In [5]:
```python
import pandas as pd

# Load the dataset
dataset_path = "C:\\Users\\nihar\\OneDrive\\Desktop\\Bootcamp\\SCMA 632\\DataSet\\Survey.csv"
survey_data = pd.read_csv(dataset_path)

# Inspect the dataset
print(survey_data.info())
print(survey_data.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70 entries, 0 to 69
Data columns (total 50 columns):
 #   Column                                   Non-Null Count  Dtype
---  ------                                   --------------  -----
 0   City                                     70 non-null     object
 1   Sex                                      70 non-null     object
 2   Age                                      70 non-null     object
 3   Occupation                               70 non-null     object
 4   Monthly Household Income                 70 non-null     object
 5   Income                                   70 non-null     int64
 6   Planning to Buy a new house              70 non-null     object
 7   Time Frame                               70 non-null     object
 8   Reasons for buying a house               70 non-null     object
 9   what type of House                       70 non-null     object
 10  Number of rooms                          70 non-null     object
 11  Size of House                            70 non-null     object
 12  Budget                                   70 non-null     object
 13  Finished/Semi Finished                   70 non-null     object
 14  Influence Decision                       70 non-null     object
 15  Maintainance                             70 non-null     object
 16  EMI                                      70 non-null     object
 17  1.Proximity to city                      70 non-null     int64
 18  2.Proximity to schools                   70 non-null     int64
 19  3. Proximity to transport                70 non-null     int64
 20  4. Proximity to work place               70 non-null     int64
 21  5. Proximity to shopping                 70 non-null     int64
 22  1. Gym/Pool/Sports facility              70 non-null     int64
 23  2. Parking space                         70 non-null     int64
 24  3.Power back-up                          70 non-null     int64
 25  4.Water supply                           70 non-null     int64
 26  5.Security                               70 non-null     int64
 27  1. Exterior look                         70 non-null     int64
 28  2. Unit size                             70 non-null     int64
 29  3. Interior design and branded components  70 non-null     int64
 30  4. Layout plan (Integrated etc.)         70 non-null     int64
 31  5. View from apartment                   70 non-null     int64
 32  1. Price                                 70 non-null     int64
 33  2. Booking amount                        70 non-null     int64
 34  3. Equated Monthly Instalment (EMI)      70 non-null     int64
 35  4. Maintenance charges                   70 non-null     int64
 36  5. Availability of loan                  70 non-null     int64
 37  1. Builder reputation                    70 non-null     int64
 38  2. Appreciation potential                70 non-null     int64
 39  3. Profile of neighbourhood              70 non-null     int64
 40  4. Availability of domestic help         70 non-null     int64
 41  Time                                     70 non-null     int64
 42  Size                                     70 non-null     int64
 43  Budgets                                  70 non-null     float64
 44  Maintainances                            70 non-null     int64
 45  EMI.1                                    70 non-null     int64
 46  ages                                     70 non-null     float64
 47  sex                                      70 non-null     object
 48  Finished/Semi Finished.1                 70 non-null     object
 49  Influence Decision.1                     70 non-null     object
dtypes: float64(2), int64(29), object(19)
memory usage: 27.5+ KB
None
              Income  1.Proximity to city  2.Proximity to schools  \
count      70.000000            70.000000               70.000000
mean    99000.000000             3.628571                3.442857
std     59670.593345             0.870972                1.016326
min     35000.000000             1.000000                2.000000
25%     55000.000000             3.000000                3.000000
50%     75000.000000             4.000000                3.000000
75%    115000.000000             4.000000                4.000000
max    200000.000000             5.000000                5.000000

       3. Proximity to transport  4. Proximity to work place  \
count                  70.000000                   70.000000
mean                    4.071429                    3.842857
std                     0.728736                    0.942333
min                     3.000000                    2.000000
25%                     4.000000                    3.000000
50%                     4.000000                    4.000000
75%                     5.000000                    5.000000
max                     5.000000                    5.000000

       5. Proximity to shopping  1. Gym/Pool/Sports facility  \
count                 70.000000                    70.000000
mean                   2.628571                     3.242857
std                    0.783367                     1.134897
min                    1.000000                     1.000000
25%                    2.000000                     3.000000
50%                    3.000000                     3.000000
75%                    3.000000                     4.000000
max                    4.000000                     5.000000

       2. Parking space  3.Power back-up  4.Water supply  ...  \
count         70.000000        70.000000       70.000000  ...
mean           3.528571         3.500000        3.914286  ...
std            0.696189         0.607919        0.675511  ...
min            2.000000         2.000000        2.000000  ...
25%            3.000000         3.000000        4.000000  ...
50%            3.500000         3.500000        4.000000  ...
75%            4.000000         4.000000        4.000000  ...
max            5.000000         5.000000        5.000000  ...

       1. Builder reputation  2. Appreciation potential  \
count              70.000000                  70.000000
mean                4.328571                   4.171429
std                 0.756066                   0.613175
min                 2.000000                   3.000000
25%                 4.000000                   4.000000
50%                 4.000000                   4.000000
75%                 5.000000                   5.000000
max                 5.000000                   5.000000

       3. Profile of neighbourhood  4. Availability of domestic help  \
count                    70.000000                         70.000000
mean                      3.842857                          3.142857
std                       0.714969                          0.982244
min                       2.000000                          1.000000
25%                       3.000000                          2.000000
50%                       4.000000                          3.000000
75%                       4.000000                          4.000000
max                       5.000000                          5.000000

            Time         Size     Budgets  Maintainances         EMI.1  \
count  70.000000    70.000000   70.000000      70.000000     70.000000
mean    7.328571  1120.000000   64.142857   38001.714286  46107.142857
std     4.994842   627.301559   40.769069   26185.208291  22468.317929
min     3.000000   300.000000   12.500000     120.000000  10000.000000
25%     3.000000   800.000000   32.500000   15000.000000  27500.000000
50%     9.000000   800.000000   52.500000   30000.000000  42500.000000
75%     9.000000  1600.000000   87.500000   50000.000000  57500.000000
max    18.000000  4000.000000  150.000000  120000.000000  80000.000000

            ages
count  70.000000
mean   44.328571
std    12.956417
min    21.500000
25%    30.500000
50%    40.500000
75%    53.000000
max    70.000000

[8 rows x 31 columns]
```

