**Assignment 4 [Pandas]**

**Saranya P**

<<Data Analytics >>

**Pandas, Pandas, Pandas**

**Table of Contents**

[**Background** 2](#_Toc18884643)

[**Pre-Work** 2](#_Toc18884644)

[**Assignment Instruction** 4](#_Toc18884645)

[**Heroes of Pymoli** 4](#_Toc18884646)

[**Player Count** 5](#_Toc18884647)

[**Purchasing Analysis (Total)** 5](#_Toc18884648)

[**Gender Demographics** 5](#_Toc18884649)

[**Purchasing Analysis (Gender)** 6](#_Toc18884650)

[**Age Demographics** 7](#_Toc18884651)

[**Purchasing Analysis (Age)** 7](#_Toc18884652)

[**Top Spenders** 8](#_Toc18884653)

[**Most Popular Items** 9](#_Toc18884654)

[**Most Profitable Items** 10](#_Toc18884655)

[**Observations:** 11](#_Toc18884656)

[**Player Count** 11](#_Toc18884657)

[**Purchasing Analysis (Total)** 11](#_Toc18884658)

[**Gender Demographics** 11](#_Toc18884659)

[**Purchasing Analysis (Gender)** 11](#_Toc18884660)

[**Age Demographics** 11](#_Toc18884661)

[**Purchasing Analysis (Age)** 12](#_Toc18884662)

[**Top Spenders** 12](#_Toc18884663)

[**Most Popular Items** 12](#_Toc18884664)

[**Most Profitable Items** 12](#_Toc18884665)

[**Hints and Considerations** 12](#_Toc18884666)

[**GitHub** 13](#_Toc18884667)

**# Unit 4 | Assignment - Pandas, Pandas, Pandas**

# **Background**

The data dive continues!

Now, it's time to take what you've learned about Python Pandas and apply it to new situations. For this assignment, you'll need to complete \*\*one of two\*\* (not both) Data Challenges. Once again, which challenge you take on is your choice. Just be sure to give it your all -- as the skills you hone will become powerful tools in your data analytics tool belt. Both of these challenges encompasses a real-world situation where your newfound Python scripting skills can come in handy. These challenges are far from easy so expect some hard work ahead!

# **Pre-Work**

1. Create a new repository for this project called `pandas-challenge`. \*\*Do not add this homework to an existing repository\*\*.

**<Sara Comment> :**

***A repository Pandas-Challenge have been created.***

<https://github.com/SaranyaPandiaraj/Pandas-Challenge>

2. Clone the new repository to your computer.

**<Sara Comment> :**

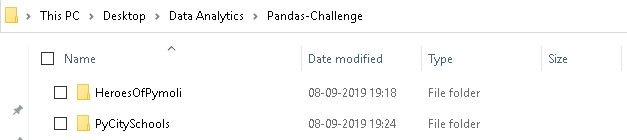
***The Pandas-Challenge have been cloned to my computer using the below command***

***<<git clone*** <https://github.com/SaranyaPandiaraj/Pandas-Challenge.git>***>>***

3. Inside your local git repository, create a directory for the Pandas Challenge you choose. Use folder names corresponding to the challenges: \*\*HeroesOfPymoli\*\* or \*\*PyCitySchools\*\*.

**<Sara Comment> :**

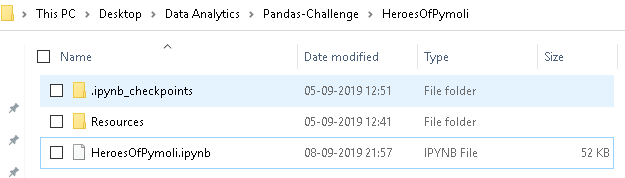
***Folder HeroesOfPymoli have been created inside my local git repository.***



