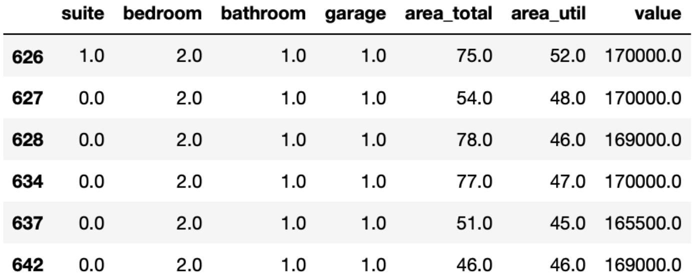
Principle Factor Analysis

- To predict a selling price (value) of real estate, I choose these numerical columns to be a feature.

1.Number of suite (suite)
2.Number of bedroom (bedroom)
3.Number of bathroom (bathroom)
4.Number of garage (garage)
5.Amount of total area (area_total)
6.Amount of useful area (area_util)

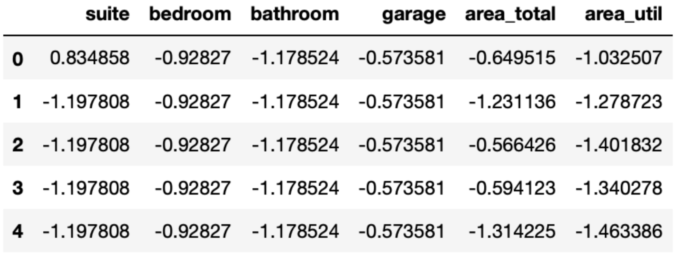


	suite	bedroom	bathroom	garage	area_total	area_util	value
626	1.0	2.0	1.0	1.0	75.0	52.0	170000.0
627	0.0	2.0	1.0	1.0	54.0	48.0	170000.0
628	0.0	2.0	1.0	1.0	78.0	46.0	169000.0
634	0.0	2.0	1.0	1.0	77.0	47.0	170000.0
637	0.0	2.0	1.0	1.0	51.0	45.0	165500.0
642	0.0	2.0	1.0	1.0	46.0	46.0	169000.0

Data frame of principal factor analysis

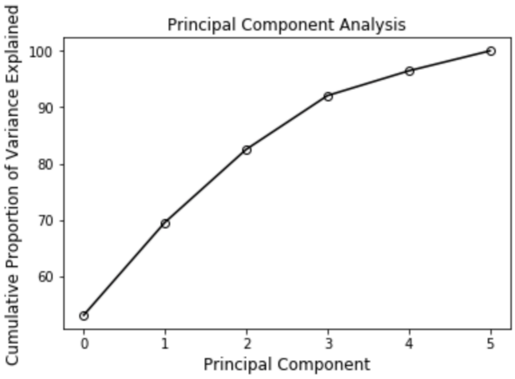
- **Reduce Dimension with Principal Component Analysis (Python)**

1. Normalize the data



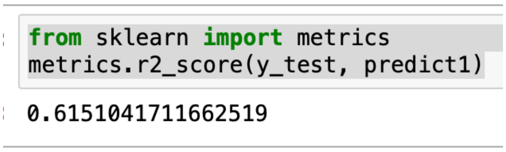
	suite	bedroom	bathroom	garage	area_total	area_util
0	0.834858	-0.92827	-1.178524	-0.573581	-0.649515	-1.032507
1	-1.197808	-0.92827	-1.178524	-0.573581	-1.231136	-1.278723
2	-1.197808	-0.92827	-1.178524	-0.573581	-0.566426	-1.401832
3	-1.197808	-0.92827	-1.178524	-0.573581	-0.594123	-1.340278
4	-1.197808	-0.92827	-1.178524	-0.573581	-1.314225	-1.463386

2. Training PCA with 6 components in first round to see the variation
    - Plot Cumulative Proportion of Variance Explained Graph to explain each component which I need to decide the number of components in the next round.



- From the output I found that the first 3 components can explain 90% of the variance.

3. Training PCA again with 3 components and calculate R2 score to see how our model fit with the data.

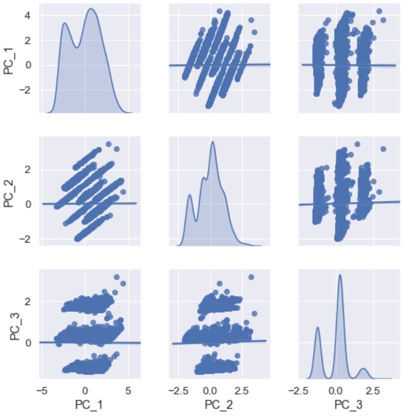


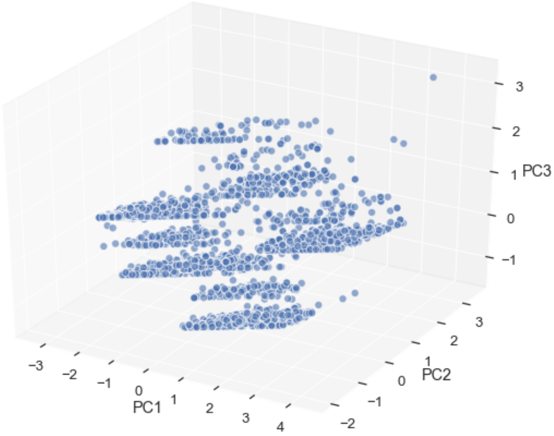
```
from sklearn import metrics
metrics.r2_score(y_test, predict1)
```

```
0.6151041711662519
```

- The accuracy comes out with R2 score : 0.615

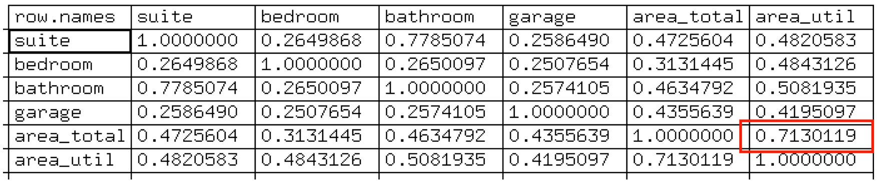
4. Visualization with dimensionality reducers : Principle Component Analysis





- **Reduce Dimension with Principal Factor Analysis (R)**

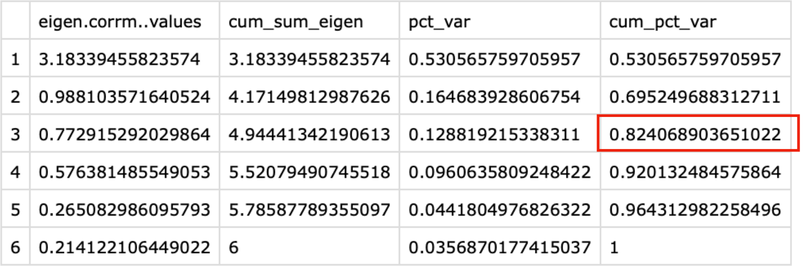
1. Create correlation matrix for the data.
   - Our data have highly correlated to each variable such as area_total and area_util. and it can will the cause of multicollinearity when we do the regression.



row.names	suite	bedroom	bathroom	garage	area_total	area_util
suite	1.0000000	0.2649868	0.7785074	0.2586490	0.4725604	0.4820583
bedroom	0.2649868	1.0000000	0.2650097	0.2507654	0.3131445	0.4843126
bathroom	0.7785074	0.2650097	1.0000000	0.2574105	0.4634792	0.5081935
garage	0.2586490	0.2507654	0.2574105	1.0000000	0.4355639	0.4195097
area_total	0.4725604	0.3131445	0.4634792	0.4355639	1.0000000	0.7130119
area_util	0.4820583	0.4843126	0.5081935	0.4195097	0.7130119	1.0000000

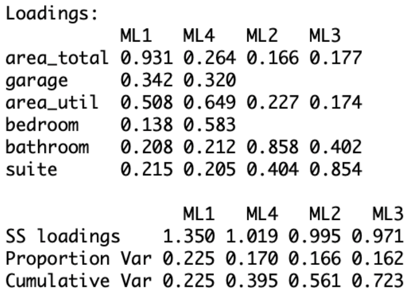
2. Finding Eigen Values
   - To find the number of factors which can use to correctly group of features, I use a- cumulative eigenvalue percentage variance and cumulative percentage variance.

   - After see the table in a cumulative percentage variance(cum_pct_var) column it- clearly that the first three factors explain approximately 82% of the variance.



	eigen.corrm..values	cum_sum_eigen	pct_var	cum_pct_var
1	3.18339455823574	3.18339455823574	0.530565759705957	0.530565759705957
2	0.988103571640524	4.17149812987626	0.164683928606754	0.695249688312711
3	0.772915292029864	4.94441342190613	0.128819215338311	0.824068903651022
4	0.576381485549053	5.52079490745518	0.0960635809248422	0.920132484575864
5	0.265082986095793	5.78587789355097	0.0441804976826322	0.964312982258496
6	0.214122106449022	6	0.0356870177415037	1

