In [1]:
```
import pandas as pd
```

In [2]:
```
data = pd.read_csv('Ecommerce Purchases')
data
```

Loading [MathJax]/extensions/Safe.js

Out[2]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card
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
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
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
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
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
...	...	...	...	...	...	...
3807	48419 James Villages\nCharlesside, MN 05281	16 Od	PM	Mozilla/5.0 (iPod; U; CPU iPhone OS 3_0 like M...	Gray Group	3158950215096082
3808	41420 Morgan Square\nNew Sylviaton, NH 91906	04 lg	AM	Mozilla/5.0 (iPod; U; CPU iPhone OS 4_3 like M...	Sanchez-Pugh	30144466817368
3809	308 Dixon Fords Suite 923\nNicoleton, MH 45145	52 YN	AM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_8)...	Graham, Walker and Frazier	869929666344262
3810	7016 Richard Center Apt. 216\nNew Paulburgh, C...	53 WU	AM	Mozilla/5.0 (Macintosh; PPC Mac OS X 10_8_4; r...	Kim, Harris and Lee	3528700667851234
3811	625 Mclaughlin Inlet Apt. 375\nSouth Ruben, CO...	94 he	PM	Mozilla/5.0 (compatible; MSIE 9.0; Windows CE;...	Banks Group	869974335345498

Loading [MathJax]/extensions/Safe.js

3812 rows × 14 columns

1. Display Top 10 Rows of the Dataset

In [5]:
```
data.head(10)
```

Out[5]:

	Address	Lot	AM or PM	Browser Info	Company	Cre
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	601192906
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	333775816
2	Unit 0065 Box 5052\nDPO AP 27450	94 vE	PM	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT ...	Simpson, Williams and Pham	67595
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	601157850
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	601145662
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	3024618
6	93971 Conway Causeway\nAndersonburgh, AZ 75107	96 Xt	AM	Mozilla/5.0 (compatible; MSIE 7.0; Windows NT ...	Gibson and Sons	601139878
7	260 Rachel Plains Suite 366\nCastroberg, WV 24...	96 pG	PM	Mozilla/5.0 (X11; Linux i686) AppleWebKit/5350...	Marshall-Collins	56125
8	2129 Dylan Burg\nNew Michelle, ME 28650	45 JN	PM	Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_7...	Galloway and Sons	18004179
9	3795 Dawson Extensions\nLake Tinafort, ID 88739	15 Ug	AM	Mozilla/5.0 (X11; Linux i686; rv:1.9.7.20) Gec...	Rivera, Buchanan and Ramirez	439628

2. Check Last 10 Rows of the Dataset

Loading [MathJax]/extensions/Safe.js

In [7]:
```
data.tail(10)
```

Loading [MathJax]/extensions/Safe.js

Out[7]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Car
3802	0843 Bobby Port\nGordonbury, VT 23015-8557	57 sr	AM	Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_5_4...	Cruz PLC	601108172021829
3803	92679 Brown Parks\nEast Jason, IA 85954	53 lx	AM	Mozilla/5.0 (Macintosh; PPC Mac OS X 10_8_4; r...	Ward Inc	34781278176534
3804	76678 Melissa Track\nSnowville, MP 02252	20 HG	AM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_5 ...	Meyer-Howe	67594064946
3805	USNS Anderson\nFPO AP 85662-4006	65 vG	PM	Mozilla/5.0 (Windows; U; Windows NT 5.0) Apple...	York-Baker	63048648328
3806	6598 Chad Extensions\nNorristown, TX 53321	49 GN	AM	Mozilla/5.0 (Windows; U; Windows NT 5.01) Appl...	Ho LLC	308870163087760
3807	48419 James Villages\nCharlesside, MN 05281	16 Od	PM	Mozilla/5.0 (iPod; U; CPU iPhone OS 3_0 like M...	Gray Group	315895021509608
3808	41420 Morgan Square\nNew Sylviaton, NH 91906	04 lg	AM	Mozilla/5.0 (iPod; U; CPU iPhone OS 4_3 like M...	Sanchez-Pugh	3014446681736
3809	308 Dixon Fords Suite 923\nNicoleton, MH 45145	52 YN	AM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_8)...	Graham, Walker and Frazier	86992966634426
3810	7016 Richard Center Apt. 216\nNew Paulburgh, C...	53 WU	AM	Mozilla/5.0 (Macintosh; PPC Mac OS X 10_8_4; r...	Kim, Harris and Lee	352870066785123
3811	625 Mclaughlin Inlet Apt. 375\nSouth Ruben, CO...	94 he	PM	Mozilla/5.0 (compatible; MSIE 9.0; Windows CE;...	Banks Group	86997433534549

