# Dependencies

import pandas as pd

import numpy as np

file\_to\_load = "Resources/purchase\_data.csv"

# Read Purchasing File and store into Pandas data frame

purchase\_data = pd.read\_csv(file\_to\_load)

Player Count[¶](https://render.githubusercontent.com/view/ipynb?commit=ece990c7351216059fa675fc9d9aaadb150d8117&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f52757467657273436f64696e67426f6f7463616d702f5255544a455232303138303944415441332f656365393930633733353132313630353966613637356663396439616161646231353064383131372f30342d50616e6461732f486f6d65776f726b2f536f6c7574696f6e732f4865726f65734f6650796d6f6c692f4865726f65734f6650796d6f6c695f536f6c7574696f6e2e6970796e623f746f6b656e3d416672474a534568376543585a5a3663714d2d694b65796943334d6e3774564b6b733563545161457741253344253344&nwo=RutgersCodingBootcamp%2FRUTJER201809DATA3&path=04-Pandas%2FHomework%2FSolutions%2FHeroesOfPymoli%2FHeroesOfPymoli_Solution.ipynb&repository_id=133828859&repository_type=Repository#Player-Count)

Display the total number of players

In [2]:

# Calculate the Number of Unique Players

player\_demographics = purchase\_data.loc[:, ["Gender", "SN", "Age"]]

player\_demographics = player\_demographics.drop\_duplicates()

num\_players = player\_demographics.count()[0]# Display the total number of players

pd.DataFrame({"Total Players": [num\_players]})

Out[2]:

|  | Total Players |
| --- | --- |
| 0 | 576 |

Purchasing Analysis (Total)[¶](https://render.githubusercontent.com/view/ipynb?commit=ece990c7351216059fa675fc9d9aaadb150d8117&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f52757467657273436f64696e67426f6f7463616d702f5255544a455232303138303944415441332f656365393930633733353132313630353966613637356663396439616161646231353064383131372f30342d50616e6461732f486f6d65776f726b2f536f6c7574696f6e732f4865726f65734f6650796d6f6c692f4865726f65734f6650796d6f6c695f536f6c7574696f6e2e6970796e623f746f6b656e3d416672474a534568376543585a5a3663714d2d694b65796943334d6e3774564b6b733563545161457741253344253344&nwo=RutgersCodingBootcamp%2FRUTJER201809DATA3&path=04-Pandas%2FHomework%2FSolutions%2FHeroesOfPymoli%2FHeroesOfPymoli_Solution.ipynb&repository_id=133828859&repository_type=Repository#Purchasing-Analysis-(Total))

Run basic calculations to obtain number of unique items, average price, etc.

Create a summary data frame to hold the results

Optional: give the displayed data cleaner formatting

Display the summary data frame

In [3]:

# Run basic calculations

average\_item\_price = purchase\_data["Price"].mean()

total\_purchase\_value = purchase\_data["Price"].sum()

purchase\_count = purchase\_data["Price"].count()

item\_count = len(purchase\_data["Item ID"].unique())

# Create a DataFrame

summary\_table = pd.DataFrame({"Number of Unique Items": item\_count,

"Total Revenue": [total\_purchase\_value],

"Number of Purchases": [purchase\_count],

"Average Price": [average\_item\_price]})

# Minor Data Munging

summary\_table = summary\_table.round(2)

summary\_table ["Average Price"] = summary\_table["Average Price"].map("${:,.2f}".format)

summary\_table ["Number of Purchases"] = summary\_table["Number of Purchases"].map("{:,}".format)

summary\_table ["Total Revenue"] = summary\_table["Total Revenue"].map("${:,.2f}".format)

summary\_table = summary\_table.loc[:,["Number of Unique Items", "Average Price", "Number of Purchases", "Total Revenue"]]

# Display the summary\_table

summary\_table

Out[3]:

|  | Number of Unique Items | Average Price | Number of Purchases | Total Revenue |
| --- | --- | --- | --- | --- |
| 0 | 183 | $3.05</td> <td>780</td> <td>$2,379.77 |  |  |

