Problem Statement:

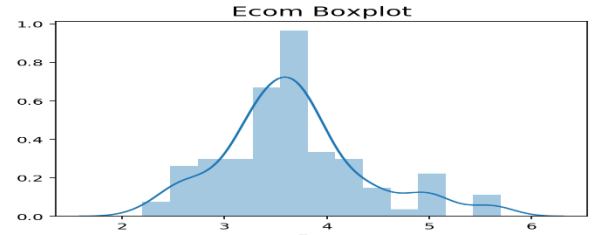
The 'Hair Salon.csv' dataset contains various variables used for the context of Market Segmentation. This particular case study is based on various parameters of a salon chain of hair products. You are expected to do Principal Component Analysis for this case study

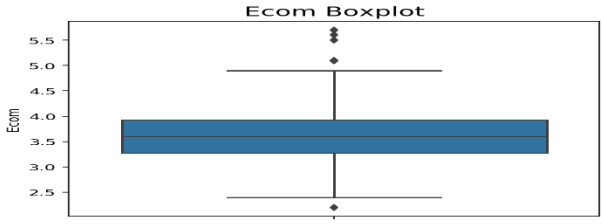
Exploratory Data Analysis:

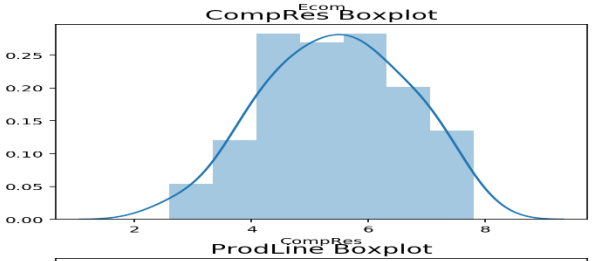
1) Perform Exploratory Data Analysis [both univariate and multivariate analysis to be performed. The inferences drawn from this should be properly documented.

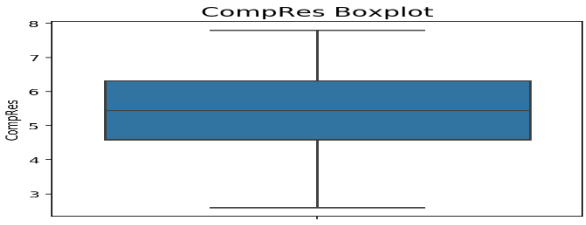
	ID	ProdQual	Ecom	TechSup	CompRes	Advertising	ProdLine	SalesFImage	ComPricing	WartyClaim	OrdBilling	DelSpeed	Satisfaction
0	1	8.5	3.9	2.5	5.9	4.8	4.9	6.0	6.8	4.7	5.0	3.7	8.2
1	2	8.2	2.7	5.1	7.2	3.4	7.9	3.1	5.3	5.5	3.9	4.9	5.7
2	3	9.2	3.4	5.6	5.6	5.4	7.4	5.8	4.5	6.2	5.4	4.5	8.9
3	4	6.4	3.3	7.0	3.7	4.7	4.7	4.5	8.8	7.0	4.3	3.0	4.8
4	5	9.0	3.4	5.2	4.6	2.2	6.0	4.5	6.8	6.1	4.5	3.5	7.1

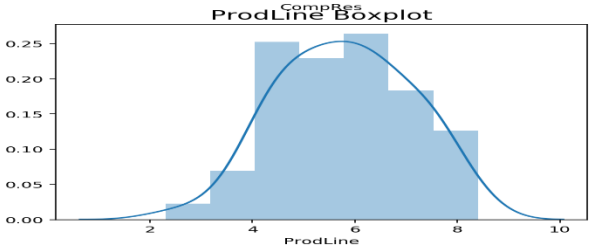
Univariate Analysis

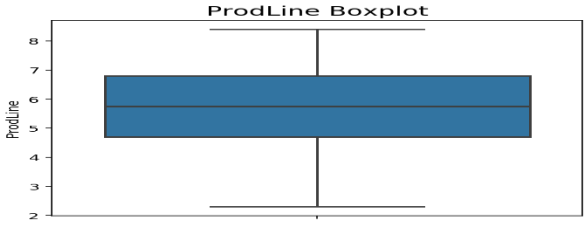




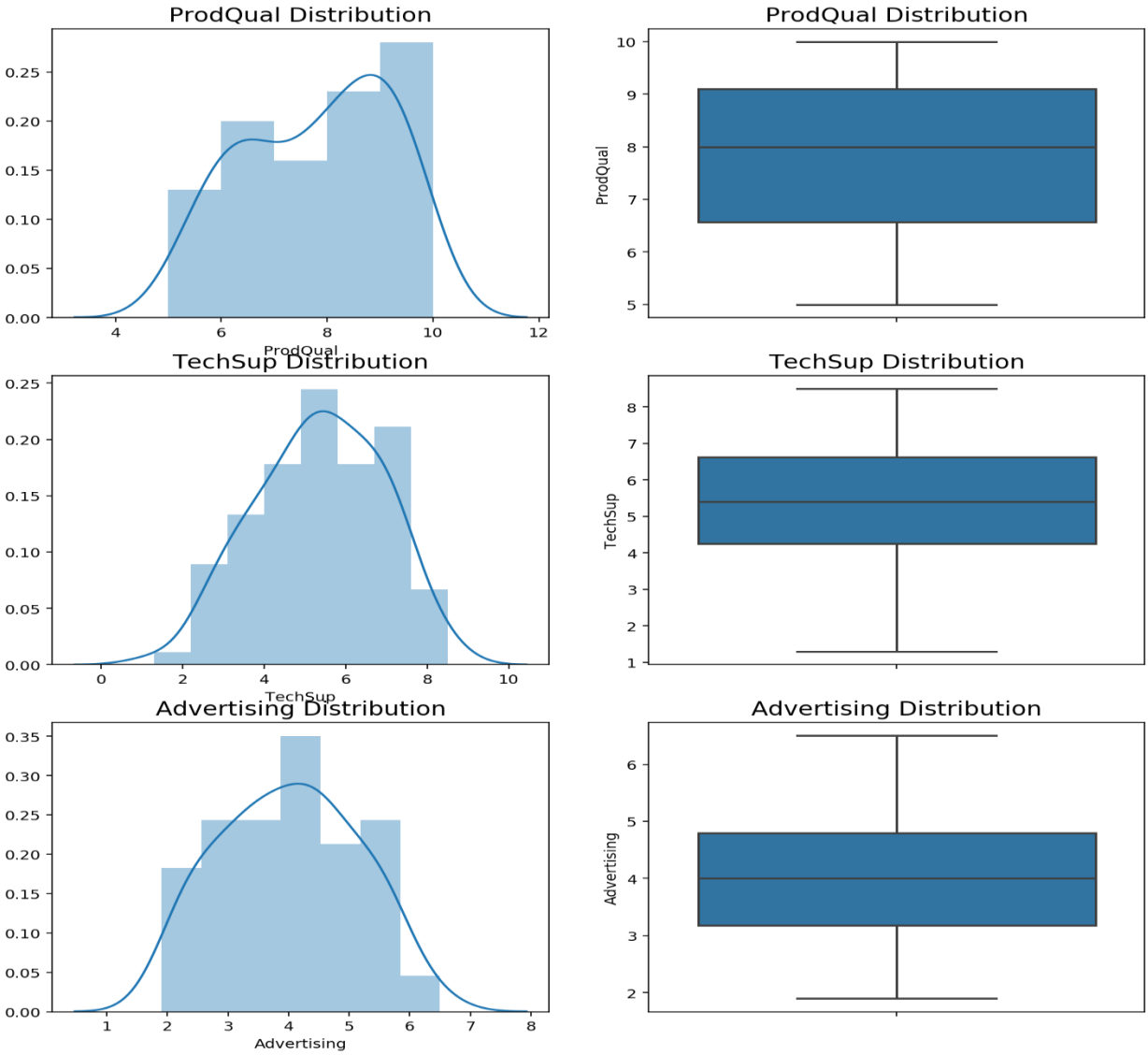






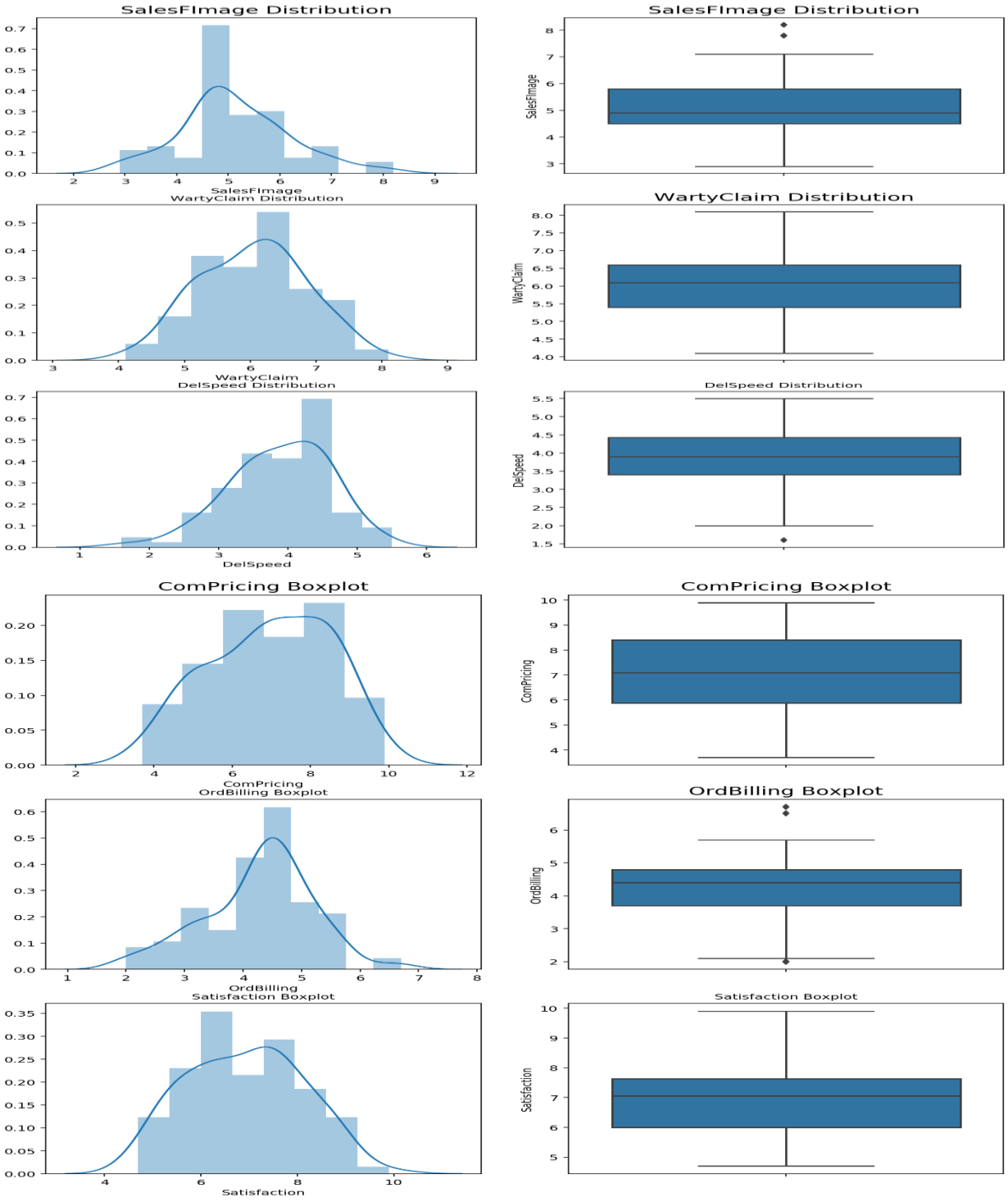


Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



It can be seen from the above two graphs that that ProdLine,CompRes,Advertising,TechSup follows normal distribution.ProdQual is normal with a very little skew. Ecom has some outliers present in the dataset and the distribution is normal.

