In [17]:
```python
import pandas as pd
import numpy as np
from pandas import Series, DataFrame
import xlsxwriter
import datetime as dt
```

In [18]:
```python
#read the first excel file
df_exams = pd.read_excel('DataAnalyst_Ecom_data_sessionCounts (2).xlsx', parse_dates = ['dim_date'])
df_exams.head()
```

Out[18]:

	dim_browser	dim_deviceCategory	dim_date	sessions	transactions	QTY
0	Safari	tablet	2012-07-01	2928	127	221
1	Internet Explorer	desktop	2012-07-01	1106	28	0
2	Chrome	tablet	2012-07-01	474	3	13
3	Amazon Silk	tablet	2012-07-01	235	4	5
4	Internet Explorer	mobile	2012-07-01	178	6	11

In [50]:
```python
#checking the type of the data
df_exams.dtypes
```

Out[50]:
```
dim_browser                   object
dim_deviceCategory            object
dim_date              datetime64[ns]
sessions                       int64
transactions                   int64
QTY                            int64
Year                           int64
Month                         object
month_name                    object
dtype: object
```

In [ ]:

In [19]:
```python
# change the month to names as a way of ordering and cleaning data.
df_exams['Year'] = df_exams['dim_date'].dt.year
df_exams['Month'] = df_exams['dim_date'].dt.month_name()
df_exams
```

Out[19]:

	dim_browser	dim_deviceCategory	dim_date	sessions	transactions	QTY	Year	Month
0	Safari	tablet	2012-07-01	2928	127	221	2012	July
1	Internet Explorer	desktop	2012-07-01	1106	28	0	2012	July
2	Chrome	tablet	2012-07-01	474	3	13	2012	July
3	Amazon Silk	tablet	2012-07-01	235	4	5	2012	July
4	Internet Explorer	mobile	2012-07-01	178	6	11	2012	July
...	...	...	...	...	...	...	...	...
7729	Internet Explorer	tablet	2013-06-30	6	0	0	2013	June
7730	error	desktop	2013-06-30	5	0	0	2013	June
7731	Edge	mobile	2013-06-30	4	0	0	2013	June
7732	SeaMonkey	desktop	2013-06-30	3	0	0	2013	June
7733	Puffin	desktop	2013-06-30	1	0	0	2013	June

7734 rows × 8 columns

In [20]:
```python
df_exams['month_name'] = df_exams['dim_date'].dt.month_name()
```

In [21]:
```python
df_exams
```

Out[21]:

	dim_browser	dim_deviceCategory	dim_date	sessions	transactions	QTY	Year	Month	month_name
0	Safari	tablet	2012-07-01	2928	127	221	2012	July	July
1	Internet Explorer	desktop	2012-07-01	1106	28	0	2012	July	July
2	Chrome	tablet	2012-07-01	474	3	13	2012	July	July
3	Amazon Silk	tablet	2012-07-01	235	4	5	2012	July	July
4	Internet Explorer	mobile	2012-07-01	178	6	11	2012	July	July
...	...	...	...	...	...	...	...	...	...
7729	Internet Explorer	tablet	2013-06-30	6	0	0	2013	June	June
7730	error	desktop	2013-06-30	5	0	0	2013	June	June
7731	Edge	mobile	2013-06-30	4	0	0	2013	June	June
7732	SeaMonkey	desktop	2013-06-30	3	0	0	2013	June	June
7733	Puffin	desktop	2013-06-30	1	0	0	2013	June	June

7734 rows × 9 columns

In [ ]:

In [22]:
```python
#creating the pivot tble for the first sheet
classic = df_exams.pivot_table(index = ['Year', 'month_name', 'dim_deviceCategory'], values= ['sessions', 'transactions', 'QTY'], aggfunc ='sum')
df1 = pd.DataFrame(classic)
```

In [23]:
```python
#creating the new column for the ECR =  transactions / sessions
classic2 = df1.assign(ECR = lambda x: x.transactions / x.sessions)
df2 = pd.DataFrame(classic2)
df2
```