Gender Demographics[¶](https://render.githubusercontent.com/view/ipynb?commit=ece990c7351216059fa675fc9d9aaadb150d8117&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f52757467657273436f64696e67426f6f7463616d702f5255544a455232303138303944415441332f656365393930633733353132313630353966613637356663396439616161646231353064383131372f30342d50616e6461732f486f6d65776f726b2f536f6c7574696f6e732f4865726f65734f6650796d6f6c692f4865726f65734f6650796d6f6c695f536f6c7574696f6e2e6970796e623f746f6b656e3d416672474a534568376543585a5a3663714d2d694b65796943334d6e3774564b6b733563545161457741253344253344&nwo=RutgersCodingBootcamp%2FRUTJER201809DATA3&path=04-Pandas%2FHomework%2FSolutions%2FHeroesOfPymoli%2FHeroesOfPymoli_Solution.ipynb&repository_id=133828859&repository_type=Repository#Gender-Demographics)

Percentage and Count of Male Players

Percentage and Count of Female Players

Percentage and Count of Other / Non-Disclosed

In [4]:

# Calculate the Number and Percentage by Gender

gender\_demographics\_totals = player\_demographics["Gender"].value\_counts()

gender\_demographics\_percents = gender\_demographics\_totals / num\_players \* 100

gender\_demographics = pd.DataFrame({"Total Count": gender\_demographics\_totals, "Percentage of Players": gender\_demographics\_percents})

# Minor Data Munging

gender\_demographics = gender\_demographics.round(2)

gender\_demographics

Out[4]:

|  | Total Count | Percentage of Players |
| --- | --- | --- |
| Male | 484 | 84.03 |
| Female | 81 | 14.06 |
| Other / Non-Disclosed | 11 | 1.91 |

Purchasing Analysis (Gender)[¶](https://render.githubusercontent.com/view/ipynb?commit=ece990c7351216059fa675fc9d9aaadb150d8117&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f52757467657273436f64696e67426f6f7463616d702f5255544a455232303138303944415441332f656365393930633733353132313630353966613637356663396439616161646231353064383131372f30342d50616e6461732f486f6d65776f726b2f536f6c7574696f6e732f4865726f65734f6650796d6f6c692f4865726f65734f6650796d6f6c695f536f6c7574696f6e2e6970796e623f746f6b656e3d416672474a534568376543585a5a3663714d2d694b65796943334d6e3774564b6b733563545161457741253344253344&nwo=RutgersCodingBootcamp%2FRUTJER201809DATA3&path=04-Pandas%2FHomework%2FSolutions%2FHeroesOfPymoli%2FHeroesOfPymoli_Solution.ipynb&repository_id=133828859&repository_type=Repository#Purchasing-Analysis-(Gender))

Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. by gender

Create a summary data frame to hold the results

Optional: give the displayed data cleaner formatting

Display the summary data frame

In [5]:

gender\_purchase\_total = purchase\_data.groupby(["Gender"]).sum()["Price"].rename("Total Purchase Value")

gender\_average = purchase\_data.groupby(["Gender"]).mean()["Price"].rename("Average Purchase Price")

gender\_counts = purchase\_data.groupby(["Gender"]).count()["Price"].rename("Purchase Count")

# Calculate Normalized Purchasing (Average Total Purchase per Person)

normalized\_total = gender\_purchase\_total / gender\_demographics["Total Count"]

# Convert to DataFrame

gender\_data = pd.DataFrame({"Purchase Count": gender\_counts, "Average Purchase Price": gender\_average, "Total Purchase Value": gender\_purchase\_total, "Normalized Totals": normalized\_total})