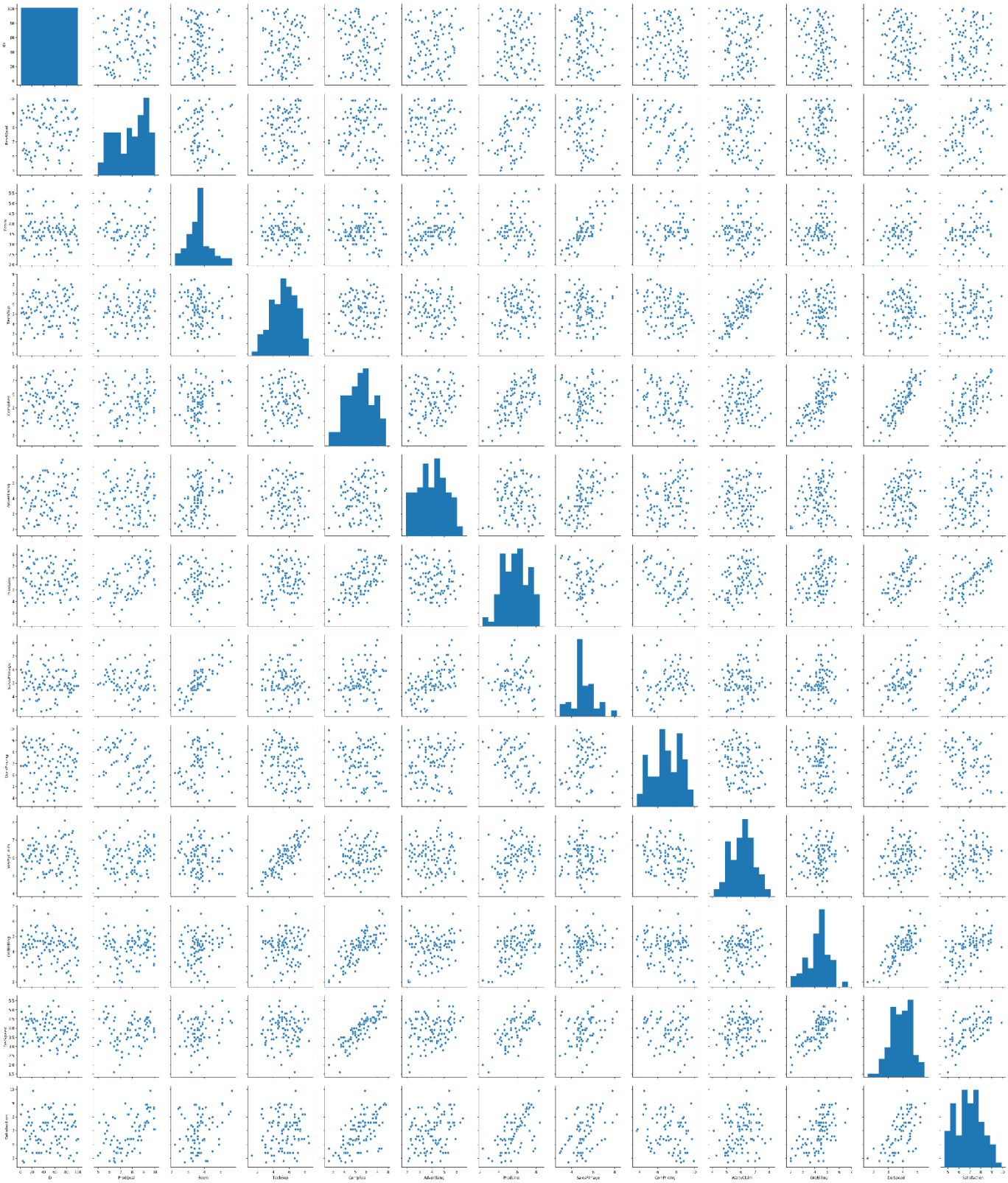
Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



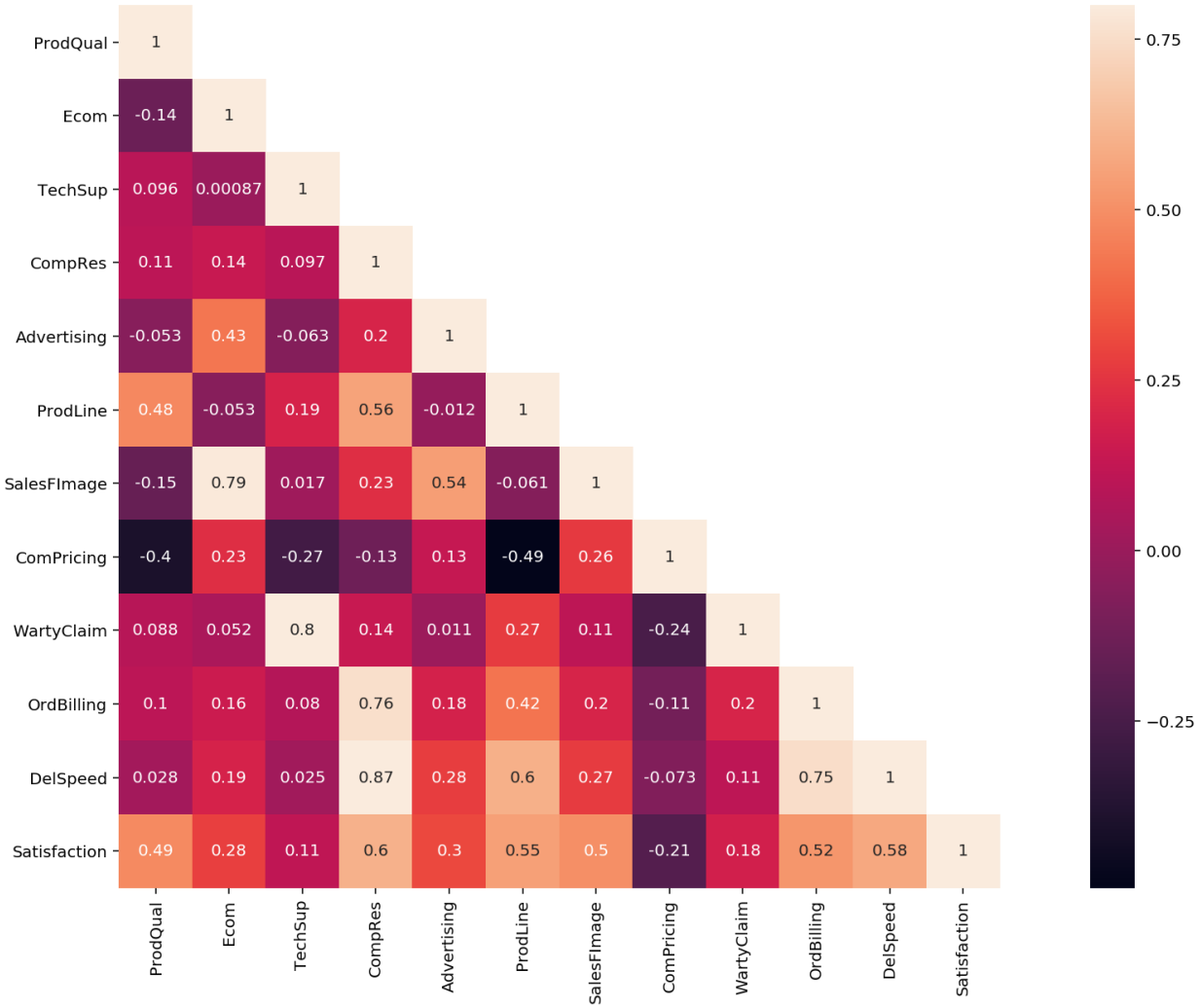
OrdBilling has some outliers.DelSpeed(with one or two outliers) and Satisfaction is slightly left skewed.WartyClaim,SalesFImage & ComPricing are normally distributed.