Out[23]:

Year	month_name	dim_deviceCategory	QTY	sessions	transactions	ECR
2012	August	desktop	23316	392079	12912	0.032932
		mobile	5572	275556	3165	0.011486
		tablet	5760	154858	3202	0.020677
	December	desktop	19947	309718	11613	0.037495
		mobile	5672	234481	3158	0.013468
		tablet	9133	245435	5158	0.021016
	July	desktop	18547	335429	10701	0.031902
		mobile	4557	274443	2576	0.009386
		tablet	8700	158717	4884	0.030772
	November	desktop	18778	320717	10350	0.032271
		mobile	3407	178828	1994	0.011150
		tablet	5947	138235	3183	0.023026
	October	desktop	17675	302682	9373	0.030966
		mobile	4446	238849	2418	0.010124
		tablet	4505	107108	2484	0.023192
	September	desktop	16507	272771	8898	0.032621
		mobile	4050	220689	2381	0.010789
		tablet	7869	169193	4379	0.025882
2013	April	desktop	34200	567510	18868	0.033247
		mobile	7752	429864	4280	0.009957
		tablet	12994	299239	7221	0.024131
	February	desktop	18437	247632	9699	0.039167
		mobile	3915	194996	2071	0.010621
		tablet	4696	107599	2396	0.022268
	January	desktop	25424	393723	13793	0.035032
		mobile	7257	341668	4360	0.012761
		tablet	6165	164601	3407	0.020699
	June	desktop	35146	554940	19370	0.034905
		mobile	13017	526481	7412	0.014078
		tablet	13728	307413	7756	0.025230
	March	desktop	17362	287837	9679	0.033627
		mobile	6455	304832	3644	0.011954
		tablet	8265	196151	4481	0.022845
	May	desktop	33208	526330	18176	0.034533
		mobile	9790	409796	5413	0.013209
		tablet	8631	228513	4800	0.021005

In [24]:
```python
# Modifying the pivot table to add it to the Monhth vs Month comparison
elvis = df_exams.pivot_table(index = ['Year', 'month_name'], values= ['sessions', 'transactions', 'QTY'], aggfunc ='sum')
elvis1 = elvis.assign(ECR = lambda x: x.transactions / x.sessions)
df3 = pd.DataFrame(elvis1)
df3
```

Out[24]:

Year	month_name	QTY	sessions	transactions	ECR
2012	August	34648	822493	19279	0.023440
	December	34752	789634	19929	0.025238
	July	31804	768589	18161	0.023629
	November	28132	637780	15527	0.024345
	October	26626	648639	14275	0.022008
	September	28426	662653	15658	0.023629
2013	April	54946	1296613	30369	0.023422
	February	27048	550227	14166	0.025746
	January	38846	899992	21560	0.023956
	June	61891	1388834	34538	0.024868
	March	32082	788820	17804	0.022570
	May	51629	1164639	28389	0.024376

In [25]:
```python
#importing the second data frame.
df_exams3 = pd.read_excel('DataAnalyst_Ecom_data_addsToCart (1).xlsx', parse_dates = ['dim_month'])
df_exams3
```

Out[25]:

	dim_year	dim_month	addsToCart
0	2012	7	191504
1	2012	8	217666
2	2012	9	123726
3	2012	10	139803
4	2012	11	186572
5	2012	12	168972
6	2013	1	147619
7	2013	2	135882
8	2013	3	109797
9	2013	4	183842
10	2013	5	136720
11	2013	6	107970

In [26]:
```python
#adding a date column and making 1st the day to have a full date format of YYYY-MM-DD
df_exams3['day'] = 1
df_exams3['date'] = pd.to_datetime(df_exams3['dim_year'].astype(str)+ '-' + df_exams3['dim_month'].astype(str) + '-'+ df_exams3['day'].astype(str), yearfirst = True)
df_exams3
```

Out[26]:

	dim_year	dim_month	addsToCart	day	date
0	2012	7	191504	1	2012-07-01
1	2012	8	217666	1	2012-08-01
2	2012	9	123726	1	2012-09-01
3	2012	10	139803	1	2012-10-01
4	2012	11	186572	1	2012-11-01
5	2012	12	168972	1	2012-12-01
6	2013	1	147619	1	2013-01-01
7	2013	2	135882	1	2013-02-01
8	2013	3	109797	1	2013-03-01
9	2013	4	183842	1	2013-04-01
10	2013	5	136720	1	2013-05-01
11	2013	6	107970	1	2013-06-01

In [27]:
```python
#adding the name of the month name
df_exams3['month_name'] = df_exams3['date'].dt.month_name()
df_exams3
```

Out[27]:

	dim_year	dim_month	addsToCart	day	date	month_name
0	2012	7	191504	1	2012-07-01	July
1	2012	8	217666	1	2012-08-01	August
2	2012	9	123726	1	2012-09-01	September
3	2012	10	139803	1	2012-10-01	October
4	2012	11	186572	1	2012-11-01	November
5	2012	12	168972	1	2012-12-01	December
6	2013	1	147619	1	2013-01-01	January
7	2013	2	135882	1	2013-02-01	February
8	2013	3	109797	1	2013-03-01	March
9	2013	4	183842	1	2013-04-01	April
10	2013	5	136720	1	2013-05-01	May
11	2013	6	107970	1	2013-06-01	June

In [ ]:

In [28]:
```python
#merging both the pivot table and the second dataframe to begin with the second worksheet.
df_all = pd.merge(df_exams3,df3, on = "month_name")
df_all
```

Out[28]:

	dim_year	dim_month	addsToCart	day	date	month_name	QTY	sessions	transactions	ECR
0	2012	7	191504	1	2012-07-01	July	31804	768589	18161	0.023629
1	2012	8	217666	1	2012-08-01	August	34648	822493	19279	0.023440
2	2012	9	123726	1	2012-09-01	September	28426	662653	15658	0.023629
3	2012	10	139803	1	2012-10-01	October	26626	648639	14275	0.022008
4	2012	11	186572	1	2012-11-01	November	28132	637780	15527	0.024345
5	2012	12	168972	1	2012-12-01	December	34752	789634	19929	0.025238
6	2013	1	147619	1	2013-01-01	January	38846	899992	21560	0.023956
7	2013	2	135882	1	2013-02-01	February	27048	550227	14166	0.025746
8	2013	3	109797	1	2013-03-01	March	32082	788820	17804	0.022570
9	2013	4	183842	1	2013-04-01	April	54946	1296613	30369	0.023422
10	2013	5	136720	1	2013-05-01	May	51629	1164639	28389	0.024376
11	2013	6	107970	1	2013-06-01	June	61891	1388834	34538	0.024868

In [29]:
```python
df_all
```

Out[29]:

	dim_year	dim_month	addsToCart	day	date	month_name	QTY	sessions	transactions	ECR
0	2012	7	191504	1	2012-07-01	July	31804	768589	18161	0.023629
1	2012	8	217666	1	2012-08-01	August	34648	822493	19279	0.023440
2	2012	9	123726	1	2012-09-01	September	28426	662653	15658	0.023629
3	2012	10	139803	1	2012-10-01	October	26626	648639	14275	0.022008
4	2012	11	186572	1	2012-11-01	November	28132	637780	15527	0.024345
5	2012	12	168972	1	2012-12-01	December	34752	789634	19929	0.025238
6	2013	1	147619	1	2013-01-01	January	38846	899992	21560	0.023956
7	2013	2	135882	1	2013-02-01	February	27048	550227	14166	0.025746
8	2013	3	109797	1	2013-03-01	March	32082	788820	17804	0.022570
9	2013	4	183842	1	2013-04-01	April	54946	1296613	30369	0.023422
10	2013	5	136720	1	2013-05-01	May	51629	1164639	28389	0.024376
11	2013	6	107970	1	2013-06-01	June	61891	1388834	34538	0.024868