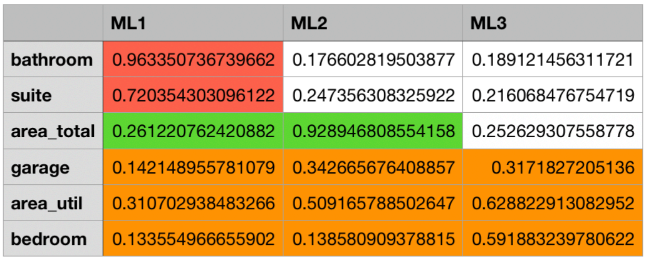
3. Reduce variable using factor analysis
    - Complies FA function to analysis each of factors.



```
Loadings:
           ML1   ML4   ML2   ML3  
area_total 0.931 0.264 0.166 0.177
garage     0.342 0.320            
area_util  0.508 0.649 0.227 0.174
bedroom    0.138 0.583            
bathroom   0.208 0.212 0.858 0.402
suite      0.215 0.205 0.404 0.854

                 ML1   ML4   ML2   ML3
SS loadings    1.350 1.019 0.995 0.971
Proportion Var 0.225 0.170 0.166 0.162
Cumulative Var 0.225 0.395 0.561 0.723
```

    - Grouping Variables

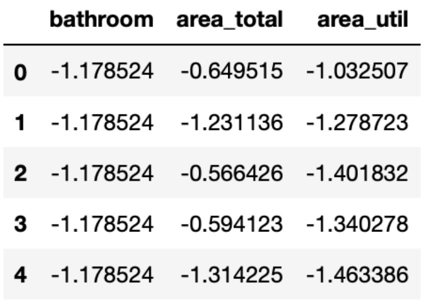


	ML1	ML2	ML3
bathroom	0.963350736739662	0.176602819503877	0.189121456311721
suite	0.720354303096122	0.247356308325922	0.216068476754719
area_total	0.261220762420882	0.928946808554158	0.252629307558778
garage	0.142148955781079	0.342665676408857	0.3171827205136
area_util	0.310702938483266	0.509165788502647	0.628822913082952
bedroom	0.133554966655902	0.138580909378815	0.591883239780622

    - The variable which can grouping will have a nearly variance for each other, and now I-can should one of them to represent another. To reduce dimension in the group 1 (red), I choose bathroom, a second group (green), and the last group (orange), I choose area-total to my features.

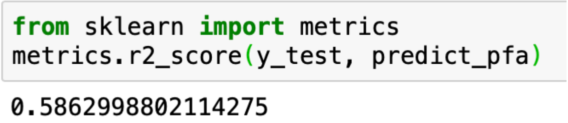
- **Compare R2 score between PCA and PFA (Python)**

1. Choose column with our new feature that I just reduce with PFA above.



	bathroom	area_total	area_util
0	-1.178524	-0.649515	-1.032507
1	-1.178524	-1.231136	-1.278723
2	-1.178524	-0.566426	-1.401832
3	-1.178524	-0.594123	-1.340278
4	-1.178524	-1.314225	-1.463386

2. Train regression model with new data.
    - I use train test split from Sklearn and set random seed same as when I train PCA.

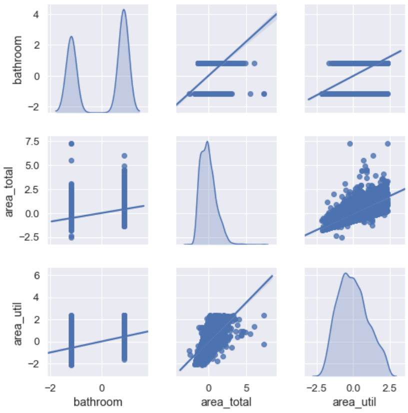


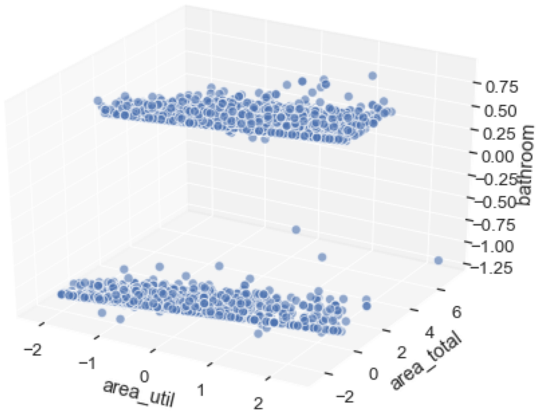
```
from sklearn import metrics
metrics.r2_score(y_test, predict_pfa)
```

```
0.5862998802114275
```

    - The R2 come out with 0.58.

3. Visualization with dimensionality reducers : Principle Factor Analysis





- **Conclusion**
    - The R2 score which come out from both technique are nearly but actually I think this score is less than I expected. And maybe the reason why the result come out with low score is I have a discrete feature which is a bathroom feature in the principle factor analysis part.