Bi-variate Analysis:

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

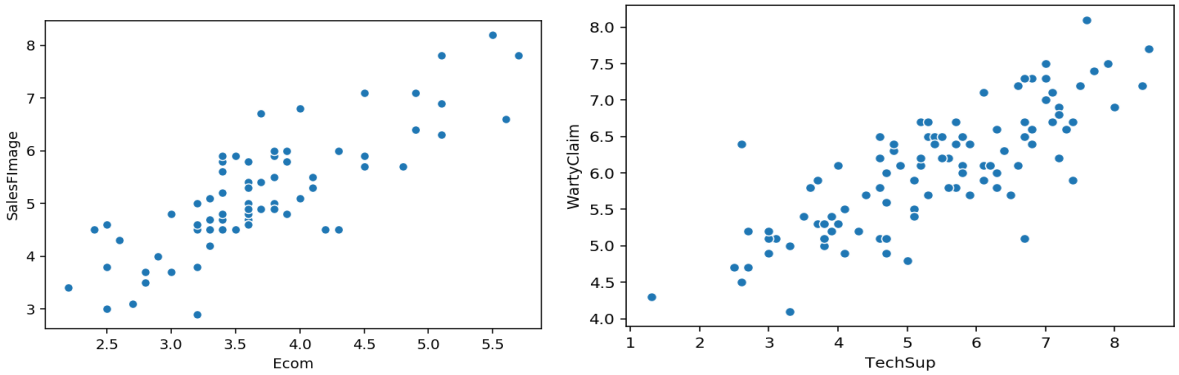


Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

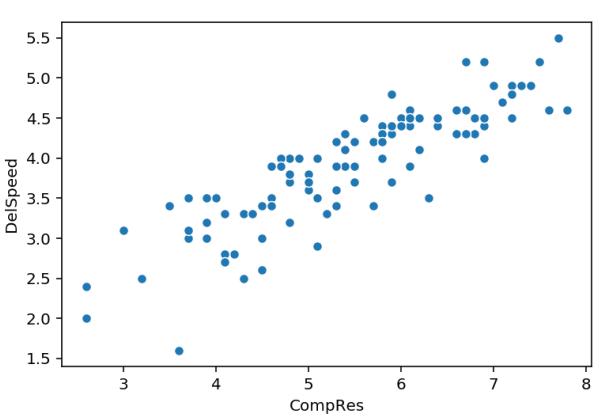


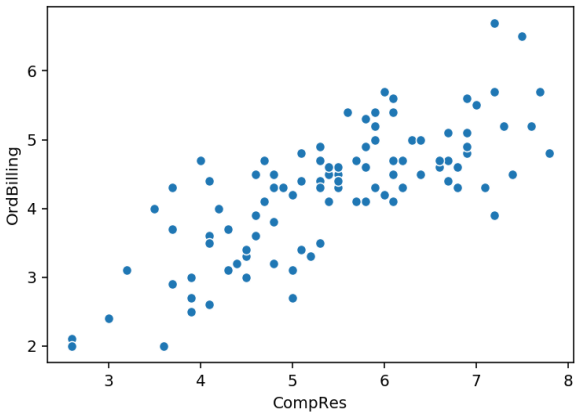
From the above correlation plot, we can see that Ecom and SalesFImage, TEchSup and WartyClaim, CompRes and DelSpeed, OrdBilling and DelSpeed,CompRes and OrdBilling are highly positive correlated.

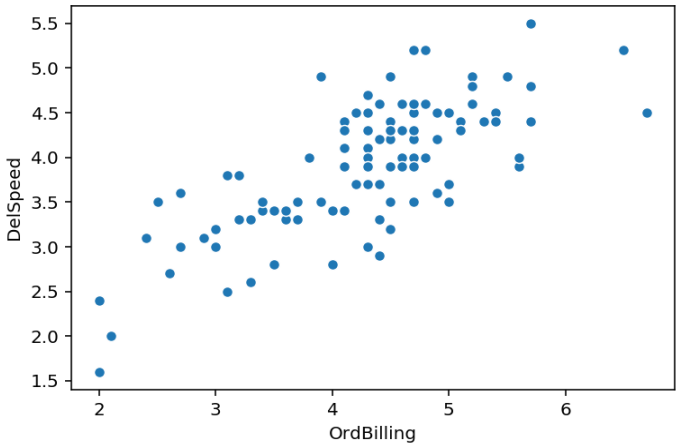
If we draw a scatter plot between the items the same can be observed.



Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.







2) Scale the variables and write the inference for using the type of scaling function for this case study.

PCA is affected by scale so you need to scale the features in your data before applying PCA. Use **StandardScaler** to help you standardize the dataset's features onto unit scale (mean = 0 and variance = 1) which is a requirement for the optimal performance of many machine learning algorithms.

The standard score of a sample x is calculated as:

$z = (x - \mu) / s$

where μ is the mean of the training samples and s is the standard deviation of the training sample. Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using transform.

Result after scaling:

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

	ProdQual	Ecom	TechSup	CompRes	Advertising	ProdLine	SalesFImage	ComPricing	WartyClaim	OrdBilling	DelSpeed	Satisfaction
0	0.496660	0.327114	-1.881421	0.380922	0.704543	-0.691530	0.821973	-0.113185	-1.646582	0.781230	-0.254531	1.081067
1	0.280721	-1.394538	-0.174023	1.462141	-0.544014	1.600835	-1.896068	-1.088915	-0.665744	-0.409009	1.387605	-1.027098
2	1.000518	-0.390241	0.154322	0.131410	1.239639	1.218774	0.634522	-1.609304	0.192489	1.214044	0.840226	1.671354
3	-1.014914	-0.533712	1.073690	-1.448834	0.615361	-0.844354	-0.583910	1.187789	1.173327	0.023805	-1.212443	-1.786038
4	0.856559	-0.390241	-0.108354	-0.700298	-1.614207	0.149004	-0.583910	-0.113185	0.069885	0.240212	-0.528220	0.153474

3) Comment on the comparison between covariance and the correlation matrix after scaling.

Covariance matrix:

```
x.cov()
```

	ProdQual	Ecom	TechSup	CompRes	Advertising	ProdLine	SalesFImage	ComPricing	WartyClaim	OrdBilling	DelSpeed	Sa
ProdQual	1.010101	-0.138549	0.096566	0.107444	-0.054013	0.482317	-0.153346	-0.405335	0.089204	0.105357	0.027998	0.4
Ecom	-0.138549	1.010101	0.000876	0.141595	0.434233	-0.053220	0.799539	0.231780	0.052422	0.157725	0.193572	0.2
TechSup	0.096566	0.000876	1.010101	0.097633	-0.063505	0.194571	0.017162	-0.273522	0.805220	0.080911	0.025698	0.1
CompRes	0.107444	0.141595	0.097633	1.010101	0.198906	0.567088	0.232072	-0.129247	0.141827	0.764514	0.873830	0.6
Advertising	-0.054013	0.434233	-0.063505	0.198906	1.010101	-0.011667	0.547680	0.135573	0.010901	0.186097	0.278650	0.3
ProdLine	0.482317	-0.053220	0.194571	0.567088	-0.011667	1.010101	-0.061935	-0.499948	0.275836	0.428695	0.607930	0.5
SalesFImage	-0.153346	0.799539	0.017162	0.232072	0.547680	-0.061935	1.010101	0.267269	0.108541	0.197098	0.274294	0.5
ComPricing	-0.405335	0.231780	-0.273522	-0.129247	0.135573	-0.499948	0.267269	1.010101	-0.247461	-0.115724	-0.073608	-0.
WartyClaim	0.089204	0.052422	0.805220	0.141827	0.010901	0.275836	0.108541	-0.247461	1.010101	0.199056	0.110500	0.1
OrdBilling	0.105357	0.157725	0.080911	0.764514	0.186097	0.428695	0.197098	-0.115724	0.199056	1.010101	0.758589	0.5
DelSpeed	0.027998	0.193572	0.025698	0.873830	0.278650	0.607930	0.274294	-0.073608	0.110500	0.758589	1.010101	0.5
Satisfaction	0.491237	0.285601	0.113735	0.609356	0.307747	0.556107	0.505258	-0.210400	0.179338	0.527002	0.582871	1.0

Correlation matrix:

```
x.corr()
```

	ProdQual	Ecom	TechSup	CompRes	Advertising	ProdLine	SalesFImage	ComPricing	WartyClaim	OrdBilling	DelSpeed	Sa
ProdQual	1.000000	-0.137163	0.095600	0.106370	-0.053473	0.477493	-0.151813	-0.401282	0.088312	0.104303	0.027718	0.4
Ecom	-0.137163	1.000000	0.000867	0.140179	0.429891	-0.052688	0.791544	0.229462	0.051898	0.156147	0.191636	0.2
TechSup	0.095600	0.000867	1.000000	0.096657	-0.062870	0.192625	0.016991	-0.270787	0.797168	0.080102	0.025441	0.1
CompRes	0.106370	0.140179	0.096657	1.000000	0.196917	0.561417	0.229752	-0.127954	0.140408	0.756869	0.865092	0.6
Advertising	-0.053473	0.429891	-0.062870	0.196917	1.000000	-0.011551	0.542204	0.134217	0.010792	0.184236	0.275863	0.3
ProdLine	0.477493	-0.052688	0.192625	0.561417	-0.011551	1.000000	-0.061316	-0.494948	0.273078	0.424408	0.601850	0.5
SalesFImage	-0.151813	0.791544	0.016991	0.229752	0.542204	-0.061316	1.000000	0.264597	0.107455	0.195127	0.271551	0.5
ComPricing	-0.401282	0.229462	-0.270787	-0.127954	0.134217	-0.494948	0.264597	1.000000	-0.244986	-0.114567	-0.072872	-0.
WartyClaim	0.088312	0.051898	0.797168	0.140408	0.010792	0.273078	0.107455	-0.244986	1.000000	0.197065	0.109395	0.1
OrdBilling	0.104303	0.156147	0.080102	0.756869	0.184236	0.424408	0.195127	-0.114567	0.197065	1.000000	0.751003	0.5
DelSpeed	0.027718	0.191636	0.025441	0.865092	0.275863	0.601850	0.271551	-0.072872	0.109395	0.751003	1.000000	0.5
Satisfaction	0.486325	0.282745	0.112597	0.603263	0.304669	0.550546	0.500205	-0.208296	0.177545	0.521732	0.577042	1.0

```
# Now without Scaling lets check out correlation matrix
df_corr = df.copy()
df_corr = df_corr.drop(["ID"],axis=1)
df_corr.corr()
```

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

	ProdQual	Ecom	TechSup	CompRes	Advertising	ProdLine	SalesFImage	ComPricing	WartyClaim	OrdBilling	DelSpeed
ProdQual	1.000000	-0.137163	0.095600	0.106370	-0.053473	0.477493	-0.151813	-0.401282	0.088312	0.104303	0.027718
Ecom	-0.137163	1.000000	0.000867	0.140179	0.429891	-0.052688	0.791544	0.229462	0.051898	0.156147	0.191636
TechSup	0.095600	0.000867	1.000000	0.096657	-0.062870	0.192625	0.016991	-0.270787	0.797168	0.080102	0.025441
CompRes	0.106370	0.140179	0.096657	1.000000	0.196917	0.561417	0.229752	-0.127954	0.140408	0.756869	0.865092
Advertising	-0.053473	0.429891	-0.062870	0.196917	1.000000	-0.011551	0.542204	0.134217	0.010792	0.184236	0.275863
ProdLine	0.477493	-0.052688	0.192625	0.561417	-0.011551	1.000000	-0.061316	-0.494948	0.273078	0.424408	0.601850
SalesFImage	-0.151813	0.791544	0.016991	0.229752	0.542204	-0.061316	1.000000	0.264597	0.107455	0.195127	0.271551
ComPricing	-0.401282	0.229462	-0.270787	-0.127954	0.134217	-0.494948	0.264597	1.000000	-0.244986	-0.114567	-0.072872
WartyClaim	0.088312	0.051898	0.797168	0.140408	0.010792	0.273078	0.107455	-0.244986	1.000000	0.197065	0.109395
OrdBilling	0.104303	0.156147	0.080102	0.756869	0.184236	0.424408	0.195127	-0.114567	0.197065	1.000000	0.751003
DelSpeed	0.027718	0.191636	0.025441	0.865092	0.275863	0.601850	0.271551	-0.072872	0.109395	0.751003	1.000000
Satisfaction	0.486325	0.282745	0.112597	0.603263	0.304669	0.550546	0.500205	-0.208296	0.177545	0.521732	0.577042

"**Covariance**" indicates the direction of the linear relationship between variables. "Correlation" on the other hand measures both the strength and direction of the linear relationship between two variables. Correlation is a function of the covariance. You can obtain the correlation coefficient of two variables by dividing the covariance of these variables by the product of the standard deviations of the same values.
We can state that above three approaches yield the same eigenvectors and eigenvalue pairs:

1.Eigen decomposition of the covariance matrix after standardizing the data.

2.Eigen decomposition of the correlation matrix.

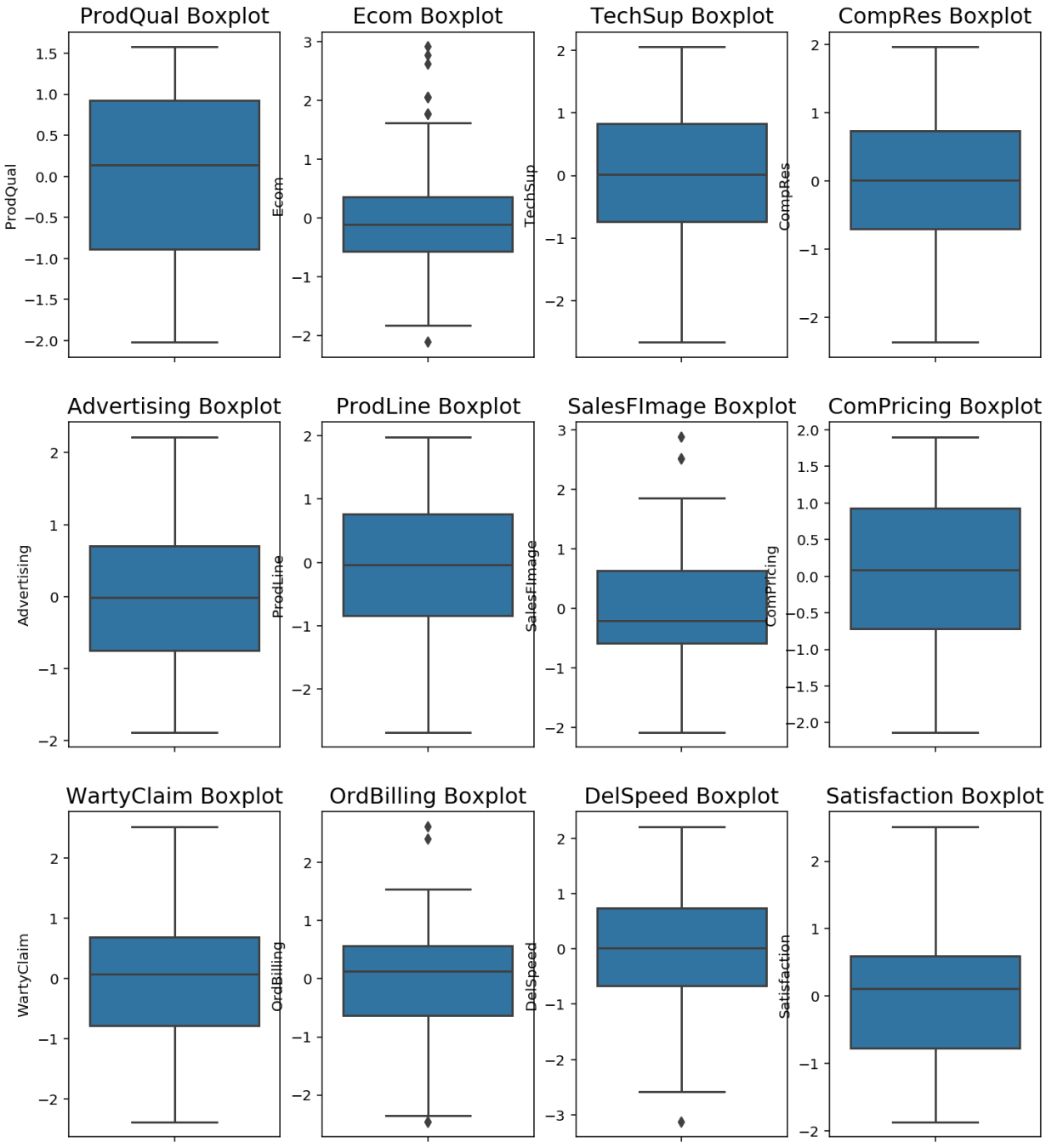
3.Eigen decomposition of the correlation matrix after standardizing the data.

Finally, we can say that after scaling - the covariance and the correlation have the same values.

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

4) Check the dataset for outliers before and after scaling. Draw your inferences from this exercise

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



From the boxplot we can see that there are not much of effect on outliers before and after scaling.

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

5) Build the covariance matrix, eigenvalues and eigenvector.

Covariance matrix

```
##Building co-variance matrix
cov = x.cov()
cov
```

	ProdQual	Ecom	TechSup	CompRes	Advertising	ProdLine	SalesFImage	ComPricing	WartyClaim	OrdBilling	DelSpeed
ProdQual	1.010101	-0.138549	0.096566	0.107444	-0.054013	0.482317	-0.153346	-0.405335	0.089204	0.105357	0.027998
Ecom	-0.138549	1.010101	0.000876	0.141595	0.434233	-0.053220	0.799539	0.231780	0.052422	0.157725	0.193572
TechSup	0.096566	0.000876	1.010101	0.097633	-0.063505	0.194571	0.017162	-0.273522	0.805220	0.080911	0.025698
CompRes	0.107444	0.141595	0.097633	1.010101	0.198906	0.567088	0.232072	-0.129247	0.141827	0.764514	0.873830
Advertising	-0.054013	0.434233	-0.063505	0.198906	1.010101	-0.011667	0.547680	0.135573	0.010901	0.186097	0.278650
ProdLine	0.482317	-0.053220	0.194571	0.567088	-0.011667	1.010101	-0.061935	-0.499948	0.275836	0.428695	0.607930
SalesFImage	-0.153346	0.799539	0.017162	0.232072	0.547680	-0.061935	1.010101	0.267269	0.108541	0.197098	0.274294
ComPricing	-0.405335	0.231780	-0.273522	-0.129247	0.135573	-0.499948	0.267269	1.010101	-0.247461	-0.115724	-0.073608
WartyClaim	0.089204	0.052422	0.805220	0.141827	0.010901	0.275836	0.108541	-0.247461	1.010101	0.199056	0.110500
OrdBilling	0.105357	0.157725	0.080911	0.764514	0.186097	0.428695	0.197098	-0.115724	0.199056	1.010101	0.758589
DelSpeed	0.027998	0.193572	0.025698	0.873830	0.278650	0.607930	0.274294	-0.073608	0.110500	0.758589	1.010101
Satisfaction	0.491237	0.285601	0.113735	0.609356	0.307747	0.556107	0.505258	-0.210400	0.179338	0.527002	0.582871

Eigen values:

```
eigvals
```

```
array([4.08369694, 2.57871152, 1.70931735, 1.22984483, 0.6423868 ,
       0.57427406, 0.40689671, 0.32775774, 0.23852472, 0.14568036,
       0.08398124, 0.10013985])
```

There are only 4 values where the eigvals are greater than 1

Eigen vectors:

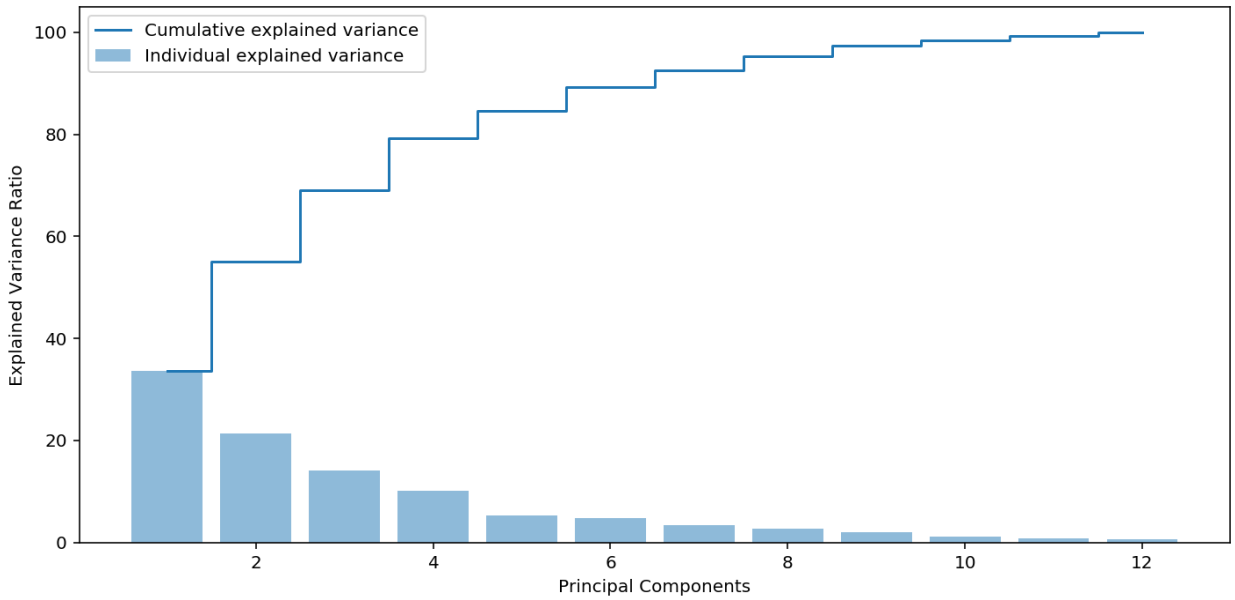
```
eigvecs
```

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
array([[-0.15855116,  0.31313152,  0.07356137,  0.61407082,  0.24964531,
        -0.36499541,  0.12640774,  0.32687751, -0.18602426,  0.2037033 ,
         0.21787575,  0.22885317],
       [-0.1661857 , -0.44059261, -0.23651951,  0.19628244,  0.18886909,
         0.46540483,  0.00824784,  0.50785197, -0.21574952,  0.03718659,
        -0.35323725, -0.02881148],
       [-0.12514332,  0.23828985, -0.61631236, -0.17941402,  0.03977108,
        -0.12392836, -0.01346077, -0.08182818, -0.54753081, -0.42475155,
         0.10580091, -0.01766533],
       [-0.42263337, -0.00134121,  0.19665426, -0.27970497,  0.03340857,
        -0.01495235, -0.00463818, -0.14929932, -0.43697539,  0.58601845,
         0.05627641, -0.37853377],
       [-0.1807615 , -0.35724531, -0.0898675 ,  0.20600014, -0.76107633,
        -0.4189084 , -0.07155058,  0.12282896, -0.04176506, -0.02836138,
        -0.04824083, -0.0968768 ],
       [-0.35283874,  0.29778667,  0.11122737,  0.10008828, -0.0250607 ,
         0.1958228 , -0.63397913,  0.22319134,  0.23246141, -0.25391841,
         0.18600871, -0.34728677],
       [-0.21794995, -0.46488879, -0.2409419 ,  0.19948826,  0.14209236,
         0.16711795,  0.02165026, -0.33410983,  0.1703657 ,  0.03993494,
         0.66500583,  0.07388433],
       [ 0.13483701, -0.41776317,  0.0516667 , -0.24079483,  0.4896484 ,
        -0.58557549, -0.34280528,  0.16338764,  0.02851369, -0.08642644,
        -0.01139137, -0.10660117],
       [-0.17499123,  0.2011842 , -0.60545958, -0.18959933,  0.02158615,
        -0.1422959 , -0.04011921,  0.10701582,  0.50449856,  0.45392345,
        -0.15868264,  0.0827785 ],
       [-0.38797945, -0.00906156,  0.15503653, -0.30668573,  0.04908379,
        -0.09117472,  0.62874224,  0.33498375,  0.25197455, -0.32105097,
         0.14716762, -0.15754746],
       [-0.4223407 , -0.05445737,  0.21799023, -0.28990302, -0.06222027,
         0.03060577, -0.23692774,  0.00146402, -0.07544805, -0.05793177,
        -0.06069937,  0.78321219],
       [-0.41302455, -0.02390379,  0.02873859,  0.33118987,  0.22967423,
        -0.14296626,  0.07520655, -0.52854183,  0.13706617, -0.21557147,
        -0.53252314, -0.10623326]])
```

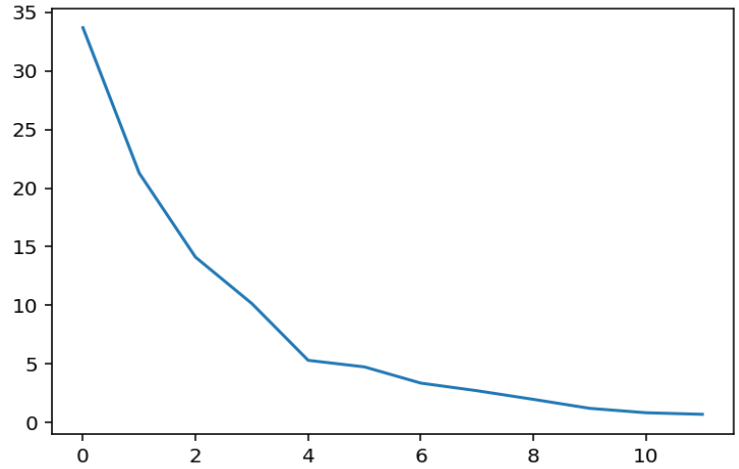
Cummulative Explained variance:

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



Scree plot

```
#Scree plot
plt.plot(var_exp)
```



From Scree plot, we observe that there is a steep drop after 4th PC component. As we can see that the total number of eigenvalues greater than 1 is 4 and from the scree plot - we see a steep drop after 4th PC component. We can conclude that 4 PC components give us the max variance of the dataset that is ~ **80%.**

6) Discuss the cumulative values of the eigenvalues. How does it help you to decide on the optimum number of principal components? What do the eigenvectors indicate? Perform PCA and export the data of the Principal Component scores into a data frame.