In [30]:
```python
#the absolute differences for the second worksheet.
df_all['Absolute_ATC'] = df_all['addsToCart'].diff(1)
```

In [31]:
```python
df_all['Absolute_QTY'] = df_all['QTY'].diff(1)
```

In [32]:
```python
df_all['Absolute_trns'] = df_all['transactions'].diff(1)
```

In [33]:
```python
df_all['Absolute_ECR'] = df_all['ECR'].diff(1)
```

In [34]:
```python
df_all['Absolute_sess'] = df_all['sessions'].diff(1)
```

In [35]:
```python
#adding the relative differences which is absolutive difference / start value eg sessions, transactions eg
df_all['Relative_ATC'] = df_all['Absolute_ATC']/df_all['addsToCart']
df_all['Relative_QTY'] = df_all['Absolute_QTY']/df_all['QTY']
df_all['Relative_trans'] = df_all['Absolute_trns']/df_all['transactions']
df_all['Relative_QTY'] = df_all['Absolute_sess']/df_all['sessions']
```

In [36]:
```python
df_all
```

Out[36]:

	dim_year	dim_month	addsToCart	day	date	month_name	QTY	sessions	transactions	ECR	Absolute_ATC	Absolute_QTY	Absolute_trns	Absolute_ECR	Absolute_sess	Relative_ATC	Relative_QTY	Relative_trans
0	2012	7	191504	1	2012-07-01	July	31804	768589	18161	0.023629	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2012	8	217666	1	2012-08-01	August	34648	822493	19279	0.023440	26162.0	2844.0	1118.0	-0.000189	53904.0	0.120193	0.065537	0.057991
2	2012	9	123726	1	2012-09-01	September	28426	662653	15658	0.023629	-93940.0	-6222.0	-3621.0	0.000190	-159840.0	-0.759258	-0.241212	-0.231256
3	2012	10	139803	1	2012-10-01	October	26626	648639	14275	0.022008	16077.0	-1800.0	-1383.0	-0.001622	-14014.0	0.114998	-0.021605	-0.096883
4	2012	11	186572	1	2012-11-01	November	28132	637780	15527	0.024345	46769.0	1506.0	1252.0	0.002338	-10859.0	0.250675	-0.017026	0.080634
5	2012	12	168972	1	2012-12-01	December	34752	789634	19929	0.025238	-17600.0	6620.0	4402.0	0.000893	151854.0	-0.104159	0.192309	0.220884
6	2013	1	147619	1	2013-01-01	January	38846	899992	21560	0.023956	-21353.0	4094.0	1631.0	-0.001283	110358.0	-0.144649	0.122621	0.075649
7	2013	2	135882	1	2013-02-01	February	27048	550227	14166	0.025746	-11737.0	-11798.0	-7394.0	0.001790	-349765.0	-0.086376	-0.635674	-0.521954
8	2013	3	109797	1	2013-03-01	March	32082	788820	17804	0.022570	-26085.0	5034.0	3638.0	-0.003175	238593.0	-0.237575	0.302468	0.204336
9	2013	4	183842	1	2013-04-01	April	54946	1296613	30369	0.023422	74045.0	22864.0	12565.0	0.000851	507793.0	0.402764	0.391630	0.413744
10	2013	5	136720	1	2013-05-01	May	51629	1164639	28389	0.024376	-47122.0	-3317.0	-1980.0	0.000954	-131974.0	-0.344661	-0.113318	-0.069745
11	2013	6	107970	1	2013-06-01	June	61891	1388834	34538	0.024868	-28750.0	10262.0	6149.0	0.000493	224195.0	-0.266278	0.161427	0.178036

In [37]:
```python
#insert the data in a new workbook and in different sheets.
with pd.ExcelWriter('Submission1.xlsx') as writer:
    df2.to_excel(writer, sheet_name = "sheet1")
    df_all.to_excel(writer, sheet_name = "sheet2")
```

In [ ]:

In [ ]: