# Analysis of E-Commerce Dataset

Author: Waleed

The dataset is downloaded from Kaggle and can be found here

The data dictionary provided for this dataset is as follow:

- 'Address' - customer's address
- 'AM or PM' - time of the day
- 'Browser Info' - info regarding the browser of the customer
- 'Company' - the company in which the customer work
- 'Credit Card' - number of the customer's credit card
- 'CC Exp Date' - the expiray date of teh customer's credit card
- 'CC Security Code' - the security code of the customer's credit card
- 'CC Provider' - name of the caompany provided the credit card
- 'Email' - customer's email
- 'Job' - customer's job title
- 'IP Address' - customers' IP Address
- 'Language' - customer's language
- 'Purchase Price' - price of the item purchased

The aim of the this project is to analyze the included sales records

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
pd.set_option('precision',2)
```

```
data['Credit Card']
```

```
0       6011929061123406
1       3337758169645356
2           675957666125
3       6011578504430710
4       6011456623207998
              ...
9995     342945015358701
9996     210033169205009
9997    6011539787356311
9998     180003348082930
9999    4139972901927273
Name: Credit Card, Length: 10000, dtype: int64
```

```
data = pd.read_csv('Ecommerce Purchases.csv')
data.head()
```

| | Address object | Lot object | AM or PM object | Browser Info o… | Company object | Crec |
|---|---|---|---|---|---|---|
| 0 | 16629 Pace Camp Apt. 448… | 46 in | PM | Opera/9.56. (X11; Linux… | Martinez-Herman | 601 |
| 1 | 9374 Jasmine Spurs Suite 50… | 28 rn | PM | Opera/8.93. (Windows 98;… | Fletcher, Richards and… | 333 |
| 2 | Unit 0065 Box 5052 DPO AP… | 94 vE | PM | Mozilla/5.0 (compatible;… | Simpson, Williams and… | |
| 3 | 7780 Julia Fords New… | 36 vm | PM | Mozilla/5.0 (Macintosh;… | Williams, Marshall and… | 601 |
| 4 | 23012 Munoz Drive Suite 33… | 20 IE | AM | Opera/9.58. (X11; Linux… | Brown, Watson and Andrews | 601 |

```
data.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
```

There are 14 columns and 10000 rows in this dataset

# Analysis of the 'AM or PM' column

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

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

**Inference**:
The majority of the orders were made at night.

```
display(data.groupby('AM or PM').agg({'Purchase Price':'mean'}).\
                rename(columns = {'Purchase Price':'Avg. Spent'}),
data.groupby('AM or PM').agg({'Purchase Price':'sum'}).\
                rename(columns = {'Purchase Price':'Tot. Spent'})
)
```

| AM or PM | Avg. Spent |
|---|---|
| AM | 50.19 |
| PM | 50.50 |

| | Tot. Spent floa… |
|---|---|

| | | |
|---|---|---|
| AM | 247519.8699999998 | |
| PM | 255953.1499999992 | |

**Inference**:
On average the values of items purchased in the evening and at night are higher of those in the morning. Moreover, the total value of the evening and nightly purcahses is higher of that of the morning purchases

# Analysis of 'Borwser Info' Column

```
data['Browser Info'].value_counts()
```

```
Mozilla/5.0 (compatible; MSIE 9.0; Windows CE; Trident/4.1)
Mozilla/5.0 (compatible; MSIE 5.0; Windows NT 6.0; Trident/3.0
Mozilla/5.0 (compatible; MSIE 9.0; Windows 95; Trident/4.1)
Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 6.2; Trident/3.1
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 5.01; Trident/5.

Mozilla/5.0 (Macintosh; PPC Mac OS X 10_7_3 rv:6.0; it-IT) App
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_5_8 rv:3.0; en-US
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_7_4; rv:1.9.3.20)
Mozilla/5.0 (Windows NT 6.0; it-IT; rv:1.9.0.20) Gecko/2014-09-
Mozilla/5.0 (iPod; U; CPU iPhone OS 4_0 like Mac OS X; sl-SI) A
Name: Browser Info, Length: 8287, dtype: int64
```

```
data['Browser Info'].str.split('/').str.get(0).value_counts(normalize = True)
```

```
Mozilla    0.79
Opera      0.21
Name: Browser Info, dtype: float64
```

**Inference**:
The majority of the customers uses Mozilla browser

# Analysis of 'Company' Column

```
data['Company'].value_counts().head(7)
```

```
Brown Ltd         15
Smith Group       15
Smith PLC         13
Smith LLC         13
Williams LLC      12
Davis and Sons    11
Smith and Sons    11
Name: Company, dtype: int64
```

**Inference**:

The top 7 company in term of the quantity of purchases are:

'Brown Ltd'
'Smith Group'
'Smith PLC'
'Smith LLC'
'Williams LLC'
'Smith and Sons'
'Davis and Sons'

