Ecommerce Purchases Exercise

In this Exercise you will be given some Fake Data about some purchases done through Amazon! Just go ahead and follow the directions and try your best to answer the questions and complete the tasks. Feel free to reference the solutions. Most of the tasks can be solved in different ways. For the most part, the questions get progressively harder.

Please excuse anything that doesn't make "Real-World" sense in the dataframe, all the data is fake and made-up.

Also note that all of these questions can be answered with one line of code.

Import pandas and read in the Ecommerce Purchases csv file and set it to a DataFrame called ecom.

In [84]:
```
```

In [6]:
```
import pandas as pd
```

Check the head of the DataFrame.

In [14]:
```
ecom=pd.read_csv("Ecommerce Purchases.csv")
df.head()
```

Out[14]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	CC Provider	Er
0	16629 Pace Camp Apt. 448\nAlexisborough, NE 77...	46 in	PM	Opera/9.56.(X11; Linux x86_64; sl-SI) Presto/2...	Martinez-Herman	6011929061123406	02/20	900	JCB 16 digit	pdunlap@yahoo.c
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28 rn	PM	Opera/8.93.(Windows 98; Win 9x 4.90; en-US) Pr...	Fletcher, Richards and Whitaker	3337758169645356	11/18	561	Mastercard	anthony41@reed.c
2	Unit 0065 Box 5052\nDPO AP 27450	94 vE	PM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Simpson, Williams and Pham	675957666125	08/19	699	JCB 16 digit	amymiller@morar harrison.c
3	7780 Julia Fords\nNew Stacy, WA 45798	36 vm	PM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_0 ...	Williams, Marshall and Buchanan	6011578504430710	02/24	384	Discover	brent16@olson-robinson.
4	23012 Munoz Drive Suite 337\nNew Cynthia, TX 5...	20 IE	AM	Opera/9.58.(X11; Linux x86_64; it-IT) Presto/2...	Brown, Watson and Andrews	6011456623207998	10/25	678	Diners Club / Carte Blanche	christopherwright@gmail.c

How many rows and columns are there?

In [15]:
```
ecom.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Address           10000 non-null  object 
 1   Lot               10000 non-null  object 
 2   AM or PM          10000 non-null  object 
 3   Browser Info      10000 non-null  object 
 4   Company           10000 non-null  object 
 5   Credit Card       10000 non-null  int64  
 6   CC Exp Date       10000 non-null  object 
 7   CC Security Code  10000 non-null  int64  
 8   CC Provider       10000 non-null  object 
 9   Email             10000 non-null  object 
 10  Job               10000 non-null  object 
 11  IP Address        10000 non-null  object 
 12  Language          10000 non-null  object 
 13  Purchase Price    10000 non-null  float64
dtypes: float64(1), int64(2), object(11)
memory usage: 1.1+ MB
```

What is the average Purchase Price?

In [16]:
```
ecom["Purchase Price"].mean()
```

Out[16]:
```
50.34730200000025
```

What were the highest and lowest purchase prices?

In [21]:
```
ecom["Purchase Price"].max()
```

Out[21]:
```
99.99
```

In [22]:
```
ecom["Purchase Price"].min()
```

Out[22]:
```
0.0
```

How many people have English 'en' as their Language of choice on the website?

In [23]:
```
ecom[ecom["Language"]=="en"].count()
```

Out[23]:
```
Address             1098
Lot                 1098
AM or PM            1098
Browser Info        1098
Company             1098
Credit Card         1098
CC Exp Date         1098
CC Security Code    1098
CC Provider         1098
Email               1098
Job                 1098
IP Address          1098
Language            1098
Purchase Price      1098
dtype: int64
```

How many people have the job title of "Lawyer" ?

In [25]:
```
ecom[ecom["Job"]=="Lawyer"].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30 entries, 470 to 9979
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Address           30 non-null     object 
 1   Lot               30 non-null     object 
 2   AM or PM          30 non-null     object 
 3   Browser Info      30 non-null     object 
 4   Company           30 non-null     object 
 5   Credit Card       30 non-null     int64  
 6   CC Exp Date       30 non-null     object 
 7   CC Security Code  30 non-null     int64  
 8   CC Provider       30 non-null     object 
 9   Email             30 non-null     object 
 10  Job               30 non-null     object 
 11  IP Address        30 non-null     object 
 12  Language          30 non-null     object 
 13  Purchase Price    30 non-null     float64
dtypes: float64(1), int64(2), object(11)
memory usage: 3.5+ KB
```

How many people made the purchase during the AM and how many people made the purchase during PM ?

(Hint: Check out value_counts())

In [26]:
```
ecom["AM or PM"].value_counts()
```

Out[26]:
```
PM    5068
AM    4932
Name: AM or PM, dtype: int64
```

What are the 5 most common Job Titles?

In [27]:
```
ecom["Job"].value_counts().head(5)
```

Out[27]:
```
Interior and spatial designer    31
Lawyer                           30
Social researcher                28
Purchasing manager               27
Research officer, political party    27
Name: Job, dtype: int64
```

Someone made a purchase that came from Lot: "90 WT" , what was the Purchase Price for this transaction?

In [30]:
```
ecom[ecom["Lot"]=="90 WT"]["Purchase Price"]
```

Out[30]:
```
513    75.1
Name: Purchase Price, dtype: float64
```

What is the email of the person with the following Credit Card Number: 4926535242672853

In [31]:
```
ecom[ecom["Credit Card"]==4926535242672853]["Email"]
```

Out[31]:
```
1234    bondellen@williams-garza.com
Name: Email, dtype: object
```

How many people have American Express as their Credit Card Provider *and* made a purchase above $95 ?

In [32]:
```
ecom[(ecom["CC Provider"]=="American express")&(ecom["Purchase Price"]>95)].count()
```

Out[32]:
```
Address             0
Lot                 0
AM or PM            0
Browser Info        0
Company             0
Credit Card         0
CC Exp Date         0
CC Security Code    0
CC Provider         0
Email               0
Job                 0
IP Address          0
Language            0
Purchase Price      0
dtype: int64
```

Hard: How many people have a credit card that expires in 2025?

In [33]:
```
sum(ecom["CC Exp Date"].apply(lambda x: x[3:])=="25")
```

Out[33]:
```
1033
```

Hard: What are the top 5 most popular email providers/hosts (e.g. gmail.com, yahoo.com, etc...)

In [34]:
```
ecom["Email"].apply(lambda x: x.split("@")[1]).value_counts().head(5)
```

Out[34]:
```
hotmail.com     1638
yahoo.com       1616
gmail.com       1605
smith.com         42
williams.com      37
Name: Email, dtype: int64
```

Great Job!