Loading [MathJax]/extensions/Safe.js

3. Check Datatype of Each column

In [8]:
```
data.dtypes
```

Out[8]:
```
Address              object
Lot                  object
AM or PM             object
Browser Info         object
Company              object
Credit Card           int64
CC Exp Date          object
CC Security Code      int64
CC Provider          object
Email                object
Job                  object
IP Address           object
Language             object
Purchase Price      float64
dtype: object
```

4. Check Null Values in the dataset

In [10]:
```
data.isnull().sum()
```

Out[10]:
```
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            1
Purchase Price      1
dtype: int64
```

5. How many rows and columns are there in our dataset?

In [12]:
```
len(data.columns)
```

Out[12]:
```
14
```

In [13]:
```
len(data)
```

Loading [MathJax]/extensions/Safe.js

Out[13]: 3812

In [14]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3812 entries, 0 to 3811
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Address           3812 non-null   object 
 1   Lot               3812 non-null   object 
 2   AM or PM          3812 non-null   object 
 3   Browser Info      3812 non-null   object 
 4   Company           3812 non-null   object 
 5   Credit Card       3812 non-null   int64  
 6   CC Exp Date       3812 non-null   object 
 7   CC Security Code  3812 non-null   int64  
 8   CC Provider       3812 non-null   object 
 9   Email             3812 non-null   object 
 10  Job               3812 non-null   object 
 11  IP Address        3812 non-null   object 
 12  Language          3811 non-null   object 
 13  Purchase Price    3811 non-null   float64
dtypes: float64(1), int64(2), object(11)
memory usage: 417.1+ KB
```

6. Highest and lowest Purchase prices

In [15]:
```
data.columns
```

Out[15]:
```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Car
d',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

In [16]:
```
data['Purchase Price']
```

Out[16]:
```
0       98.14
1       70.73
2        0.95
3       78.04
4       77.82
        ...
3807    70.32
3808     8.30
3809     3.73
3810    49.73
3811      NaN
Name: Purchase Price, Length: 3812, dtype: float64
```

In [17]:
```
data['Purchase Price'].max()
```

Loading [MathJax]/extensions/Safe.js

Out[17]: 99.99

In [18]:
```
data['Purchase Price'].min()
```

Out[18]: 0.0

7. Average Purchase Price

In [19]:
```
data['Purchase Price'].mean()
```

Out[19]: 50.30394647074259

8. How many people have french 'fr' as their Language?

In [20]:
```
data.columns
```

Out[20]:
```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Car
d',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

In [25]:
```
data[data['Language'] == 'fr']
```

Loading [MathJax]/extensions/Safe.js

Out[25]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card
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
19	125 Hall Summit\nBoothton, IL 41721	99 CU	PM	Mozilla/5.0 (compatible; MSIE 7.0; Windows NT ...	Turner-Mckinney	676343504830
53	PSC 9431, Box 7059\nAPO AA 29285-1363	14 qD	AM	Opera/9.34. (X11; Linux x86_64; it-IT) Presto/2...	Higgins, Cardenas and Kennedy	869972604798355
76	49206 Campbell Port\nNorth Cliffordshire, HI 3...	71 iu	PM	Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_5_9...	Jacobs-Tucker	6011343518820988
82	493 Smith Valleys Suite 004\nNew Madelineville...	35 ls	PM	Mozilla/5.0 (iPod; U; CPU iPhone OS 4_0 like M...	Robinson, Johnston and Valdez	4351359627548412
...	...	...	...	...	...	...
3762	0926 Rachel Extension Suite 013\nWalkerstad, W...	68 yh	AM	Mozilla/5.0 (compatible; MSIE 7.0; Windows NT ...	Young, Schmidt and Aguilar	346974891556275
3771	52156 Phillips Haven\nTylerfort, KS 62987-6005	37 wg	AM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_9;...	Richard-Jackson	676132781672
3786	Unit 0568 Box 2984\nDPO AE 61274-6097	87 BW	PM	Opera/9.74. (X11; Linux x86_64; sl-SI) Presto/2...	Barber-White	30086395868246
3798	8461 Morales Way Apt. 798\nHernandezside, WA 6...	79 aL	PM	Mozilla/5.0 (compatible; MSIE 8.0; Windows CE;...	Smith PLC	4112009432819785
3809	308 Dixon Fords Suite 923\nNicoleton, MH 45145	52 YN	AM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_8)...	Graham, Walker and Frazier	869929666344262

Loading [MathJax]/extensions/Safe.js

453 rows × 14 columns

In [26]:
```
len(data[data['Language'] == 'fr'])
```

Out[26]:
```
453
```

In [27]:
```
data[data['Language'] == 'fr'].count()
```

Out[27]:
```
Address             453
Lot                 453
AM or PM            453
Browser Info        453
Company             453
Credit Card         453
CC Exp Date         453
CC Security Code    453
CC Provider         453
Email               453
Job                 453
IP Address          453
Language            453
Purchase Price      453
dtype: int64
```

9. Job Title contains Engineer

In [28]:
```
data.columns
```

Out[28]:
```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Car
d',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

