In [2]:
```
# Dependencies and Setup
import pandas as pd
import numpy as np

# File to Load (Remember to Change These)
purchase_data =  pd.read_csv("purchase_data.csv")

purchase_df = pd.DataFrame(purchase_data)
purchase_df.head()
#Q1 Print total number of players.
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-2-a5d26af2f180> in <module>()
      4 
      5 # File to Load (Remember to Change These)
----> 6 purchase_data =  pd.read_csv("purchase_data.csv")
      7 
      8 purchase_df = pd.DataFrame(purchase_data)

C:\Intel\New folder\lib\site-packages\pandas\io\parsers.py in parser_f(filepa
th_or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, pre
fix, mangle_dupe_cols, dtype, engine, converters, true_values, false_values,
 skipinitialspace, skiprows, nrows, na_values, keep_default_na, na_filter, ve
rbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, d
ate_parser, dayfirst, iterator, chunksize, compression, thousands, decimal, l
ineterminator, quotechar, quoting, escapechar, comment, encoding, dialect, tu
pleize_cols, error_bad_lines, warn_bad_lines, skipfooter, doublequote, delim_
whitespace, low_memory, memory_map, float_precision)
    676                     skip_blank_lines=skip_blank_lines)
    677 
--> 678         return _read(filepath_or_buffer, kwds)
    679 
    680     parser_f.__name__ = name

C:\Intel\New folder\lib\site-packages\pandas\io\parsers.py in _read(filepath_
or_buffer, kwds)
    438 
    439     # Create the parser.
--> 440     parser = TextFileReader(filepath_or_buffer, **kwds)
    441 
    442     if chunksize or iterator:

C:\Intel\New folder\lib\site-packages\pandas\io\parsers.py in __init__(self,
 f, engine, **kwds)
    785             self.options['has_index_names'] = kwds['has_index_names']
    786 
--> 787         self._make_engine(self.engine)
    788 
    789     def close(self):

C:\Intel\New folder\lib\site-packages\pandas\io\parsers.py in _make_engine(se
lf, engine)
   1012     def _make_engine(self, engine='c'):
   1013         if engine == 'c':
-> 1014             self._engine = CParserWrapper(self.f, **self.options)
   1015         else:
   1016             if engine == 'python':

C:\Intel\New folder\lib\site-packages\pandas\io\parsers.py in __init__(self,
 src, **kwds)
   1706         kwds['usecols'] = self.usecols
   1707 
-> 1708         self._reader = parsers.TextReader(src, **kwds)
   1709 
   1710         passed_names = self.names is None

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader.__cinit__()
```

```
pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._setup_parser_sou
rce()

FileNotFoundError: File b'purchase_data.csv' does not exist
```

In [ ]:
```
#PLAYER COUNT
#Q1 Print total number of players.

total_players = purchase_df["SN"].nunique()

total_players
```

In [ ]:
```
#PURCHASING ANALYSIS:
# Q2 Number of unique items

unique_items = pd.Series(purchase_df["Item ID"]).nunique()
unique_items
```

In [ ]:
```
# Q3Average purchase price
average_item_price = purchase_df["Price"].mean()
average_item_price
```

In [ ]:
```
#Q4 Total number of purchases
total_purchase = purchase_df["Purchase ID"].count()
total_purchase
```

In [ ]:
```
#Q5 Total Revenue

total_revenue = purchase_df['Price'].sum()
total_revenue
Purchase_Analysis = pd.DataFrame({"Total_players": [total_players],
                                  "Unique_items": [unique_items],
                                  "Total_purchase":[total_purchase],
                                  "Average_item_price": [average_item_price],
                                  "Total_revenue": [total_revenue]})
Purchase_Analysis.round(2)
```

In [ ]:
```
# GENDER DEMOGRAPHICS(Method 1)

#Q6 Percentage and count of Male players
total_gender_count = purchase_df["SN"].count()
total_gender_count
Unique_Gender_counts = purchase_df['Gender'].value_counts()
Unique_Gender_counts
Total_perc_count = (Unique_Gender_counts / total_gender_count)*100
Total_perc_count
Gender_Type = pd.Series(purchase_df["Gender"]).unique()
Gender_Type
#Gender_df = pd.DataFrame({"Gender": ["Male", "Female", "Other"],
                      #    "Total Count": [Unique_Gender_counts],
                     #     "Percentage of Players": [Total_perc_count]
                      #   })
#Gender_df.set_index('Gender')
#Gender_df
```