# Minor Data Munging

gender\_data["Average Purchase Price"] = gender\_data["Average Purchase Price"].map("${:,.2f}".format)

gender\_data["Total Purchase Value"] = gender\_data["Total Purchase Value"].map("${:,.2f}".format)

gender\_data ["Purchase Count"] = gender\_data["Purchase Count"].map("{:,}".format)

gender\_data["Avg Total Purchase per Person"] = gender\_data["Normalized Totals"].map("${:,.2f}".format)

gender\_data = gender\_data.loc[:, ["Purchase Count", "Average Purchase Price", "Total Purchase Value", "Avg Total Purchase per Person"]]

gender\_data

Out[5]:

|  | Purchase Count | Average Purchase Price | Total Purchase Value | Avg Total Purchase per Person |
| --- | --- | --- | --- | --- |
| Gender |  |  |  |  |
| Female | 113 | $3.20 | $361.94</td> <td>$4.47 |  |
| Male | 652 | $3.02</td> <td>$1,967.64 | $4.07</td> </tr> <tr> <th>Other / Non-Disclosed</th> <td>15</td> <td>$3.35 | $50.19</td> <td>$4.56 |

Age Demographics[¶](https://render.githubusercontent.com/view/ipynb?commit=ece990c7351216059fa675fc9d9aaadb150d8117&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f52757467657273436f64696e67426f6f7463616d702f5255544a455232303138303944415441332f656365393930633733353132313630353966613637356663396439616161646231353064383131372f30342d50616e6461732f486f6d65776f726b2f536f6c7574696f6e732f4865726f65734f6650796d6f6c692f4865726f65734f6650796d6f6c695f536f6c7574696f6e2e6970796e623f746f6b656e3d416672474a534568376543585a5a3663714d2d694b65796943334d6e3774564b6b733563545161457741253344253344&nwo=RutgersCodingBootcamp%2FRUTJER201809DATA3&path=04-Pandas%2FHomework%2FSolutions%2FHeroesOfPymoli%2FHeroesOfPymoli_Solution.ipynb&repository_id=133828859&repository_type=Repository#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

In [6]:

# Establish the bins

age\_bins = [0, 9.90, 14.90, 19.90, 24.90, 29.90, 34.90, 39.90, 99999]

group\_names = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]

# Categorize the existing players using the age bins

player\_demographics["Age Ranges"] = pd.cut(player\_demographics["Age"], age\_bins, labels=group\_names)

# Calculate the Numbers and Percentages by Age Group

age\_demographics\_totals = player\_demographics["Age Ranges"].value\_counts()

age\_demographics\_percents = age\_demographics\_totals / num\_players \* 100

age\_demographics = pd.DataFrame({"Total Count": age\_demographics\_totals, "Percentage of Players": age\_demographics\_percents})

# Minor Data Munging

age\_demographics = age\_demographics.round(2)

# Display Age Demographics Table

age\_demographics.sort\_index()

Out[6]:

|  | Total Count | Percentage of Players |
| --- | --- | --- |
| <10 | 17 | 2.95 |
| 10-14 | 22 | 3.82 |
| 15-19 | 107 | 18.58 |
| 20-24 | 258 | 44.79 |
| 25-29 | 77 | 13.37 |
| 30-34 | 52 | 9.03 |
| 35-39 | 31 | 5.38 |
| 40+ | 12 | 2.08 |

Purchasing Analysis (Age)[¶](https://render.githubusercontent.com/view/ipynb?commit=ece990c7351216059fa675fc9d9aaadb150d8117&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f52757467657273436f64696e67426f6f7463616d702f5255544a455232303138303944415441332f656365393930633733353132313630353966613637356663396439616161646231353064383131372f30342d50616e6461732f486f6d65776f726b2f536f6c7574696f6e732f4865726f65734f6650796d6f6c692f4865726f65734f6650796d6f6c695f536f6c7574696f6e2e6970796e623f746f6b656e3d416672474a534568376543585a5a3663714d2d694b65796943334d6e3774564b6b733563545161457741253344253344&nwo=RutgersCodingBootcamp%2FRUTJER201809DATA3&path=04-Pandas%2FHomework%2FSolutions%2FHeroesOfPymoli%2FHeroesOfPymoli_Solution.ipynb&repository_id=133828859&repository_type=Repository#Purchasing-Analysis-(Age))

