rent bicycles on a very short-term basis for a price. This allows people to borrow a bike from point A and return it at point B, though they can also return it to the same location if they'd like to just go for a ride. Regardless, each bike can serve several users per day. + +Thanks to the rise in information technologies, it is easy for a user of the system to access a dock within the system to unlock or return bicycles. These technologies also provide a wealth of data that can be used to explore how these bike-sharing systems are used. + +In this project, you will use data provided by Motivate, a bike share system provider for many major cities in the United States, to uncover bike share usage patterns. You will compare the system usage between three large cities: Chicago, New York City, and Washington, DC. + +#The Datasets + +Randomly selected data for the first six months of 2017 are

		provided for all three cities. All three of the data files contain the same core six (6) columns: + + Start Time (e.g., 2017-01-01 00:07:57) + End Time (e.g., 2017-01-01 00:20:53) + Trip Duration (in seconds - e.g., 776) + Start Station (e.g., Broadway & Barry Ave) + End Station (e.g., Sedgwick St & North Ave) + User Type (Subscriber or Customer) + +The Chicago and New York City files also have the following two columns: + + Gender + Birth Year + +Data for the first 10 rides in the new_york_city.csv file + +The original files are much larger and messier, and you don't need to download them, but they can be accessed here if you'd like to see them (Chicago, New York City, Washington). These files had more columns and they differed in format in many cases. Some data wrangling has been performed to condense these files to the above core six columns to make your analysis and the evaluation of your Python skills more straightforward. In the Data Wrangling course that comes later in the Data Analyst Nanodegree program, students learn how to wrangle the dirtiest, messiest datasets, so don't worry, you won't miss out on learning this important skill!

		+ +Statistics Computed + +You will learn about bike share use in Chicago, New York City, and Washington by computing a variety of descriptive statistics. In this project, you'll write code to provide the following information: :

		diff --git a/README.md b/README.md index a3334b4..0389bff 100644 --- a/README.md +++ b/README.md @@ -1,2 +1,69 @@ # Programming-for-Data-Science-Nanodegree-Program Programming for Data Science Nanodegree Program Project + +#Bike Share Data + +Over the past decade, bicycle-sharing systems have been growing in number and popularity in cities across the world. Bicycle-sharing systems allow users to rent bicycles on a very short-term basis for a price. This allows people to borrow a bike from point A and return it at point B, though they can also return it to the same location if they'd like to just go for a ride. Regardless, each bike can serve several users per day. + +Thanks to the rise in information technologies, it is easy for a user of the system to access a dock within the system to unlock or return bicycles. These technologies also provide a wealth of data that can be used to explore how these bike-sharing systems are used. + +In this project, you will use data

		provided by Motivate, a bike share system provider for many major cities in the United States, to uncover bike share usage patterns. You will compare the system usage between three large cities: Chicago, New York City, and Washington, DC. + +#The Datasets + +Randomly selected data for the first six months of 2017 are provided for all three cities. All three of the data files contain the same core six (6) columns: + + Start Time (e.g., 2017-01-01 00:07:57) + End Time (e.g., 2017-01-01 00:20:53) + Trip Duration (in seconds - e.g., 776) + Start Station (e.g., Broadway & Barry Ave) + End Station (e.g., Sedgwick St & North Ave) + User Type (Subscriber or Customer) + +The Chicago and New York City files also have the following two columns: + + Gender + Birth Year + +Data for the first 10 rides in the new_york_city.csv file + +The original files are much larger and messier, and you don't need to download them, but they can be accessed here if you'd like to see them (Chicago, New York City, Washington). These files had more columns and they differed in format in many cases. Some data

		wrangling has been performed to condense these files to the above core six columns to make your analysis and the evaluation of your Python skills more straightforward. In the Data Wrangling course that comes later in the Data Analyst Nanodegree program, students learn how to wrangle the dirtiest, messiest datasets, so don't worry, you won't miss out on learning this important skill! + +Statistics Computed :
C.	After each change, stage and commit your changes. When you commit your work, you should use a descriptive message of the changes made. Your changes should be small and aligned with your commit message.	git add . git commit -m “Improve function choice docstrings” git add . git commit -m “Changes made to README.md documentation”
D.	Push your changes to the remote repository branch.	git push origin documentation
E.	Switch back to the *master* branch.	git checkout master
F.	Check the local repository log to see how *all the branches* have changed.	git log --oneline --graph --all
G.	Go to Github. Notice that you now have two branches available for your project, and when you change branches the README changes.	**No git command needed**

4. Refactor Code

Now you will be working in your local repository, on the code in your BikeShare python file to make improvements to its efficiency and readability. You should repeat steps C through E three times to make at least three commits as you refactor.

	Tasks	**Git Commands**
A.	Create a branch named *refactoring* on your local repository.	git checkout -b refactoring
B.	Switch to the *refactoring* branch.	git checkout -b refactoring
C.	Similar to the process you used in making the documentation changes, make 2 or more changes in refactoring your code.	**No git command needed (edit the code in your python file)**
D.	*For each change,* stage and commit your work with a descriptive message of the changes made.	git commit -m "Add print statement warning regarding Washington's lack of user data" git commit -m “Add exception to trip duration stats”
E.	Push your commits to your remote repository branch.	git push origin refactoring
F.	Switch back to the *master* branch.	git checkout master
G.	Check the local repository log to see how *all the branches* have changed.	git log --graph --all --oneline
H.	Go to GitHub. Notice that you now have 3 branches. Notice how the files change as you move through the branches.	**No git command needed**

5. Merge Branches

	Tasks	**Git Commands**
A.	Switch to the *master* branch.	git checkout master
B.	Pull the changes you and your coworkers might have made in the passing days (in this case, you won't have any updates, but pulling changes is often the first thing you do each day).	git pull origin
C.	Since your changes are all ready to go, merge all the branches into the master. Address any merge conflicts. If you split up your work among your branches correctly, you should have no merge conflicts.	git merge refactoring git merge documentation
D.	You should see a message that shows the changes to the files, insertions, and deletions.	**No git command needed**
E.	Push the repository to your remote repository.	Git push origin
F.	Go to GitHub. Notice that your master branch has all of the changes.	**No git command needed**

Submission:

This concludes the project.

- Please review this document to make sure you entered all the required response fields in all four sections.
- Download this document as a PDF file.
- Submit the PDF file on the Project Submission page within the Udacity Classroom.