In [7]:
```python
from sklearn.preprocessing import StandardScaler

# Select only numerical variables
numerical_data = survey_data.select_dtypes(include=['int64', 'float64'])
```

In [9]:
```python
# Standardize the data
scaler = StandardScaler()
survey_data_scaled = scaler.fit_transform(numerical_data)
```

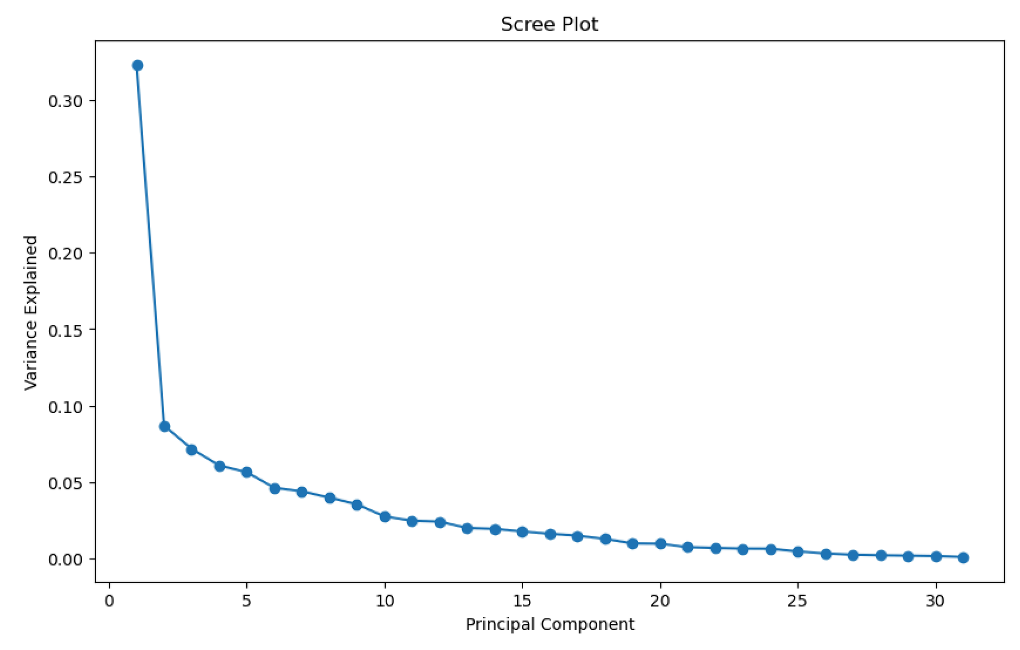
In [11]:
```python
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns
```

In [13]:
```python
# Perform PCA
pca = PCA()
pca_result = pca.fit_transform(survey_data_scaled)
```

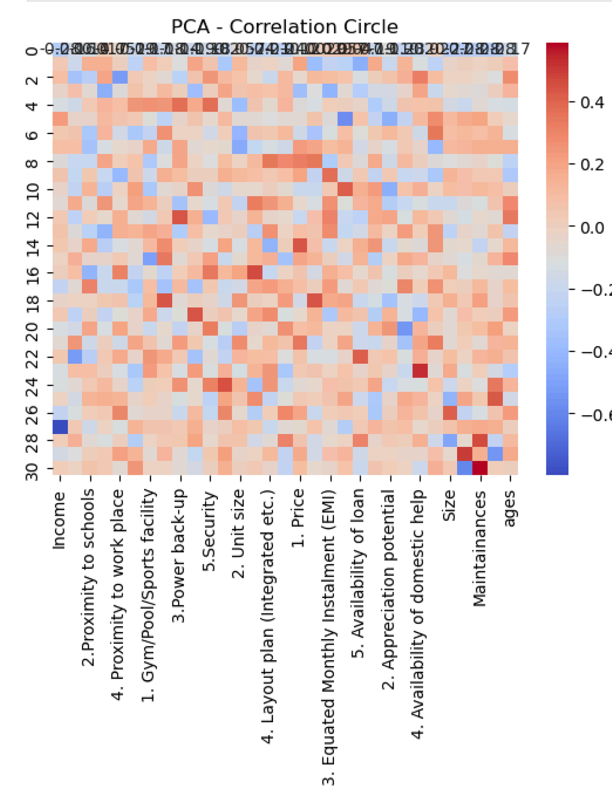
In [15]:
```python
# Summary of PCA results
explained_variance = pca.explained_variance_ratio_
print(f'Explained variance: {explained_variance}')
```

```
Explained variance: [0.32332366 0.08683237 0.07160269 0.06073222 0.05629444 0.04609823
 0.0437729  0.0396524  0.03529825 0.02731472 0.02451182 0.02392867
 0.01975551 0.01917115 0.01748088 0.01597239 0.01474926 0.01261225
 0.00966406 0.00947208 0.00718757 0.00666932 0.00626669 0.00620703
 0.00443108 0.0030697  0.00223431 0.00186911 0.00164179 0.00139072
 0.00079276]
```

In [17]:
```python
# Visualize the scree plot
plt.figure(figsize=(10, 6))
plt.plot(range(1, len(explained_variance) + 1), explained_variance, marker='o')
plt.title('Scree Plot')
plt.xlabel('Principal Component')
plt.ylabel('Variance Explained')
plt.show()
```



In [19]:
```python
# Visualize the variables on the principal component map (Correlation Circle)
sns.heatmap(pd.DataFrame(pca.components_, columns=numerical_data.columns),
            cmap='coolwarm', annot=True)
plt.title('PCA - Correlation Circle')
plt.show()
```



In [23]:
```
!pip install factor_analyzer
```

```
Collecting factor_analyzer
  Downloading factor_analyzer-0.5.1.tar.gz (42 kB)
     ---------------------------------------- 0.0/42.8 kB ? eta -:--:--
     --------------------------- ---------- 30.7/42.8 kB 660.6 kB/s eta 0:00:01
     -------------------------------------- 42.8/42.8 kB 417.7 kB/s eta 0:00:00
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
  Preparing metadata (pyproject.toml): started
  Preparing metadata (pyproject.toml): finished with status 'done'
Requirement already satisfied: pandas in c:\users\nihar\anaconda3\lib\site-packages (from factor_analyzer) (2.1.4)
Requirement already satisfied: scipy in c:\users\nihar\anaconda3\lib\site-packages (from factor_analyzer) (1.11.4)
Requirement already satisfied: numpy in c:\users\nihar\anaconda3\lib\site-packages (from factor_analyzer) (1.26.4)
Requirement already satisfied: scikit-learn in c:\users\nihar\anaconda3\lib\site-packages (from factor_analyzer) (1.2.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\nihar\anaconda3\lib\site-packages (from pandas->factor_analyzer) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\nihar\anaconda3\lib\site-packages (from pandas->factor_analyzer) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\nihar\anaconda3\lib\site-packages (from pandas->factor_analyzer) (2023.3)
Requirement already satisfied: joblib>=1.1.1 in c:\users\nihar\anaconda3\lib\site-packages (from scikit-learn->factor_analyzer) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\nihar\anaconda3\lib\site-packages (from scikit-learn->factor_analyzer) (2.2.0)
Requirement already satisfied: six>=1.5 in c:\users\nihar\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas->factor_analyzer) (1.16.0)
Building wheels for collected packages: factor_analyzer
  Building wheel for factor_analyzer (pyproject.toml): started
  Building wheel for factor_analyzer (pyproject.toml): finished with status 'done'
  Created wheel for factor_analyzer: filename=factor_analyzer-0.5.1-py2.py3-none-any.whl size=42623 sha256=23f730d2cf4d59188639bc10abb24905488d8dfc037c119a25e1280770869b1d
  Stored in directory: c:\users\nihar\appdata\local\pip\cache\wheels\fa\f7\53\a55a8a56668a6fe0199e0e02b6e0ae3007ec35cdf6e4c25df7
Successfully built factor_analyzer
Installing collected packages: factor_analyzer
Successfully installed factor_analyzer-0.5.1
```