4. Add your Jupyter notebook to this folder. This will be the main script to run for analysis.

**<Sara Comment> :**

***Jupyter Notebook have been created inside the HeroesOfPymoli Folder.***



5. Push the above changes to GitHub or GitLab.

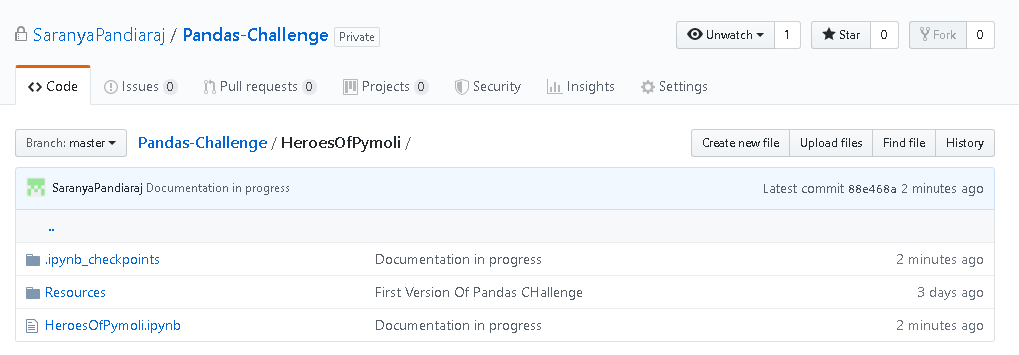
**<Sara Comment> :**

***The above changes have been pushed to GitLab using the below commands.***

***git add.***

***git commit -m “Initial Version”***

***git push***



# **Assignment Instruction**

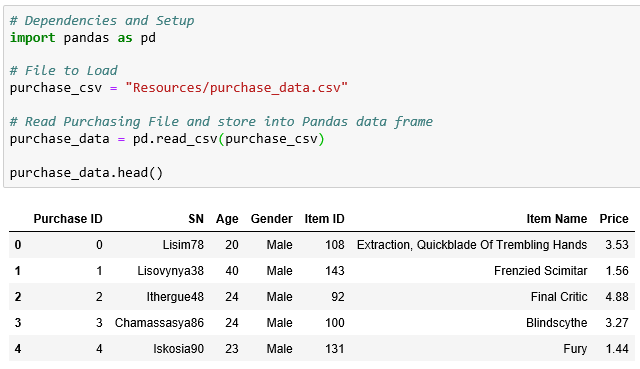
## **Heroes of Pymoli**



Congratulations! After a lot of hard work in the data munging mines, you've landed a job as Lead Analyst for an independent gaming company. You've been assigned the task of analyzing the data for their most recent fantasy game Heroes of Pymoli.Like many others in its genre, the game is free-to-play, but players are encouraged to purchase optional items that enhance their playing experience. As a first task, the company would like you to generate a report that breaks down the game's purchasing data into meaningful insights.

**<Sara Comment> :**

***Importing Dependencies, Reading the Purchase Data file and storing into Pandas data frame (purchase\_data).***



Your final report should include each of the following:

### **Player Count**

***.***

\* Total Number of Players

**<Sara Comment> :**

***The Total Unique Number of Players is calculated using nunique() function and is stored in Player\_Count\_df Data Frame.***



### **Purchasing Analysis (Total)**

\* Number of Unique Items

\* Average Purchase Price

\* Total Number of Purchases

\* Total Revenue

**<Sara Comment> :**

* ***The Number of Unique Item is calculated using nunique() function on Item ID column.***
* ***Average Purchase Price is calculated using mean() function on Price Column.***
* ***Total Number of Purchases is calculated using count() function on Purchase ID Column.***
* ***Total Revenue is calculated using sum() function on Price Column***

***All the above column calculations are stored in Purchasing\_Analysis\_Total\_df data frame.***

***The Data Frame has been formatted using <dataframe name>.style.format({“Column\_name” : “Required format”* }) syntax.**