In [ ]:
```
# GENDER DEMOGRAPHICS(Method 2)

Group_Gender = purchase_df[[ "Gender"]]
Group_Gender
#male_count = Group_Gender.loc[Group_Gender.Gender == "Male"].count()
#male_count1= (purchase_df.Gender == "Male").count()
male_count = purchase_df.query('Gender == "Male"').Gender.count()
male_count
female_count = purchase_df.query('Gender == "Female"').Gender.count()
female_count
#other_non-disclosed_count = purchase_df.query('Gender == "Other / Non-Disclos
ed"').Gender.count()
Other_df= purchase_df.groupby(['Gender']).get_group(('Other / Non-Disclosed'))
.count()
other_count = Other_df.count()
other_count

Male_Percent = male_count/total_gender_count*100
Female_Percent = female_count/total_gender_count*100
Other_Percent = other_count/total_gender_count*100

Gender_demo_df =pd.DataFrame({"Gender": ["Male", "Female", "Other/Non-Disclose
d"],
                             "Total Count": [male_count, female_count, other_c
ount],
                             "Percentage of Players": [Male_Percent, Female_Pe
rcent,Other_Percent]})
Gender_demo_df.set_index('Gender').round(2)
```

In [ ]:
```
# Purchasing Analysis
#Q1 Get Purchase count, avg purchase price, avg purchase total per person by g
ender

# Purchase count
male_price_count = purchase_df.groupby(['Gender']).get_group(('Male'))['Price'
].sum()
male_avg_price= male_price_count/male_count
female_price_count = purchase_df.groupby(['Gender']).get_group(('Female'))['Pr
ice'].sum()
female_avg_price=female_price_count/female_count
other_price_count = purchase_df.groupby(['Gender']).get_group(('Other / Non-Di
sclosed'))['Price'].sum()
other_avg_price= other_price_count/other_count
male_purch_perperson = male_price_count/total_purchase
male_purch_perperson
female_purch_perperson = female_price_count/total_purchase
other_purch_perperson = other_price_count/total_purchase
other_purch_perperson

purch_analysis_df = pd.DataFrame({"Gender": ["Male", "Female", "Other/non-disc
losed"],
                                  "Purchase Count": [male_count, female_count,
 other_count],
                                  "Average Purchase Price": [male_avg_price, f
emale_avg_price, other_avg_price],
                                  "Total purchase value": [male_price_count, f
emale_price_count, other_price_count],
                                  "Avg Total Purchase per Person":[male_purch_
perperson, female_purch_perperson ,other_purch_perperson ] })
purch_analysis_df.set_index('Gender').round(2)
```

In [ ]:
```
#Age Demographics
max_age = purchase_df['Age'].max()
min_age = purchase_df['Age'].min()
print(max_age, min_age)
#establish bins

bins =[0,10,14,19,24,29,34,39,45]
label_names = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "4
0+"]

purchase_df["Age_bin"] = pd.cut(purchase_df["Age"], bins, labels=label_names)
```

In [ ]:
```
bin_df = purchase_df.copy()
bin_df["Age_bin"] = pd.cut(bin_df["Age"], bins, labels=label_names)
bin_df.head()
```

In [3]:
```
#Total count for Age demographivs and purchase count for Purchase analysis (Ag
e)

count_10 = bin_df.groupby(["Age_bin"]).get_group(("<10"))["SN"].count()
count_10
count_14 = bin_df.groupby(["Age_bin"]).get_group(("10-14"))["SN"].count()
count_14
count_19 = bin_df.groupby(["Age_bin"]).get_group(("15-19"))["SN"].count()
count_24 = bin_df.groupby(["Age_bin"]).get_group(("20-24"))["SN"].count()
count_29 = bin_df.groupby(["Age_bin"]).get_group(("25-29"))["SN"].count()
count_34 = bin_df.groupby(["Age_bin"]).get_group(("30-34"))["SN"].count()
count_39 = bin_df.groupby(["Age_bin"]).get_group(("35-39"))["SN"].count()
count_45 = bin_df.groupby(["Age_bin"]).get_group(("40+"))["SN"].count()
totalcount = count_10 + count_14+ count_19 + count_24 + count_29 + count_34 +
count_39+count_45
Age_Demos = pd.DataFrame({"Bins":["<10", "10-14", "15-19", "20-24", "25-29",
"30-34", "35-39", "40+"],
                          "Total Counts":[count_10, count_14, count_19, count_
24, count_29, count_34, count_39, count_45],
                          "Percentage of players":[count_10/totalcount*100, co
unt_14/totalcount*100, count_19/totalcount*100, count_24/totalcount*100, count
_29/totalcount*100, count_34/totalcount*100, count_39/totalcount*100, count_45
/totalcount*100]
                           })

Age_Demos.set_index('Bins').round(2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-3-6b826485e68b> in <module>()
      1 #Total count for Age demographivs and purchase count for Purchase ana
lysis (Age)
      2 
----> 3 count_10 = bin_df.groupby(["Age_bin"]).get_group(("<10"))["SN"].count
()
      4 count_10
      5 count_14 = bin_df.groupby(["Age_bin"]).get_group(("10-14"))["SN"].cou
nt()

NameError: name 'bin_df' is not defined
```