In [25]:
```python
from factor_analyzer import FactorAnalyzer, calculate_kmo, calculate_bartlett_sphericity
```

In [27]:
```python
# Determine the number of factors using parallel analysis (not directly available in Python, we use KMO and Bartlett's test)
kmo_all, kmo_model = calculate_kmo(survey_data_scaled)
bartlett_test, p_value = calculate_bartlett_sphericity(survey_data_scaled)
print(f'KMO Test: {kmo_model}, Bartlett\'s Test: {bartlett_test}, p-value: {p_value}')
```

```
KMO Test: 0.7154075637394491, Bartlett's Test: 1562.964633118281, p-value: 1.0805923579037055e-118
C:\Users\nihar\anaconda3\Lib\site-packages\factor_analyzer\utils.py:244: UserWarning: The inverse of the variance-covariance matrix was calculated using the Moore-Penrose generalized matrix inversion, due to its determinant being at or very close to zero.
  warnings.warn(
```

In [29]:
```python
from factor_analyzer import FactorAnalyzer
import seaborn as sns
import matplotlib.pyplot as plt
```

In [31]:
```python
# Perform Factor Analysis with the chosen number of factors (e.g., 3 factors)
fa = FactorAnalyzer(n_factors=3, rotation='varimax')
fa.fit(survey_data_scaled)
```

Out[31]:
```
FactorAnalyzer
FactorAnalyzer(rotation='varimax', rotation_kwargs={})
```

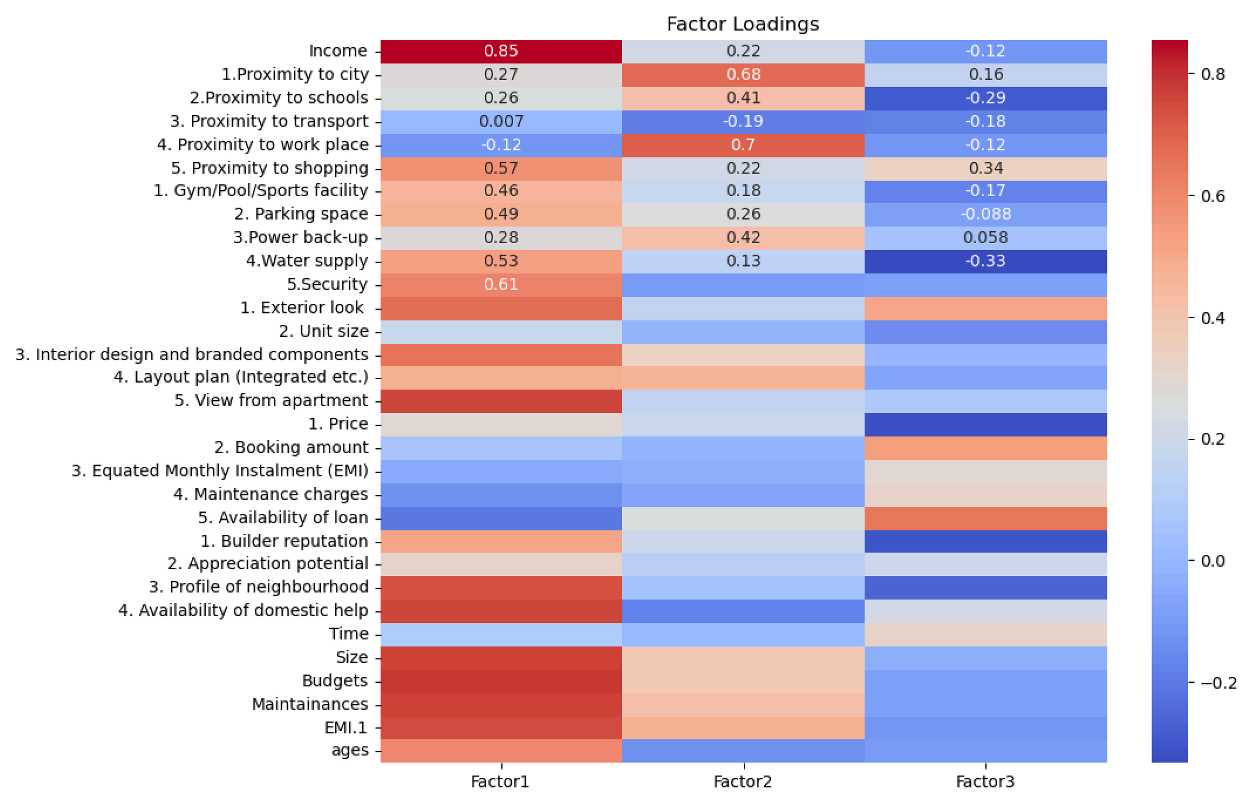
In [33]:
```python
# Extract Factor Loadings
fa_loadings = fa.loadings_
```