In [29]:
```
data['Job']
```

Out[29]:
```
0       Scientist, product/process development
1                            Drilling engineer
2                    Customer service manager
3                            Drilling engineer
4                                  Fine artist
                          ...
3807                 Plant breeder/geneticist
3808        Environmental health practitioner
3809                              Hotel manager
3810                 Engineer, communications
3811                    Multimedia specialist
Name: Job, Length: 3812, dtype: object
```

In [38]:
```
len(data[data['Job'].str.contains('Engineer',case=False)])
```

Out[38]:
```
389
```

Loading [MathJax]/extensions/Safe.js

10. Find Email of the person with the following IP address:132.207.160.22

In [39]:
```
data[data.columns
```

Out[39]:
```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Car
d',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

In [44]:
```
data[data['IP Address']=="132.207.160.22"]['Email']
```

Out[44]:
```
2    amymiller@morales-harrison.com
Name: Email, dtype: object
```

11. How many people have mastercard as their credit card provider and made a purchase above 50?

In [45]:
```
data.columns
```

Out[45]:
```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Car
d',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

In [51]:
```
len(data[(data['CC Provider']=="Mastercard") & (data['Purchase Price']>50)])
```

Out[51]:
```
176
```

12. Find Email of the Person with the following credit card number: 675957666125

In [52]:
```
data.columns
```

Out[52]:
```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Car
d',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

In [60]:
```
data[data['Credit Card']==675957666125]['Email']
```

Loading [MathJax]/extensions/Safe.js

Out[60]:
```
2    amymiller@morales-harrison.com
Name: Email, dtype: object
```

13. How many people purchase during the AM and How many people Purchase During PM?

In [61]:
```
data.columns
```

Out[61]:
```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Car
d',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

In [63]:
```
data['AM or PM'].value_counts()
```

Out[63]:
```
AM or PM
PM    1975
AM    1837
Name: count, dtype: int64
```

14. How many people have a credit card that Expires in 2020?

In [64]:
```
data.columns
```

Out[64]:
```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Car
d',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

In [69]:
```
data['CC Exp Date']
```

Out[69]:
```
0       02/20
1       11/18
2       08/19
3       02/24
4       10/25
        ...
3807    02/17
3808    02/24
3809    04/24
3810    11/18
3811    01/22
Name: CC Exp Date, Length: 3812, dtype: object
```

In [70]:
```
def fun():
    =0
```

Loading [MathJax]/extensions/Safe.js

```
    for date in data['CC Exp Date']:
        if date.split('/')[1]=='20':
            count=count+1
    print(count)
```

In [71]:
```
fun()
```

```
361
```

In [74]:
```
len(data[data['CC Exp Date'].apply(lambda x:x[3:]=='20')])
```

Out[74]:
```
361
```

15. Top 5 most popular email providers(e.g.gmail.com,yahoo.com,etc..

In [75]:
```
list1 = []
for email in data['Email']:
    list1.append(email.split('@')[1])
```

In [77]:
```
data['temp'] = list1
```

In [78]:
```
data.head(1)
```

Out[78]:

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date
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	02/2(

In [80]:
```
data['temp'].value_counts().head()
```

Out[80]:
```
temp
yahoo.com      647
hotmail.com    635
gmail.com      585
smith.com       14
johnson.com     12
Name: count, dtype: int64
```

In [82]:
```
data['Email'].apply(lambda x:x.split('@')[1]).value_counts().head()
```

Loading [MathJax]/extensions/Safe.js

Out[82]:
```
Email
yahoo.com      647
hotmail.com    635
gmail.com      585
smith.com       14
johnson.com     12
Name: count, dtype: int64
```

Thank You

In [ ]:

Loading [MathJax]/extensions/Safe.js