Bin the purchase\_data data frame by age

Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. in the table below

Create a summary data frame to hold the results

Optional: give the displayed data cleaner formatting

Display the summary data frame

In [7]:

# Bin the Purchasing Data

purchase\_data["Age Ranges"] = pd.cut(purchase\_data["Age"], age\_bins, labels=group\_names)

# Run basic calculations

age\_purchase\_total = purchase\_data.groupby(["Age Ranges"]).sum()["Price"].rename("Total Purchase Value")

age\_average = purchase\_data.groupby(["Age Ranges"]).mean()["Price"].rename("Average Purchase Price")

age\_counts = purchase\_data.groupby(["Age Ranges"]).count()["Price"].rename("Purchase Count")

# Calculate Normalized Purchasing (Average Purchase Total per Person)

normalized\_total = age\_purchase\_total / age\_demographics["Total Count"]

# Convert to DataFrame

age\_data = pd.DataFrame({"Purchase Count": age\_counts, "Average Purchase Price": age\_average, "Total Purchase Value": age\_purchase\_total, "Normalized Totals": normalized\_total})

# Minor Data Munging

age\_data["Average Purchase Price"] = age\_data["Average Purchase Price"].map("${:,.2f}".format)

age\_data["Total Purchase Value"] = age\_data["Total Purchase Value"].map("${:,.2f}".format)

age\_data ["Purchase Count"] = age\_data["Purchase Count"].map("{:,}".format)

age\_data["Avg Total Purchase per Person"] = age\_data["Normalized Totals"].map("${:,.2f}".format)

age\_data = age\_data.loc[:, ["Purchase Count", "Average Purchase Price", "Total Purchase Value", "Avg Total Purchase per Person"]]

# Display the Age Table

age\_data

Out[7]:

|  | Purchase Count | Average Purchase Price | Total Purchase Value | Avg Total Purchase per Person |
| --- | --- | --- | --- | --- |
| 10-14 | 28 | $2.96 | $82.78</td> <td>$3.76 |  |
| 15-19 | 136 | $3.04</td> <td>$412.89 | $3.86</td> </tr> <tr> <th>20-24</th> <td>365</td> <td>$3.05 | $1,114.06</td> <td>$4.32 |
| 25-29 | 101 | $2.90</td> <td>$293.00 | $3.81</td> </tr> <tr> <th>30-34</th> <td>73</td> <td>$2.93 | $214.00</td> <td>$4.12 |
| 35-39 | 41 | $3.60</td> <td>$147.67 | $4.76</td> </tr> <tr> <th>40+</th> <td>13</td> <td>$2.94 | $38.24</td> <td>$3.19 |
| <10 | 23 | $3.35</td> <td>$77.13 | $4.54 |  |

Top Spenders[¶](https://render.githubusercontent.com/view/ipynb?commit=ece990c7351216059fa675fc9d9aaadb150d8117&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f52757467657273436f64696e67426f6f7463616d702f5255544a455232303138303944415441332f656365393930633733353132313630353966613637356663396439616161646231353064383131372f30342d50616e6461732f486f6d65776f726b2f536f6c7574696f6e732f4865726f65734f6650796d6f6c692f4865726f65734f6650796d6f6c695f536f6c7574696f6e2e6970796e623f746f6b656e3d416672474a534568376543585a5a3663714d2d694b65796943334d6e3774564b6b733563545161457741253344253344&nwo=RutgersCodingBootcamp%2FRUTJER201809DATA3&path=04-Pandas%2FHomework%2FSolutions%2FHeroesOfPymoli%2FHeroesOfPymoli_Solution.ipynb&repository_id=133828859&repository_type=Repository#Top-Spenders)

Run basic calculations to obtain the results in the table below

Create a summary data frame to hold the results

Sort the total purchase value column in descending order

Optional: give the displayed data cleaner formatting

Display a preview of the summary data frame

In [8]:

# Calculations

user\_total = purchase\_data.groupby(["SN"]).sum()["Price"].rename("Total Purchase Value")

user\_average = purchase\_data.groupby(["SN"]).mean()["Price"].rename("Average Purchase Price")

user\_count = purchase\_data.groupby(["SN"]).count()["Price"].rename("Purchase Count")

# Convert to DataFrame

user\_data = pd.DataFrame({"Total Purchase Value": user\_total, "Average Purchase Price": user\_average, "Purchase Count": user\_count})

# Display Table

user\_sorted = user\_data.sort\_values("Total Purchase Value", ascending=False)

# Minor Data Munging

user\_sorted["Average Purchase Price"] = user\_sorted["Average Purchase Price"].map("${:,.2f}".format)

user\_sorted["Total Purchase Value"] = user\_sorted["Total Purchase Value"].map("${:,.2f}".format)

user\_sorted = user\_sorted.loc[:,["Purchase Count", "Average Purchase Price", "Total Purchase Value"]]

# Display DataFrame

user\_sorted.head(5)

Out[8]:

|  | Purchase Count | Average Purchase Price | Total Purchase Value |
| --- | --- | --- | --- |
| SN |  |  |  |
| Lisosia93 | 5 | $3.79 | $18.96</td> </tr> <tr> <th>Idastidru52</th> <td>4</td> <td>$3.86 | $15.45</td> </tr> <tr> <th>Chamjask73</th> <td>3</td> <td>$4.61 | $13.83</td> </tr> <tr> <th>Iral74</th> <td>4</td> <td>$3.40 | $13.62</td> </tr> <tr> <th>Iskadarya95</th> <td>3</td> <td>$4.37 | $13.10 |

Most Popular Items[¶](https://render.githubusercontent.com/view/ipynb?commit=ece990c7351216059fa675fc9d9aaadb150d8117&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f52757467657273436f64696e67426f6f7463616d702f5255544a455232303138303944415441332f656365393930633733353132313630353966613637356663396439616161646231353064383131372f30342d50616e6461732f486f6d65776f726b2f536f6c7574696f6e732f4865726f65734f6650796d6f6c692f4865726f65734f6650796d6f6c695f536f6c7574696f6e2e6970796e623f746f6b656e3d416672474a534568376543585a5a3663714d2d694b65796943334d6e3774564b6b733563545161457741253344253344&nwo=RutgersCodingBootcamp%2FRUTJER201809DATA3&path=04-Pandas%2FHomework%2FSolutions%2FHeroesOfPymoli%2FHeroesOfPymoli_Solution.ipynb&repository_id=133828859&repository_type=Repository#Most-Popular-Items)

Retrieve the Item ID, Item Name, and Item Price columns

Group by Item ID and Item Name. Perform calculations to obtain purchase count, item price, and total purchase value

Create a summary data frame to hold the results

Sort the purchase count column in descending order

Optional: give the displayed data cleaner formatting

Display a preview of the summary data frame

In [9]:

# Extract item Data

item\_data = purchase\_data.loc[:,["Item ID", "Item Name", "Price"]]

# Perform basic calculations

total\_item\_purchase = item\_data.groupby(["Item ID", "Item Name"]).sum()["Price"].rename("Total Purchase Value")

average\_item\_purchase = item\_data.groupby(["Item ID", "Item Name"]).mean()["Price"]

item\_count = item\_data.groupby(["Item ID", "Item Name"]).count()["Price"].rename("Purchase Count")

# Create new DataFrame

item\_data\_pd = pd.DataFrame({"Total Purchase Value": total\_item\_purchase, "Item Price": average\_item\_purchase, "Purchase Count": item\_count})

# Sort Values

item\_data\_count\_sorted = item\_data\_pd.sort\_values("Purchase Count", ascending=False)