```
#The number of PC components in df is always equal to number of the column
s available(Here 12 columns are considered)
from sklearn.decomposition import PCA
```

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
pca = PCA(n_components=12)
pc = pca.fit_transform(x)
PCAdf = pd.DataFrame(data = pc, columns = ['pc1', 'pc2','pc3','pc4','pc5',
'pc6','pc7',
                                           'pc8','pc9','pc10','pc11','pc12
'])
PCAdf.head()
```

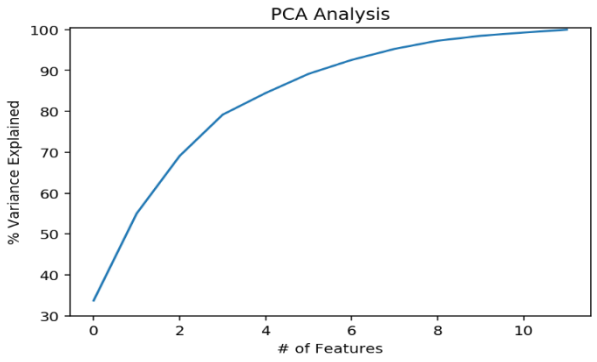
	pc1	pc2	pc3	pc4	pc5	pc6	pc7	pc8	pc9	pc10	pc11	pc12
0	-0.490389	1.580229	1.943132	1.371775	0.066902	0.027757	1.232500	0.421646	0.180880	0.117379	-0.335596	-0.016342
1	-0.495644	-2.485075	1.985439	-1.037741	0.936743	-0.431697	-1.260651	-0.262177	-0.938607	0.533566	0.232000	-0.201334
2	-2.727909	-0.761250	0.187628	1.265916	1.089905	0.151429	0.506831	0.408908	0.618624	-0.690811	0.168011	-0.193841
3	2.236864	0.176334	-1.864573	-1.345408	0.616691	1.156490	0.051858	-0.877704	0.506194	-0.480498	-0.290054	-0.252905
4	0.644061	-1.392029	0.265044	0.374907	-1.280251	-0.133765	0.441874	-0.212862	0.446216	-0.257864	0.085753	0.033235

```
var=np.cumsum(np.round(pca.explained_variance_ratio_, decimals=3)*100)
var #cumulative sum of variance explained with [n] features
```

```
array([ 33.7,  55. ,  69.1,  79.2,  84.5,  89.2,  92.6,  95.3,  97.3,
        98.5,  99.3, 100. ])
```

The Cumulative **%** gives the percentage of variance accounted for by the n components. For example, the cumulative percentage for the second component is the sum of the percentage of variance for the first and second components. It helps in deciding the number of components by selecting the components which explained the high variance

In the above array we see that the first feature explains 33.7% of the variance within our data set while the first two explain 55% and so on. If we employ 4 features we capture **~ 80%** of the variance within the dataset, thus we gain very little by implementing an additional feature (think of this as diminishing marginal return on total variance explained).



7) Write the explicit form of the first PC (in terms of Eigen Vectors)

```
eigvecs[0]
```

```
array([-0.15855116,  0.31313152,  0.07356137,  0.61407082,  0.24964531,
       -0.36499541,  0.12640774,  0.32687751, -0.18602426,  0.2037033 ,
        0.21787575,  0.22885317])
```

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

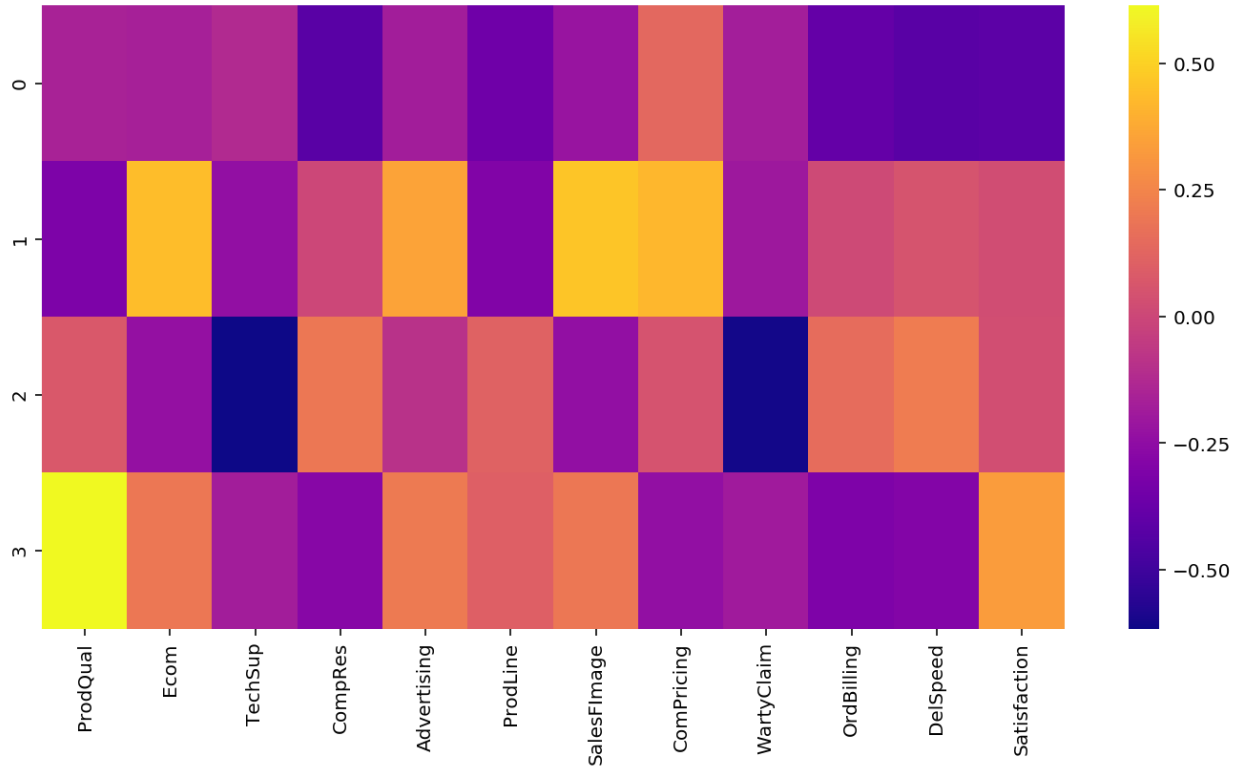
PC1 : -0.15855116*ProdQual*
*+0.31313152*Ecom+0.07356137*TechSup+0.614070820*CompRes+0.24964531*Advertising* -
*0.36499541*ProdLine+0.12640774*SalesFImage+0.32687751*ComPricing -
0.18602426*WartyClaim+0.2037033*OrdBilling+0.21787575*DelSpeed+0.22885317*Satisfaction*.

8) Mention the business implication of using the Principal Component Analysis for this case study.

We find that only 4 PC components gives us a variance of about ~**80**% , the number of PC components can also be found out based on Eigen values and scree plot.

Business Implications:

Correlation between the components and features:



The heatmap and the colour bar basically represent the correlation between the *various features* and the *principal component* itself

In the above case study it is observed that all the total number of columns has been reduced from 1**2 to 4** and **the multi-collinearity** has also been **removed**.

PCA is a statistical technique and uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables. PCA also is a tool to reduce multidimensional data to lower dimensions while retaining most of the information. Principal Component Analysis (PCA) is a well-established mathematical technique for reducing the

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

dimensionality of data, while keeping as much variation as possible. This PCA can only be done on continuous variables.

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.