



▼ 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. ****

Check the head of the DataFrame.

```
import pandas as pd
df=pd.read_csv("Ecommerce Purchases.csv")
df.head()
```

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	Secu
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	
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	
				Mozilla/5.0				

**** How many rows and columns are there? ****

21450

vindows and Pnam

df.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? ****

df["Purchase Price"].mean()

50.34730200000025

**** What were the highest and lowest purchase prices? ****

```
df["Purchase Price"].max()
```

```
99.99
```

```
df["Purchase Price"].min()
```

```
0.0
```

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

```
df[df["Language"]=="en"].count()
```

```
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" ? ****

```
df[df["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\(\)](#)) **

```

df["AM or PM"].value_counts()

PM      5068
AM      4932
Name: AM or PM, dtype: int64

```

**** What are the 5 most common Job Titles? ****

```

df["Job"].value_counts().head(5)

Interior and spatial designer    31
Lawyer                          30
Social researcher                28
Purchasing manager              27
Designer, jewellery             27
Name: Job, dtype: int64

```

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

```

df[df["Lot"] == "90 WT"]["Purchase Price"]

513      75.1
Name: Purchase Price, dtype: float64

```

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

```

df[df["Credit Card"]== 4926535242672853]["Email"]

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 ?****

```
df[(df["CC Provider"]=="American Express")&(df["Purchase Price"]>95)].count()
```

```
Address      39
Lot          39
AM or PM     39
Browser Info 39
Company      39
Credit Card  39
CC Exp Date  39
CC Security Code 39
CC Provider  39
Email        39
Job          39
IP Address   39
Language     39
Purchase Price 39
dtype: int64
```

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

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

```
1033
```

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

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

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

Great Job!