```
data.groupby('Company').agg({'Purchase Price':'sum'}).\
                sort_values('Purchase Price', ascending = False).\
                rename(columns = {'Purchase Price':'Revenue'}).head(7)
```

| | Revenue float64 |
|---|---|
| Brown Ltd | 796.38 |
| Williams LLC | 723.6300000000001 |
| Smith LLC | 637.92 |
| Smith PLC | 601.37 |
| Johnson Ltd | 600.61 |
| Smith Group | 592.5400000000001 |
| Johnson PLC | 567.3299999999999 |

**Inference**:
The top 7 companies in term of revenue are:

'Brown Ltd'
'Williams LLC'
'Smith LLC'
'Smith PLC'
'Johnson Ltd'
'Smith Group'
'Johnson PLC'

# Analysis of the 'CC Provider' Column

```
display(data['CC Provider'].value_counts(),
        data.groupby('CC Provider').agg({'Purchase Price':'sum'}).\
        sort_values('Purchase Price',ascending =False).\
        rename(columns = {'Purchase Price':'Tot. Spent'}),
        data.groupby('CC Provider').agg({'Purchase Price':'mean'}).\
        sort_values('Purchase Price',ascending =False).\
        rename(columns = {'Purchase Price':'Avg. Spent'})
       )
```

```
JCB 16 digit                  1716
VISA 16 digit                 1715
JCB 15 digit                   868
American Express               849
Maestro                        846
Voyager                        829
Discover                       817
Mastercard                     816
VISA 13 digit                  777
Diners Club / Carte Blanche    767
Name: CC Provider, dtype: int64
```

| | Tot. Spent |
|---|---|
| **CC Provider** | |
| **VISA 16 digit** | 85528.86 |
| **JCB 16 digit** | 84597.33 |
| **JCB 15 digit** | 44376.60 |
| **Voyager** | 43085.77 |
| **American Express** | 42865.52 |
| **Maestro** | 42620.78 |
| **Discover** | 42208.13 |
| **Mastercard** | 40835.10 |
| **VISA 13 digit** | 39976.54 |
| **Diners Club / Carte Blanche** | 37378.39 |

| | Avg. Spent floa… 48.733233376792… |
|---|---|
| Voyager | 51.97318455971051 |
| Discover | 51.66233782129751 |
| VISA 13… | 51.44985842985841 |
| JCB 15 digit | 51.12511520737328 |
| American… | 50.48942285041218 |
| Maestro | 50.3791725768321 |
| Mastercard | 50.043014705882406 |
| VISA 16… | 49.871055393586 |
| JCB 16 digit | 49.299143356643256 |
| Diners Club … | 48.73323337679264 |

## Inferences:

### Quantity

JCB 16 digit is the card for the largest number of orders while Diners Club / Carte Blanche is card used for

the lowest number of orders.

**Total Revnue**
VISA 16 digit is the card associated with the largest revnue while Diners Club / Carte Blanche is the card associated with the lowest revenue.

**Avergae Revenue**
Voyager is the card with the highest average revenue while Diners Club / Carte Blanche is the card with the lowest avgerage revenue

# Analysis of 'Job' Column

```
data['Job'].value_counts().head(10)
```

```
Interior and spatial designer        31
Lawyer                               30
Social researcher                    28
Purchasing manager                   27
Research officer, political party    27
Designer, jewellery                  27
Charity fundraiser                   26
Social worker                        26
Dietitian                            26
Special educational needs teacher    26
Name: Job, dtype: int64
```

```
print('The top 10 most common jobs within the customers are:')
for job in data['Job'].value_counts().head(10).index:
    print(job)
```

```
The top 10 most common jobs within the customers are:
Interior and spatial designer
Lawyer
Social researcher
Purchasing manager
Research officer, political party
Designer, jewellery
Charity fundraiser
Social worker
Dietitian
Special educational needs teacher
```

**Inference**

The top 10 most common jobs within the customers are:

- Interior and spatial designer
- Lawyer
- Social researcher
- Purchasing manager
- Research officer, political party
- Designer, jewellery
- Charity fundraiser
- Social worker
- Dietitian
- Special educational needs teacher

```
job_top_10_total_rv = data.groupby('Job').agg({'Purchase Price':'sum'}).\
                        sort_values('Purchase Price', ascending = False).head(10).\
                        rename(columns ={'Purchase Price':' Total Rev.'})
job_top_10_avg_rv = data.groupby('Job').agg({'Purchase Price':'mean'}).\
                        sort_values('Purchase Price', ascending = False).head(10).\
                        rename(columns = {'Purchase Price':'Average Rev.'})