### **Gender Demographics**

\* Percentage and Count of Male Players

\* Percentage and Count of Female Players

\* Percentage and Count of Other / Non-Disclosed

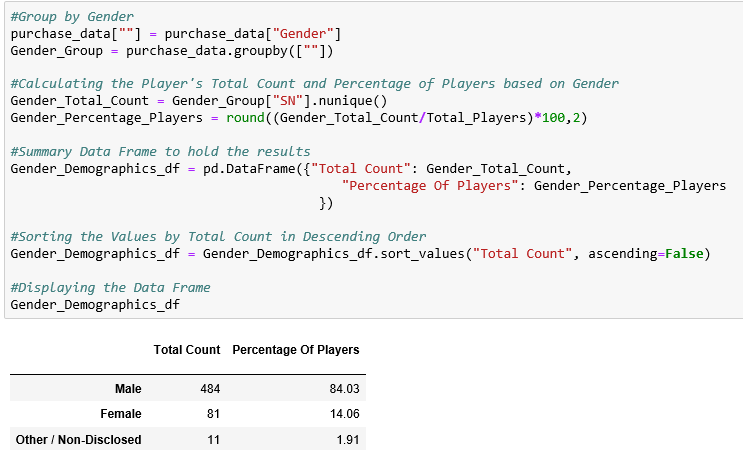
**<Sara Comment> :**

* ***The purchase\_data data frame has been grouped based on Gender and stored in Gender\_Group.***
* ***The count of Male, Female & Other/Non-Disclosed is calculated using nunique() function on SN Column based on Gender\_Group result.***
* ***The Percentage of Male, Female & Other/Non-Disclosed is calculated by (Total Count by Gender Wise/ Total Player) \* 100 and has been rounded up to 2 decimal point using round() function.***

***All the above column calculations are stored in Gender\_demographics\_df data frame.***

***The Data frame has been sorted using <dataframe name>.sort\_values(“Column\_name”, ascending=False) based on the Total Count by Gender wise.***

***The Data Frame has been formatted using <dataframe name>.style.format({“Column\_name” : “Required format”* }) syntax.**



### **Purchasing Analysis (Gender)**

\* The below each broken by gender

\* Purchase Count

\* Average Purchase Price

\* Total Purchase Value

\* Average Purchase Total per Person by Gender

**<Sara Comment> :**

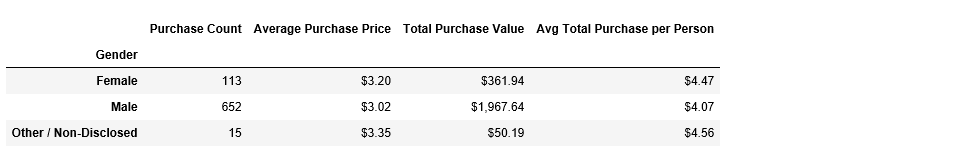
* ***The purchase\_data data frame has been grouped based on Gender and stored in Gender\_Group.***
* ***The Purchase Count is calculated using nuinque() function on Purchase ID column based on Gender\_Group result.***
* ***The Average Purchase Price is calculated using mean() function on Price column based on Gender\_Group result.***
* ***The Total Purchase Value is calculated using sum() function on Price column based on Gender\_Group result.***
* ***The Average Purchase Total per Person by Gender Value is calculated using by Total Purchase Value/Total UniquePlayers based on Gender\_Group result.***

***All the above column calculations are stored in Purchasing\_Analysis\_Gender\_df data frame.***

***The Data frame index has been sorted using sort\_index() function.***

***The Data Frame has been formatted using <dataframe name>.style.format({“Column\_name” : “Required format”* }) syntax.**





### **Age Demographics**

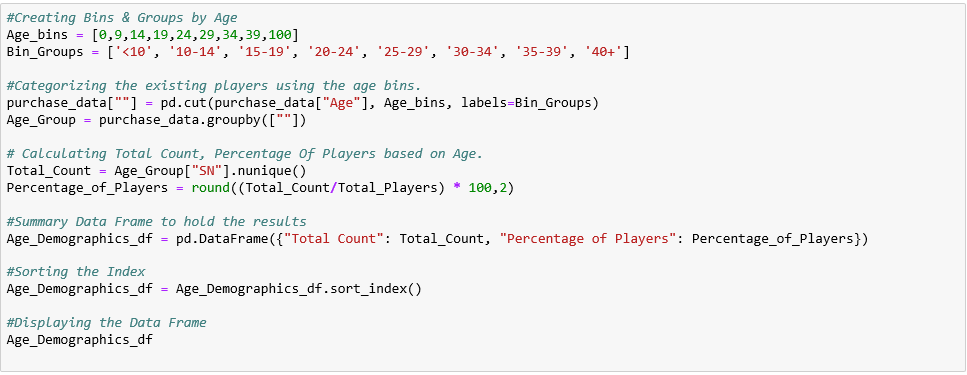
* Establish bins for ages
* Categorize the existing players using the age bins. Hint: use pd.cut()
* Calculate the numbers and percentages by age group
* Create a summary data frame to hold the results
* Optional: round the percentage column to two decimal points
* Display Age Demographics Table

**<Sara Comment> :**

* ***A Bin has been created to store the age values.***
* ***Age Group Range has been created for the Bin.***
* ***The existing players has been categorized using the Age bins by applying pd.cut() function on purchase\_data data frame. Then the data frame has been grouped based on Age Column and stored in Age\_Group.***
* ***The number of players is calculated using nunique() function on SN Column based on Age\_Group result.***
* ***The Percentage of Players is calculated by (Total Count by Age Wise/ Total Player) \* 100 and has been rounded up to 2 decimal point using round() function.***

***All the above column calculations are stored in Age\_demographics\_df data frame.***

***The Data frame index has been sorted using sort\_index() function.***





### **Purchasing Analysis (Age)**

\* The below each broken into bins of 4 years (i.e. &lt;10, 10-14, 15-19, etc.)

\* Purchase Count

\* Average Purchase Price

\* Total Purchase Value

\* Average Purchase Total per Person by Age Group

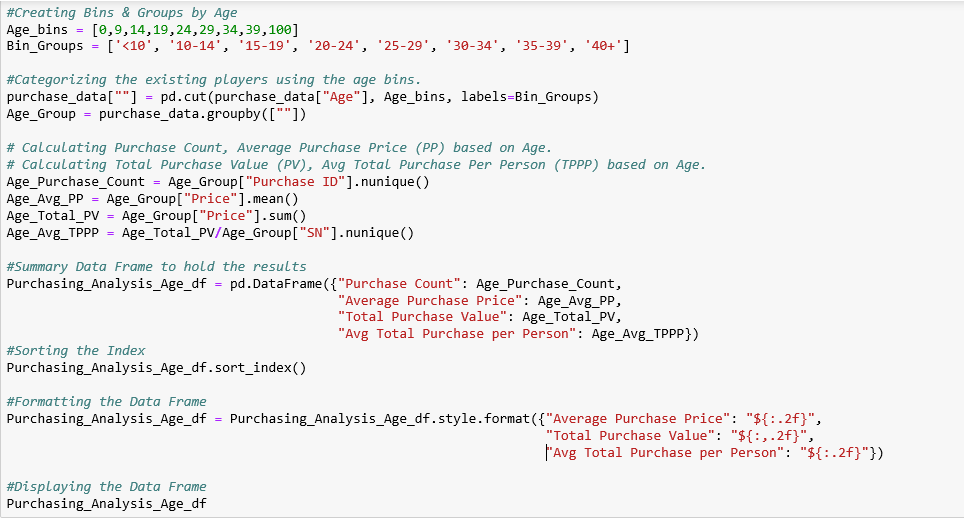
**<Sara Comment> :**

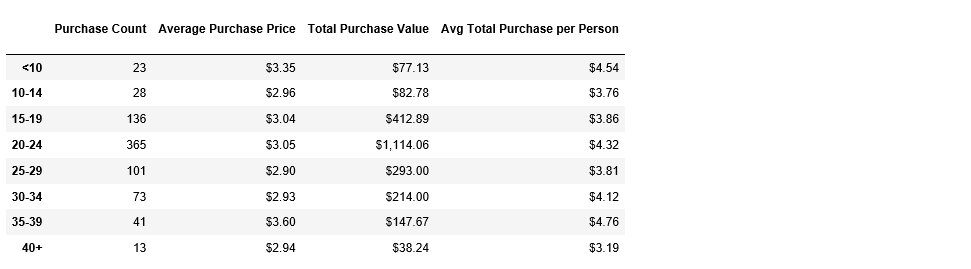
* ***A Bin has been created to store the age values.***
* ***Age Group Range has been created for the Bin.***
* ***The existing players has been categorized using the Age bins by applying pd.cut() function on purchase\_data data frame. Then the data frame has been grouped based on Age Column and stored in Age\_Group.***
* ***The Purchase Count is calculated using nuinque() function on Purchase ID column based on Age\_Group result.***
* ***The Average Purchase Price is calculated using mean() function on Price column based on Age\_Group result.***
* ***The Total Purchase Value is calculated using sum() function on Price column based on Age\_Group result.***
* ***The Average Purchase Total per Person by Age Group is calculated using by Total Purchase Value/Total UniquePlayers based on Age\_Group result..***

***All the above column calculations are stored in Purchasing\_Analysis\_Age\_df data frame.***

***The Data frame index has been sorted using sort\_index() function.***

***The Data Frame has been formatted using <dataframe name>.style.format({“Column\_name” : “Required format”* }) syntax.**





### **Top Spenders**

\* Identify the the top 5 spenders in the game by total purchase value, then list (in a table):

\* SN

\* Purchase Count

\* Average Purchase Price

\* Total Purchase Value

**<Sara Comment> :**

* ***The purchase\_data data frame has been grouped based on SN and stored in SN\_Group.***
* ***The Purchase Count is calculated using nuinque() function on Purchase ID column based on SN\_Group result.***
* ***The Average Purchase Price is calculated using mean() function on Price column based on SN\_Group result.***
* ***The Total Purchase Value is calculated using sum() function on Price column based on SN\_Group result.***

***All the above column calculations are stored in Top\_Spenders\_df data frame.***

***The Data frame has been sorted using <dataframe name>.sort\_values(“Column\_name”, ascending=False) based on the Total Purchase Value.***

***To show the preview of the first 5 Data, <dataframe name>.head() function is used.***

***The Data Frame has been formatted using <dataframe name>.style.format({“Column\_name” : “Required format”* }) syntax.**





### **Most Popular Items**

\* Identify the 5 most popular items by purchase count, then list (in a table):

\* Item ID

\* Item Name

\* Purchase Count

\* Item Price

\* Total Purchase Value

**<Sara Comment> :**

* ***The purchase\_data data frame has been grouped based on Item ID & Item Name and stored in Item\_Group.***
* ***The Purchase Count is calculated using nuinque() function on Purchase ID column based on Item\_Group result.***
* ***The Item Price is calculated using mean() function on Price column based on Item\_Group result.***
* ***The Total Purchase Value is calculated using sum() function on Price column based on Item\_Group result.***

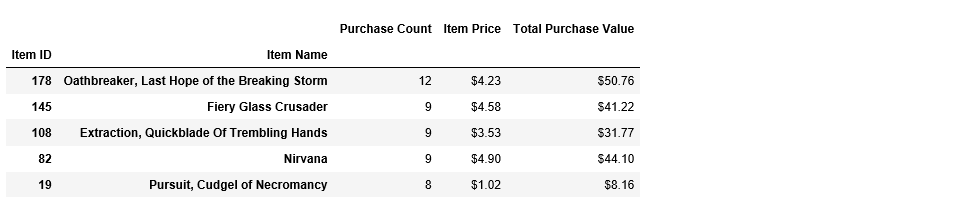
***All the above column calculations are stored in Most\_Popular\_Item\_df data frame.***

