

Ecommerce Purchases Exercise

```
In [2]: import pandas as pd
```

```
In [3]: ecomm=pd.read_csv('Ecommerce Purchases')
```

Check the head of the DataFrame.

In [5]: ecomm.head()

Out[5]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	CC Provider	Email
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.com
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.com
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@morales- harrison.com
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.info
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.com

How many rows and columns are there?

```
In [6]: ecomm.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: 1.1+ MB
```

What is the average Purchase Price?

```
In [7]: ecomm['Purchase Price'].mean()
```

```
Out[7]: 50.34730200000025
```

What were the highest and lowest purchase prices?

```
In [8]: ecomm['Purchase Price'].max()
```

```
Out[8]: 99.989999999999995
```

```
In [9]: ecomm['Purchase Price'].min()
```

```
Out[9]: 0.0
```

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

```
In [11]: ecomm[ecomm['Language']=='en']['Language'].value_counts()
```

```
Out[11]: en      1098  
         Name: Language, dtype: int64
```

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

```
In [13]: ecomm[ecomm['Job']=='Lawyer']['Job'].value_counts()
```

```
Out[13]: Lawyer      30  
         Name: Job, 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 [14]: ecomm['AM or PM'].value_counts()
```

```
Out[14]: PM      5068  
         AM      4932  
         Name: AM or PM, dtype: int64
```

What are the 5 most common Job Titles?

```
In [17]: ecomm['Job'].value_counts().head(5)
```

```
Out[17]: Interior and spatial designer    31  
         Lawyer                        30  
         Social researcher                28  
         Research officer, political party 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 [21]: ecomm[ecomm['Lot']=='90 WT']['Purchase Price']
```

```
Out[21]: 513    75.1  
         Name: Purchase Price, dtype: float64
```

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

```
In [26]: ecomm[ecomm['Credit Card']== 4926535242672853]['Email']
```

```
Out[26]: 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 [44]: ecomm[(ecomm['CC Provider']=='American Express') & (ecomm['Purchase Price'] > 95)]['Email'].count()
```

```
Out[44]: 39
```

Find how many cards are of Diners

```
In [32]: ecomm[ecomm['CC Provider'].apply(lambda x: x[0:6]=='Diners)]['CC Provider'].count()
```

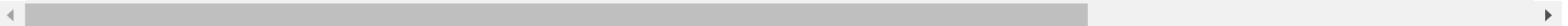
```
Out[32]: 767
```

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

```
In [16]: ecomm[ecomm['CC Exp Date'].apply(lambda x: x[3:]=='25')].head()
```

Out[16]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	CC Provider	Email
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@gmai
5	7502 Powell Mission Apt. 768\nTravisland, VA 3...	21 XT	PM	Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_8_5...	Silva- Anderson	30246185196287	07/25	7169	Discover	ynguyen@gmail.com
7	260 Rachel Plains Suite 366\nCastroberg, WV 24...	96 pG	PM	Mozilla/5.0 (X11; Linux i686) AppleWebKit/5350...	Marshall- Collins	561252141909	06/25	256	VISA 13 digit	phillip48@parks.info
13	118 Melton Via Suite 681\nAlexanderbury, FL 32104	36 bu	PM	Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_8...	Keller PLC	4603635169938574	01/25	557	VISA 16 digit	caitlin57@yahoo.com
15	31730 Chelsea Crest\nBlakemouth, CT 90395-0620	41 Cj	PM	Opera/8.95. (Windows NT 5.0; en-US) Presto/2.9....	Garcia- Steele	180069437020404	04/25	404	Diners Club / Carte Blanche	amanda39@yahoo.com



```
In [28]: sum(ecomm['CC Exp Date'].apply(lambda x: x[3:]=='25'))
```

Out[28]: 1033

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

```
In [75]: ecomm['Email'].apply(lambda email: email.split('@')[1]).value_counts().head(5)
```

```
Out[75]: hotmail.com      1638  
         yahoo.com       1616  
         gmail.com       1605  
         smith.com        42  
         williams.com     37  
         Name: Email, dtype: int64
```