display(job_top_10_total_rv, job_top_10_avg_rv)
```

| | Total Rev. |
|---|---|
| **Job** | |
| **Dietitian** | 1605.30 |
| **Lawyer** | 1603.85 |
| **Purchasing manager** | 1577.97 |
| **Therapist, art** | 1526.31 |
| **Clinical cytogeneticist** | 1495.92 |
| **Research officer, political party** | 1488.79 |
| **Designer, jewellery** | 1482.20 |
| **Interior and spatial designer** | 1466.20 |
| **Network engineer** | 1421.73 |
| **Social researcher** | 1416.34 |

| | Average Rev. f… |
|---|---|
| | 66.034285714285… |

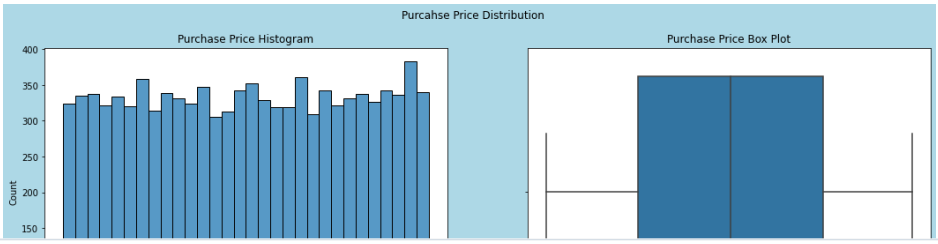
| | | |
|---|---|---|
| | | |
| Trade mark… | 73.80272727272727 | |
| Transl ator | 70.77764705882353 | |
| Invest ment… | 69.09 | |
| Accoun tant,… | 69.08266666666667 | |
| Design er,… | 68.71466666666666 | |
| Clinic al… | 67.99636363636365 | |
| Psycho logis… | 67.76722222222223 | |
| Specia l… | 66.75285714285714 | |
| Politi cian'… | 66.10692307692308 | |
| Advert ising… | 66.0342857142857 | |

# Analysis of the 'Purchase Price' Column

```
plt.figure(figsize = (18,6), facecolor = 'lightblue')
plt.suptitle('Purcahse Price Distribution')

plt.subplot(1,2,1)
sns.histplot(x = data['Purchase Price'], bins = 30)
plt.title('Purchase Price Histogram')


plt.subplot(1,2,2)
sns.boxplot(x = data['Purchase Price'])
plt.title('Purchase Price Box Plot')
plt.show()
```




```
print('Maximum Purchase Price')
print(data['Purchase Price'].max())
print('Minimum Purchase Price')
print(data['Purchase Price'].min())
print('Average Purchase Price')
print(data['Purchase Price'].mean())
```

```
Maximum Purchase Price
99.99
Minimum Purchase Price
0.0
Average Purchase Price
50.347302
```

# Business Questions

### What are the email(s) of the cusomters(s) with the Purchase Price of 0.0?

```
data.loc[data ['Purchase Price'] == data['Purchase Price'].min()]
```

| | Address object | Lot object | AM or PM object | Browser Info o… | Company object | Crec |
|---|---|---|---|---|---|---|
| 2876 | 332 Jones Parkways East… | 39 GT | AM | Mozilla/5.0 (Macintosh; U;… | Lyons, Diaz and Clark | 420 |
| 5487 | 465 Mallory Ways North… | 93 OH | PM | Mozilla/5.0 (X11; Linux… | Flynn and Sons | 30 |

### How many poeple have English as their language?

```
((data['Language']=='en') & (data['Job'] == 'Social researcher')).sum()
```

```
3
```

### What are the Emails for the customer working as a "Investment Analyst" and have"English" as their language?

```
data.loc[(data['Job'] == 'Investment analyst') & (data['Language'] == 'en'), ['Email']]
```

| | Email object |
|---|---|
| 8470 | charlesjeremy@hill.com |

### Show the email of customer with ip address of 156.210.0.254?

```
data.loc[data['IP Address'] == '156.210.0.254', ['Email']]
```

| | Email object |
|---|---|
| 9997 | tyler16@gmail.com |



```
len(data.loc[(data['CC Provider'] == 'VISA 16 digit') & (d )
```

```
518
```

### Find the emails of the customers with credit cards of 6011456623207998 and 180003348082930??

```
data.loc[data['Credit Card'].isin([6011578504430710,6011456623207998,180003348082930]), ['Ema
```

| | Email object |
|---|---|

| | | |
|---|---|---|
| | | |
| 3 | brent16@olson-robinson.info | |
| 4 | christopherwright@gmail.com | |
| 9998 | elizabethmoore@reid.net | |

How many customers have cardit cards that expires in 2022?

```
print('Number of customer with cardit cards expiring in 2022')
len(data.loc[data['CC Exp Date'].str.split('/').str.get(1) == '22'])
```

```
Number of customer with cardit cards expiring in 2022
```

```
996
```

## What is the most popular email provider for the customer?

```
print('The most popualr email provider for the customer is')
data['Email'].str.split('@').str.get(1).str.split('.').str.get(0).value_counts().head(1)
```

```
The most popualr email provider for the customer is
```

```
hotmail    1638
Name: Email, dtype: int64
```