# Minor Data Munging

item\_data\_count\_sorted["Item Price"] = item\_data\_count\_sorted["Item Price"].map("${:,.2f}".format)

item\_data\_count\_sorted["Purchase Count"] = item\_data\_count\_sorted["Purchase Count"].map("{:,}".format)

item\_data\_count\_sorted["Total Purchase Value"] = item\_data\_count\_sorted["Total Purchase Value"].map("${:,.2f}".format)

item\_popularity = item\_data\_count\_sorted.loc[:,["Purchase Count", "Item Price", "Total Purchase Value"]]

item\_popularity.head(5)

Out[9]:

|  |  | Purchase Count | Item Price | Total Purchase Value |
| --- | --- | --- | --- | --- |
| Item ID | Item Name |  |  |  |
| 178 | Oathbreaker, Last Hope of the Breaking Storm | 12 | $4.23 | $50.76</td> </tr> <tr> <th>145</th> <th>Fiery Glass Crusader</th> <td>9</td> <td>$4.58 | $41.22</td> </tr> <tr> <th>108</th> <th>Extraction, Quickblade Of Trembling Hands</th> <td>9</td> <td>$3.53 | $31.77</td> </tr> <tr> <th>82</th> <th>Nirvana</th> <td>9</td> <td>$4.90 | $44.10</td> </tr> <tr> <th>19</th> <th>Pursuit, Cudgel of Necromancy</th> <td>8</td> <td>$1.02 | $8.16 |

Most Profitable Items[¶](https://render.githubusercontent.com/view/ipynb?commit=ece990c7351216059fa675fc9d9aaadb150d8117&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f52757467657273436f64696e67426f6f7463616d702f5255544a455232303138303944415441332f656365393930633733353132313630353966613637356663396439616161646231353064383131372f30342d50616e6461732f486f6d65776f726b2f536f6c7574696f6e732f4865726f65734f6650796d6f6c692f4865726f65734f6650796d6f6c695f536f6c7574696f6e2e6970796e623f746f6b656e3d416672474a534568376543585a5a3663714d2d694b65796943334d6e3774564b6b733563545161457741253344253344&nwo=RutgersCodingBootcamp%2FRUTJER201809DATA3&path=04-Pandas%2FHomework%2FSolutions%2FHeroesOfPymoli%2FHeroesOfPymoli_Solution.ipynb&repository_id=133828859&repository_type=Repository#Most-Profitable-Items)

Sort the above table by total purchase value in descending order

Optional: give the displayed data cleaner formatting

Display a preview of the data frame

In [10]:

# Item Table (Sorted by Total Purchase Value)

item\_total\_purchase = item\_data\_pd.sort\_values("Total Purchase Value", ascending=False)

# Minor Data Munging

item\_total\_purchase["Item Price"] = item\_total\_purchase["Item Price"].map("${:,.2f}".format)

item\_total\_purchase["Purchase Count"] = item\_total\_purchase["Purchase Count"].map("{:,}".format)

item\_total\_purchase["Total Purchase Value"] = item\_total\_purchase["Total Purchase Value"].map("${:,.2f}".format)

item\_profitable = item\_total\_purchase.loc[:,["Purchase Count", "Item Price", "Total Purchase Value"]]

item\_profitable.head(5)

Out[10]:

|  |  | Purchase Count | Item Price | Total Purchase Value |
| --- | --- | --- | --- | --- |
| Item ID | Item Name |  |  |  |
| 178 | Oathbreaker, Last Hope of the Breaking Storm | 12 | $4.23 | $50.76</td> </tr> <tr> <th>82</th> <th>Nirvana</th> <td>9</td> <td>$4.90 | $44.10</td> </tr> <tr> <th>145</th> <th>Fiery Glass Crusader</th> <td>9</td> <td>$4.58 | $41.22</td> </tr> <tr> <th>92</th> <th>Final Critic</th> <td>8</td> <td>$4.88 | $39.04</td> </tr> <tr> <th>103</th> <th>Singed Scalpel</th> <td>8</td> <td>$4.35 | $34.80 |