In [1]:

```
import pandas as pd
```

In [15]:

```
df= pd.read_csv('Ecommerce Purchases')
```

Check the head of the DataFrame.

In [16]:

```
df.head()
```

Out[16]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	
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	
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	M
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	
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	
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	

How many rows and columns are there?

In [18]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
Address             10000 non-null object
Lot                 10000 non-null object
AM or PM            10000 non-null object
Browser Info        10000 non-null object
Company             10000 non-null object
Credit Card         10000 non-null int64
CC Exp Date         10000 non-null object
CC Security Code    10000 non-null int64
CC Provider         10000 non-null object
Email               10000 non-null object
Job                 10000 non-null object
IP Address          10000 non-null object
Language            10000 non-null object
Purchase Price      10000 non-null float64
dtypes: float64(1), int64(2), object(11)
memory usage: 664.1+ KB
```

What is the average Purchase Price?

In [20]:

```
df['Purchase Price'].mean()
```

Out[20]:

```
50.347302
```

What were the highest and lowest purchase prices?

In [22]:

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

Out[22]:

```
0.0
```

In [23]:

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

Out[23]:

```
99.99
```

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

In [25]:

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

Out[25]:

```
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 [28]:

```
df[df['Job']=='lawyer'].count()
```

Out[28]:

```
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
```

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

(Hint: Check out value_counts() (http://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.value_counts.html))

In [39]:

```
df['AM or PM'].value_counts()
```

Out[39]:

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

What are the 5 most common Job Titles?

In [50]:

```
df['Job'].value_counts().head(5)
```

Out[50]:

```
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?

In [51]:

```
df[df['Lot'] == '90 WT']['Purchase Price']
```

Out[51]:

```
513    75.1
Name: Purchase Price, dtype: float64
```

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

In [53]:

```
df[df['Credit Card'] == 4926535242672853]['Email']
```

Out[53]:

```
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 [58]:

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

Out[58]:

```
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?

In [62]:

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

Out[62]:

```
1033
```

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

In [63]:

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

Out[63]:

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

Great Job!