Assignment #1

MACS 30000, Dr. Evans
Due Wednesday, Oct. 10 at 11:30am

For this assignment, you will work in groups of 3 or 4 students. There are eight groups of 4 and two groups of 3. You will be assigned your membership in a group based on the alphabetical student listing from the course registration. I have posted these group assignments in a GitHub issue on our course repository here (https://github.com/UC-MACSS/persp-analysis_A18/issues/4). There are 10 groups.

1. **Basic Git and GitHub steps for the class (1 point).** Complete the following steps.

   (a) Make sure Git software is installed on your computer.

   (b) Sign up for your own personal GitHub account at GitHub.com. Once you have a Git Handle, make sure to record it in the survey instrument that the TA's created for the class (link here).

   (c) Go to the course repository on GitHub (https://github.com/UC-MACSS/persp-analysis_A18) and click the "Watch" button in the upper-right region of the page. Doing this will sign you up to receive notifications whenever the repository receives an update.

   (d) "Fork" the course repository on GitHub (https://github.com/UC-MACSS/persp-analysis_A18) by clicking the "Fork" button in the upper-right region of the page. Doing this will create a fork or copy of this repository on your own personal GitHub account.

   (e) Read the Git and GitHub tutorial in our course repository.

2. **Git and GitHub group project (9 points).** Each of you is assigned to a group according to the assignments in the GitHub issue on our course repository here (https://github.com/UC-MACSS/persp-analysis_A18/issues/4). This issue also assigns each of you a role, which is either *Core Developer*, *Contributor 1*, *Contributor 2*, or *Contributor 3*. For groups with only three members, *Contributor 1* will have to take on the additional tasks of *Contributor 3*. Each person in the group will receive the same grade. So feel free to help each other on each others assigned tasks. This portion of the assignment will be graded by the TA's checking your respective repositories to make sure that each of the following steps was completed correctly.

   (a) *Core Developer* create a new repository named `DescrStats` on her personal GitHub account. Select the options of "Public repository" and "Initialize this repository with a README". DO NOT select the option "Add .gitignore" or "Add a license". NOTE: DO NOT clone this repository inside of another Git repository.

(b) *Core Developer* adds *Contributor 3* as a "contributor" to the repository. This gives *Contributor 3* the same "merge" privileges as *Core Developer*.

(c) *Contributor 1*, *Contributor 2*, and *Contributor 3* fork this repository.

(d) *Core Developer* creates a branch named "python" and submits a pull request to the "master" branch in which she has added the Python file `DescrStats.py`, the data file `datafile.txt`, and the file `.gitignore` to the repository. The files `DescrStats.py` and `datafile.txt` are in the Assignment 1 directory (here). The file `.gitignore` is in the main directory of the course repository (here).

(e) *Contributor 3* merges this pull request with a comment, "LGTM" (this means "looks good to me".

(f) *Contributor 1* creates a branch on her fork named "read" in which she updates the `README.md` to have the following text.

```
# Documentation for `DescrStats.py` and `DescrStats()` function

This repository contains the Python module `DescrStats.py`, which defines the
    function `DescrStats()`. This function can be used by importing the module
    and using the function on a datafile that is a comma-delimited, single
    variable text file (`.txt`) in which the data are organized as a single
    column with each row representing an observation.

```python
import DescStats as ds

ds.DescrStats(`datafile.txt')
```

We have placed a sample data file (`datafile.txt`) in this repository. You can use
    this data file to test your
```

Contributor 1 submits this change to the main repository "master" branch by submitting a pull request.

(g) *Core Developer* merges with comment "Thanks for updating the `README.md`. It looks good".

(h) *Contributor 2* finds an error in the `DescrStats.py` code. The `data.variance()` function is incorrect. *Contributor 2* creates a branch named "varfix" and makes the following change line 17 of the module.

```
data_var = data.var()
```

Contributor 2 then submits this as a pull request.

(i) *Contributor 3* merges with the comment, "Nice work catching this."

(j) *Contributor 3* creates the following issue, entitled "Extra DescrStats() features" with the issue label "enhancement", that uses check boxes (tasks lists) for a list of features to add to `DescrStats.py`.

```
I think we should add more features to the `DescrStats()` function in
    `DescrStats.py`.
- [] Return descriptive statistics as a tuple of scalars
- [] Add the `minimum`, `maximum`, and `medium` statistics
- [] Include a Boolean that tells the function to plot a histogram of the data
```

(k) *Contributor 3* creates a branch named "returnvals" and makes the following change to the 'DescrStats.py' module.

```
def DescrStat(datafile):
    data = np.loadtxt(datafile)
    data_mean = data.mean()
    data_std = data.std()
    data_var = data.var()

    print('Mean of the data is:', data_mean)
    print('Standard deviation of the data is:', data_std)
    print('Variance of the data is:', data_var)

    return data_mean, data_std, data_var
```

Contributor 3 submits this change as a PR to the main repository.

(l) *Core Developer* merges this PR with the message "Thanks for that increased functionality".

(m) *Contributor 3* checks the first box in the issue entitled, "Extra DescrStats() features".

(n) *Core Developer* realizes that the module 'DescrStats.py' does not have good internal documentation. She creates a branch named "docstr" and makes the following changes.

```
def DescrStat(datafile):
    '''
    ---------------------------------------------------------------------
    This function prints and returns descriptive statistics on a comma-
    delimited text file of a single variable
    ---------------------------------------------------------------------
    INPUTS:
    datafile = string, path of the data file to be used in the function

    OTHER FUNCTIONS AND FILES CALLED BY THIS FUNCTION: None

    OBJECTS CREATED WITHIN FUNCTION:
    data = (N,) vector, data from datafile
    data_mean = scalar, mean of the data
    data_std = scalar >= 0, standard deviation of the data
    data_var = scalar >= 0, variance of the data

    FILES CREATED BY THIS FUNCTION: None

    RETURNS: data_mean, data_std, data_var
    ---------------------------------------------------------------------
    '''
    data = np.loadtxt(datafile)
    data_mean = data.mean()
    data_std = data.std()
    data_var = data.var()

    print('Mean of the data is:', data_mean)
    print('Standard deviation of the data is:', data_std)
    print('Variance of the data is:', data_var)

    return data_mean, data_std, data_var
```

Core Developer submits these changes as a pull request to the "master" branch.

(o) *Contributor 3* merges this PR with the comment, "Thanks. That really makes this function more usable and readable."