***The Data frame has been sorted using <dataframe name>.sort\_values(“Column\_name”, ascending=False) based on the Purchase Count.***

***To show the preview of the first 5 Data, <dataframe name>.head() function is used.***

***The Data Frame has been formatted using <dataframe name>.style.format({“Column\_name” : “Required format”* }) syntax.**





### **Most Profitable Items**

\* Identify the 5 most profitable items by total purchase value, then list (in a table):

\* Item ID

\* Item Name

\* Purchase Count

\* Item Price

\* Total Purchase Value

**<Sara Comment> :**

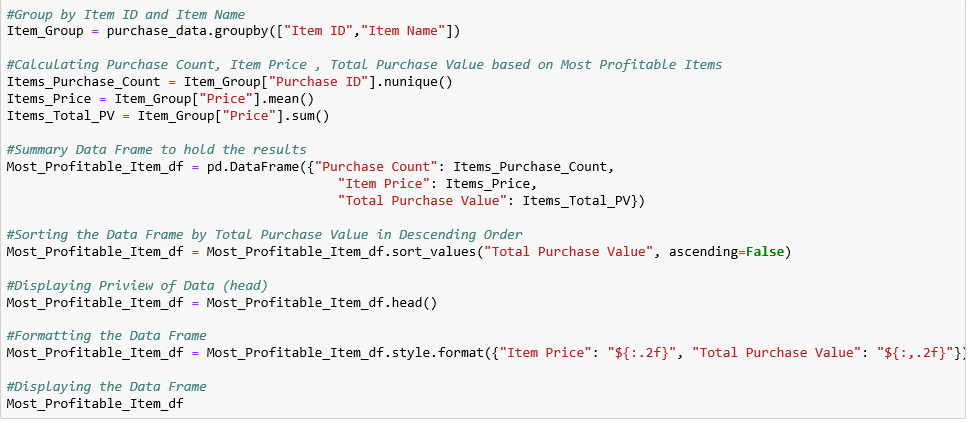
* ***The purchase\_data data frame has been grouped based on Item ID & Item Name and stored in Item\_Group.***
* ***The Purchase Count is calculated using nuinque() function on Purchase ID column based on Item\_Group result.***
* ***The Item Price is calculated using mean() function on Price column based on Item\_Group result.***
* ***The Total Purchase Value is calculated using sum() function on Price column based on Item\_Group result.***

***All the above column calculations are stored in Most\_Profitable\_Item\_df data frame.***

***The Data frame has been sorted using <dataframe name>.sort\_values(“Column\_name”, ascending=False) based on the Total Purchase Value.***

***To show the preview of the first 5 Data, <dataframe name>.head() function is used.***

***The Data Frame has been formatted using <dataframe name>.style.format({“Column\_name” : “Required format”* }) syntax.**





# **Observations:**

## **Player Count**

* As per the Purchase Data CSV (purchase\_data.csv), the total purchase ID count is **780** but the number of unique players is **576** only. From this we can come to conclusion that there are purchasers who buy the Heroes of Pymoli Game repeatedly.

## **Purchasing Analysis (Total)**

* From the Purchasing Analysis Data, we can conclude that the Total Revenue of the **183** Unique items is **$2379.77** of having **780** Total Number of Purchases with an average price of **$3.05**.

## **Gender Demographics**

* From the Gender Demographics, we can conclude that **the Male Count** and the Percentage of Male Players (84%) is higher than the Female Player and Other/Non-Disclosed Genders.

## **Purchasing Analysis (Gender)**

* From the Purchasing Analysis based on Gender, we can conclude three points,

1. Total Purchase Value is higher in males because the Purchase Count in Male Gender is higher. (It’s obvious since the Percentage of Male Players is 84%)
2. The Average Purchase Price in the Other/Non-Disclosed Gender Category is higher than the Other Genders having a value of **$3.35**.
3. The Average Total Purchase Per Person in the Female Gender Category is higher than the Other Genders having a value of **$4.47.**

## **Age Demographics**

* Based on the Age Demographics Data, we can conclude that,

1. The Age-Group between **20-24** is having the highest number of players (258) followed by Age-Group 15-19 having 107 players.
2. The Age-Group in the **Group 40+** is having the least of 12 players followed by Ager-Group <10 having 17 players.

## **Purchasing Analysis (Age)**

* From the Purchasing Analysis based on Age Group, we can conclude three points,

1. Total Purchase Value is higher in the **Age Group 20-24 ($1114.06)** followed by the Age Group (15-19) having $412.89 value since the Purchase count in these Age Groups are high and the Age Group 40+ having the least of $38.24.
2. The Average Purchase Price is higher in the **Age Group 35-39** of having $3.60 value and least in the Age Group 25-29 of having $2.90
3. The Average Total Purchase Per Person is higher in the **Age Group 35-39** of having $4.76 value and least in the Age Group 40+ of having $3.19

## **Top Spenders**

* Based on the Top Spenders data, we can conclude that **Lisosia93** tops of purchasing 5 items with a highest Total Purchase Value of **$18.96** followed by Idastidru52 having the purchase value of $15.45 The Average Purchase price of Lisosia93 is $3.79.

## **Most Popular Items**

* From the Most Popular Data Frame, we can conclude that "**Oathbreaker, Last Hope of the Breaking Storm**" item is the most popular item of having 12 Purchase Count of $50.76 Total Purchase Value followed by "Fiery Glass Crusader" having 9 purchase count and $41.22 Total Purchase Value.

## **Most Profitable Items**

* From the Most Profitable Data Frame, we can conclude that "**Oathbreaker, Last Hope of the Breaking Storm**" item is the most Profitable item of having $50.76 Total Purchase Value followed by "Nirvana" having $44.10 Total Purchase Value.

# **Hints and Considerations**

\* These are challenging activities for a number of reasons. For one, these activities will require you to analyze thousands of records. Hacking through the data to look for obvious trends in Excel is just not a feasible option. The size of the data may seem daunting, but pandas will allow you to efficiently parse through it.

\* Second, these activities will also challenge you by requiring you to learn on your feet. Don't fool yourself into thinking: "I need to study pandas more closely before diving in." Get the basic gist of the library and then \_immediately\_ get to work. When facing a daunting task, it's easy to think: "I'm just not ready to tackle it yet." But that's the surest way to never succeed. Learning to program requires one to constantly tinker, experiment, and learn on the fly. You are doing exactly the \_right\_ thing, if you find yourself constantly practicing Google-Fu and diving into documentation. There is just no way (or reason) to try and memorize it all. Online references are available for you to use when you need them. So use them!

\* Take each of these tasks one at a time. Begin your work, answering the basic questions: "How do I import the data?" "How do I convert the data into a DataFrame?" "How do I build the first table?" Don't get intimidated by the number of asks. Many of them are repetitive in nature with just a few tweaks. Be persistent and creative!

\* Expect these exercises to take time! Don't get discouraged if you find yourself spending hours initially with little progress. Force yourself to deal with the discomfort of not knowing and forge ahead. Consider these hours an investment in your future!

\* As always, feel encouraged to work in groups and get help from your TAs and Instructor. Just remember, true success comes from mastery and \_not\_ a completed homework assignment. So challenge yourself to truly succeed!

As final considerations:

\* You must use the Pandas Library and the Jupyter Notebook.

\* You must submit a link to your Jupyter Notebook with the viewable Data Frames.

\* You must include a written description of three observable trends based on the data.

\* See [Example Solution](HeroesOfPymoli/HeroesOfPymoli\_starter.ipynb) for a reference on expected format.

# **GitHub**

**<Sara Comment> :**

The link to the Pandas Assignment is in the below link.

< <https://github.com/SaranyaPandiaraj/Pandas-Challenge> >