In [4]:
```
# Age Purchase Analysis
pcount_10 = bin_df.groupby(["Age_bin"]).get_group(("<10"))["Price"].sum()

pcount_14 = bin_df.groupby(["Age_bin"]).get_group(("10-14"))["Price"].sum()
count_14
pcount_19 = bin_df.groupby(["Age_bin"]).get_group(("15-19"))["Price"].sum()
pcount_24 = bin_df.groupby(["Age_bin"]).get_group(("20-24"))["Price"].sum()
pcount_29 = bin_df.groupby(["Age_bin"]).get_group(("25-29"))["Price"].sum()
pcount_34 = bin_df.groupby(["Age_bin"]).get_group(("30-34"))["Price"].sum()
pcount_39 = bin_df.groupby(["Age_bin"]).get_group(("35-39"))["Price"].sum()
pcount_45 = bin_df.groupby(["Age_bin"]).get_group(("40+"))["Price"].sum()

totalpurchase = pcount_10 + pcount_14+ pcount_19 + pcount_24 + pcount_29 + pco
unt_34 + pcount_39+ pcount_45
avg_purch_count= [pcount_10/count_10, pcount_14/count_14, pcount_19/count_19,
pcount_24/count_24, pcount_29/count_29, pcount_34/count_34, pcount_39/count_39
, pcount_45/count_45]
Total_avg_purch_count= sum(avg_purch_count)
Age_Purch_Demos = pd.DataFrame({"Bins":["<10", "10-14", "15-19", "20-24", "25-
29", "30-34", "35-39", "40+"],
                          "Purchase Counts":[count_10, count_14, count_19, cou
nt_24, count_29, count_34, count_39, count_45],
                    "Average Purchase Count": [pcount_10/count_10, pcount_14/c
ount_14, pcount_19/count_19, pcount_24/count_24, pcount_29/count_29, pcount_34
/count_34, pcount_39/count_39, pcount_45/count_45],
                          "Total Purchase":[pcount_10, pcount_14, pcount_19, p
count_24, pcount_29, pcount_34, pcount_39, pcount_45],
                         "Avg total Purchase per person": [pcount_10/Total_avg
_purch_count, pcount_14/Total_avg_purch_count, pcount_19/Total_avg_purch_count
, pcount_24/Total_avg_purch_count, pcount_29/Total_avg_purch_count, pcount_34/
Total_avg_purch_count, pcount_39/Total_avg_purch_count, pcount_45/Total_avg_pu
rch_count]    })

#Total_avg_purch_count = Age_Purch_Demos["Average Purchase Count"].sum()
Age_Purch_Demos.set_index('Bins').round(2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-4-c9ff64374739> in <module>()
      1 # Age Purchase Analysis
----> 2 pcount_10 = bin_df.groupby(["Age_bin"]).get_group(("<10"))["Price"].s
um()
      3 
      4 pcount_14 = bin_df.groupby(["Age_bin"]).get_group(("10-14"))["Price"]
.sum()
      5 count_14

NameError: name 'bin_df' is not defined
```

In [5]:
```
# Top Spenders

new_df = purchase_df[["SN", "Price", ]]
new_df
purch_count = purchase_df.groupby(["SN"])["Price"].count()
total_purch = purchase_df.groupby(["SN"])["Price"].sum()
avg_purch = total_purch/purch_count

Top_spend = pd.DataFrame({"Purchase Count": purch_count,
                         "Average Purchase Price": avg_purch,
                         "Total Purchase Value": total_purch})
Top_spend.sort_values("Total Purchase Value", ascending = False).head(5).round
(2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-5-ce916d6baa07> in <module>()
      1 # Top Spenders
      2 
----> 3 new_df = purchase_df[["SN", "Price", ]]
      4 new_df
      5 purch_count = purchase_df.groupby(["SN"])["Price"].count()

NameError: name 'purchase_df' is not defined
```

In [6]:
```
# Most Popular Items
df_new = purchase_df.loc[:,["Item ID","Item Name", "Price"]]

#profitable_item =df_new.sort_values()
purch_count=df_new.groupby([ "Item ID","Item Name"]).count()["Price"].rename(
"Purchase Count")

#purchase_df.sort_values(by=["Item Name"])
purch_count
item_price =df_new.groupby(["Item ID","Item Name"]).mean()["Price"].rename("It
em Price")
total_purchase= df_new.groupby(["Item ID","Item Name"]).sum()["Price"].rename(
"Total Purchase Value")

Summary_data = pd.DataFrame({"Purchase Count":purch_count,
                             "Item Price": item_price,
                             "Total Purchase Value": total_purchase} )


popular_items=Summary_data.sort_values("Purchase Count", ascending =False)
popular_items

Summary_data
#total_purchase
#item_count =df_new.groupby(["Item Name"]).count()
#item_count.max()

#group_itemnameid = purchase_df.groupby(["Item ID"]).count()
#group_itemnameid
most_profitable = Summary_data.sort_values("Total Purchase Value", ascending =
 False)

most_profitable
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-6-21524ae8d03d> in <module>()
      1 # Most Popular Items
----> 2 df_new = purchase_df.loc[:,["Item ID","Item Name", "Price"]]
      3 
      4 #profitable_item =df_new.sort_values()
      5 purch_count=df_new.groupby([ "Item ID","Item Name"]).count()["Price"]
.rename("Purchase Count")

NameError: name 'purchase_df' is not defined
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

```
In [ ]:
```