In [35]:
```python
# Convert loadings to DataFrame for better readability
fa_loadings_df = pd.DataFrame(fa_loadings, index=numerical_data.columns, columns=[f'Factor{i+1}' for i in range(fa_loadings.shape[1])])
```

In [37]:
```python
# Print Factor Loadings
print("Factor Loadings:\n", fa_loadings_df)
```

```
Factor Loadings:
                                             Factor1   Factor2   Factor3
Income                                     0.854368  0.223421 -0.122097
1.Proximity to city                        0.273435  0.676886  0.160120
2.Proximity to schools                     0.255414  0.414996 -0.288447
3. Proximity to transport                  0.007036 -0.193255 -0.183912
4. Proximity to work place                -0.123135  0.698454 -0.123997
5. Proximity to shopping                   0.570211  0.220357  0.339787
1. Gym/Pool/Sports facility                0.463474  0.176776 -0.173102
2. Parking space                           0.486947  0.256420 -0.088400
3.Power back-up                            0.276637  0.415499  0.058022
4.Water supply                             0.529570  0.133722 -0.333262
5.Security                                 0.610129 -0.102809 -0.082725
1. Exterior look                           0.665966  0.158172  0.509328
2. Unit size                               0.177732 -0.020372 -0.138686
3. Interior design and branded components  0.659207  0.332463 -0.005462
4. Layout plan (Integrated etc.)           0.490574  0.473421 -0.060010
5. View from apartment                     0.757326  0.151734  0.089531
1. Price                                   0.285582  0.180088 -0.318506
2. Booking amount                          0.064717 -0.021876  0.536904
3. Equated Monthly Instalment (EMI)       -0.053834 -0.037522  0.280665
4. Maintenance charges                    -0.125445 -0.069941  0.328757
5. Availability of loan                   -0.203902  0.252198  0.649799
1. Builder reputation                      0.508115  0.195520 -0.308135
2. Appreciation potential                  0.315163  0.123985  0.198809
3. Profile of neighbourhood                0.732493  0.048999 -0.262209
4. Availability of domestic help           0.754177 -0.179168  0.212676
Time                                       0.095711  0.003007  0.314520
Size                                       0.767762  0.378715 -0.031844
Budgets                                    0.785836  0.379924 -0.083078
Maintainances                              0.768112  0.416432 -0.080610
EMI.1                                      0.743914  0.489109 -0.124165
ages                                       0.595834 -0.136588 -0.105576
```

In [39]:
```python
# Plot Factor Analysis results
plt.figure(figsize=(10, 8))
sns.heatmap(fa_loadings_df, annot=True, cmap='coolwarm')
plt.title('Factor Loadings')
plt.show()
```


Factor Loadings
Income
1.Proximity to city
2.Proximity to schools
3. Proximity to transport
4. Proximity to work place
5. Proximity to shopping
1. Gym/Pool/Sports facility
2. Parking space
3.Power back-up
4.Water supply
5.Security
1. Exterior look
2. Unit size
3. Interior design and branded components
4. Layout plan (Integrated etc.)
5. View from apartment
1. Price
2. Booking amount
3. Equated Monthly Instalment (EMI)
4. Maintenance charges
5. Availability of loan
1. Builder reputation
2. Appreciation potential
3. Profile of neighbourhood
4. Availability of domestic help
Time
Size
Budgets
Maintainances
EMI.1
ages
0.85 0.22 -0.12
0.27 0.68 0.16
0.26 0.41 -0.29
0.007 -0.19 -0.18
-0.12 0.7 -0.12
0.57 0.22 0.34
0.46 0.18 -0.17
0.49 0.26 -0.088
0.28 0.42 0.058
0.53 0.13 -0.33
0.61
Factor1
Factor2
Factor3
0.8
0.6
0.4
0.2
0.0
−0.2

In [ ]: