sem-Copy1

June 18, 2023

Tarea 3

Instrucciones

Los resultados de los ejericicios propuestos se deben entregar como un notebook por correo electronico a *juancaros@udec.cl* el dia 9/6 hasta las 21:00. Es importante considerar que el código debe poder ejecutarse en cualquier computadora con la data original del repositorio. Recordar la convencion para el nombre de archivo ademas de incluir en su documento titulos y encabezados por seccion. Utilizar la base de datos *ifood_df.csv*.

Como se indica en la Tabla 1, las variables describen el comportamiento de un set de consumidores en una tienda de retail. Las variables categoricas (e.g. educacion, estado civil) ya han sido convertidas a variables binarias (una por cada categoria).

[17]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.formula.api as smf
import sklearn
import scipy
from scipy.linalg import eigh, cholesky
from scipy.stats import norm
import linearmodels.panel as lmp
from pylab import plot, show, axis, subplot, xlabel, ylabel, grid
from sklearn.preprocessing import StandardScaler
import semopy
import seaborn as sns
from factor_analyzer import FactorAnalyzer
from sklearn.decomposition import PCA
from IPython.display import Image


%matplotlib inline
```

[7]:
```
Image(filename='../data/dictionary.png', width=600, height=900, unconfined=True)
```

[7]:

Feature	Description
AcceptedCmp1	1 if costumer accepted the offer in the 1st campaign, 0 otherwise
AcceptedCmp2	1 if costumer accepted the offer in the 2nd campaign, 0 otherwise
AcceptedCmp3	1 if costumer accepted the offer in the 3rd campaign, 0 otherwise
AcceptedCmp4	1 if costumer accepted the offer in the 4th campaign, 0 otherwise
AcceptedCmp5	1 if costumer accepted the offer in the 5th campaign, 0 otherwise
Response (target)	1 if costumer accepted the offer in the last campaign, 0 otherwise
Complain	1 if costumer complained in the last 2 years
DtCustomer	date of customer's enrollment with the company
Education	customer's level of education
Marital	customer's marital status
Kidhome	number of small children in customer's household
Teenhome	number of teenagers in customer's household
Income	customer's yearly household income
MntFishProducts	amount spent on fish products in the last 2 years
MntMeatProducts	amount spent on meat products in the last 2 years
MntFruits	amount spent on fruits in the last 2 years
MntSweetProducts	amount spent on sweet products in the last 2 years
MntWines	amount spent on wines in the last 2 years
MntGoldProds	amount spent on *gold* products in the last 2 years
NumDealsPurchases	number of purchases made with discount
NumCatalogPurchases	number of purchases made using catalogue
NumStorePurchases	number of purchases made directly in stores
NumWebPurchases	number of purchases made through company's web site
NumWebVisitsMonth	number of visits to company's web site in the last month
Recency	number of days since the last purchase

Table 1: Meta-data table

Preguntas:

1. Cargue la base de datos y realice los ajustes necesarios para su uso (missing values, recodificar variables, etcetera). Identifique los tipos de datos que se encuentran en la base, realice estadisticas descriptivas sobre las variables importantes (Hint: Revisar la distribuciones, datos faltantes, outliers, etc.) y limpie las variables cuando sea necesario.

R: Las variables estan balancedas, aproximadas a valores enteros, sin perdida de informacion. Variables como educacion y estado marital estan expresadas de forma binaria.

[14]:
```
df_store=pd.read_csv('../data/ifood_df.csv')
df_store.dtypes
```

[14]:
```
Recency                 int64
Income                  int64
Kidhome                 int64
Teenhome                int64
MntWines                int64
MntFruits               int64
MntMeatProducts         int64
```

```
MntFishProducts          int64
MntSweetProducts         int64
MntGoldProds             int64
NumDealsPurchases        int64
NumWebPurchases          int64
NumCatalogPurchases      int64
NumStorePurchases        int64
NumWebVisitsMonth        int64
AcceptedCmp3             int64
AcceptedCmp4             int64
AcceptedCmp5             int64
AcceptedCmp1             int64
AcceptedCmp2             int64
Complain                 int64
Z_CostContact            int64
Z_Revenue                int64
Response                 int64
Age                      int64
Customer_Days            int64
marital_Divorced         int64
marital_Married          int64
marital_Single           int64
marital_Together         int64
marital_Widow            int64
education_2n Cycle       int64
education_Basic          int64
education_Graduation     int64
education_Master         int64
education_PhD            int64
MntTotal                 int64
MntRegularProds          int64
AcceptedCmpOverall       int64
dtype: object
```

2. Realice un PCA usando las variables de numero de compras y cantidad gastada en los diversos items. En particular, identifique los valores propios y determine el numero optimo de componentes. Luego estime y grafique la distribucion de los componentes. Ademas discuta la importancia relativa de las variables sobre cada uno de los componentes estimados. Que se puede concluir de este analisis?

R: El analisis muestra tres compononentes principales que contribuyen informacion significativa al analisis (sobre las variables estandarizadas). El segundo y tercer componente aumentan su varianza en funcion de los valores del primero. A la inversa, valores negativos de PC1 se asocian a una menor varianza en PC2 y PC3.

[112]:
```
df=df_store[['MntWines','MntFruits','MntMeatProducts','MntFishProducts','MntSweetProducts'
 ↪,'MntGoldProds','NumDealsPurchases','NumWebPurchases','NumCatalogPurchases',
```

```
                 'NumStorePurchases']]
df.describe()
```

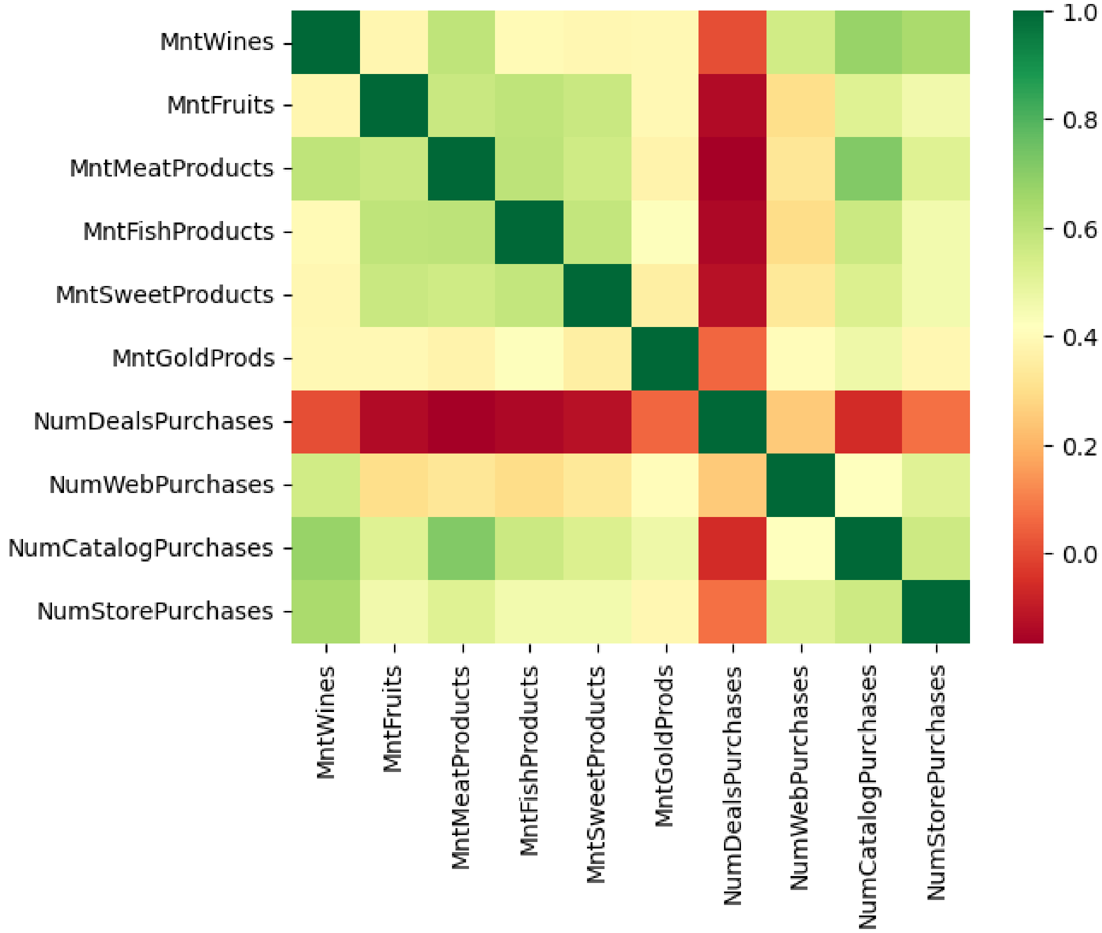
[112]:

```
          MntWines    MntFruits  MntMeatProducts  MntFishProducts  \
count  2205.000000  2205.000000      2205.000000      2205.000000
mean    306.164626    26.403175       165.312018        37.756463
std     337.493839    39.784484       217.784507        54.824635
min       0.000000     0.000000         0.000000         0.000000
25%      24.000000     2.000000        16.000000         3.000000
50%     178.000000     8.000000        68.000000        12.000000
75%     507.000000    33.000000       232.000000        50.000000
max    1493.000000   199.000000      1725.000000       259.000000

       MntSweetProducts  MntGoldProds  NumDealsPurchases  NumWebPurchases  \
count       2205.000000   2205.000000        2205.000000      2205.000000
mean          27.128345     44.057143           2.318367         4.100680
std           41.130468     51.736211           1.886107         2.737424
min            0.000000      0.000000           0.000000         0.000000
25%            1.000000      9.000000           1.000000         2.000000
50%            8.000000     25.000000           2.000000         4.000000
75%           34.000000     56.000000           3.000000         6.000000
max          262.000000    321.000000          15.000000        27.000000

       NumCatalogPurchases  NumStorePurchases
count          2205.000000        2205.000000
mean              2.645351           5.823583
std               2.798647           3.241796
min               0.000000           0.000000
25%               0.000000           3.000000
50%               2.000000           5.000000
75%               4.000000           8.000000
max              28.000000          13.000000
```

[16]:

```
sns.heatmap(df.corr(), cmap='RdYlGn')
```

[16]: <AxesSubplot:>

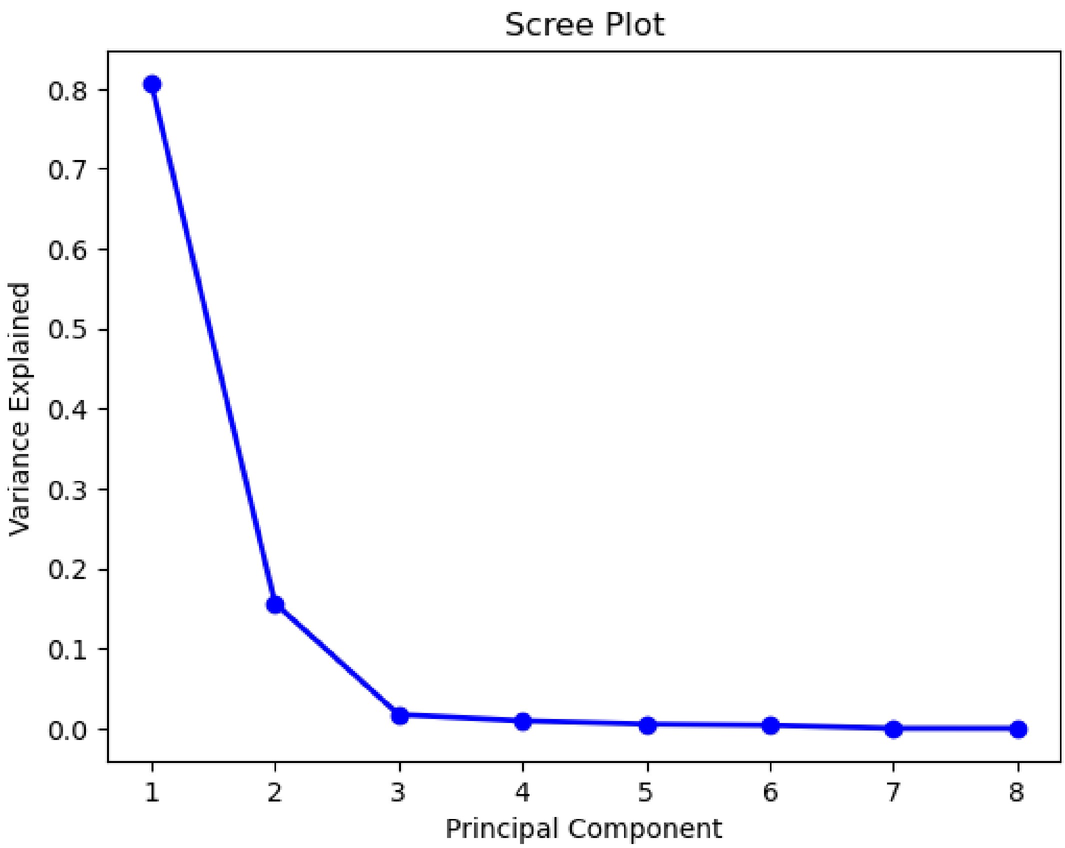


[67]:
```
scaler = StandardScaler()
df_st = scaler.fit_transform(df)
pca = PCA(n_components=8)
pca_features = pca.fit_transform(df)
print(pca.explained_variance_ratio_)
```

```
[8.06886286e-01 1.56527924e-01 1.74127928e-02 9.58296648e-03
 5.41714601e-03 4.07760213e-03 3.90496989e-05 2.46915096e-05]
```

[26]:
```
#scree plot using explained variance proportion

PC_values = np.arange(pca.n_components_) + 1
plt.plot(PC_values, pca.explained_variance_ratio_, 'o-', linewidth=2,␣
 ↪color='blue')
plt.title('Scree Plot')
plt.xlabel('Principal Component')
plt.ylabel('Variance Explained')
```

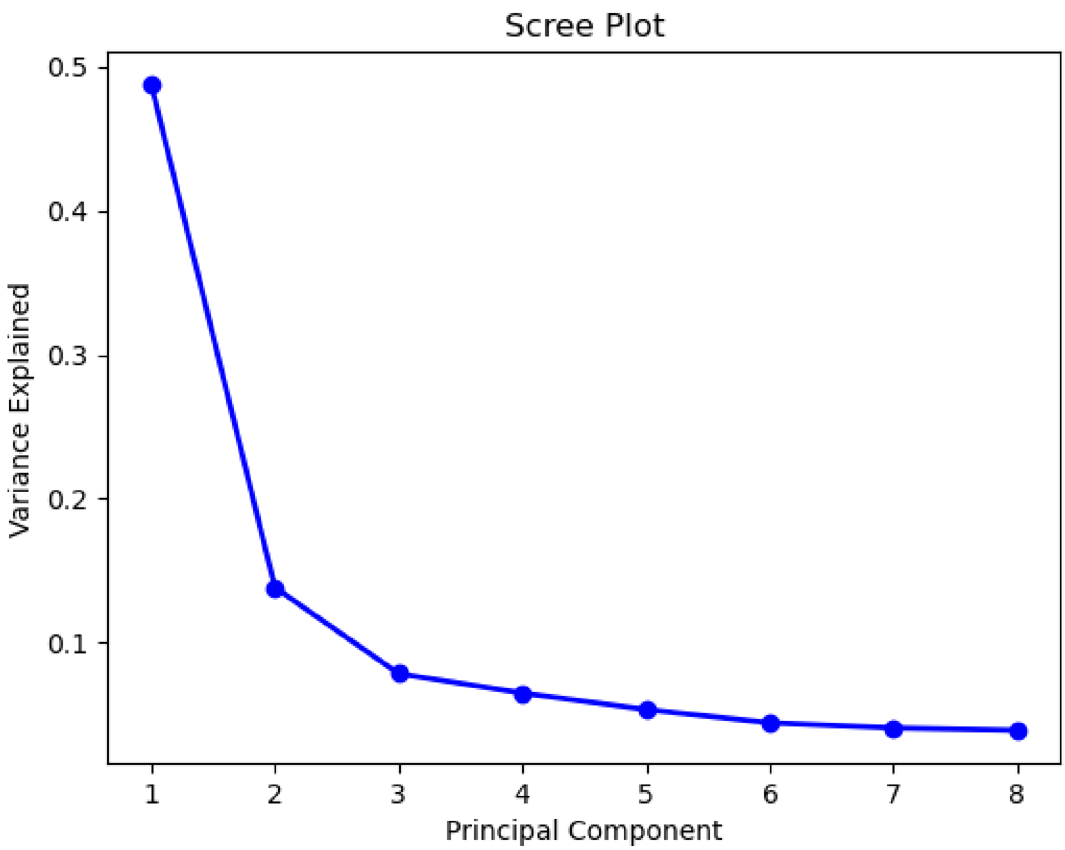
```
plt.show()
```



[27]:
```
pca = PCA(n_components=8)
pca_features = pca.fit_transform(df_st)
print(pca.explained_variance_ratio_)
```

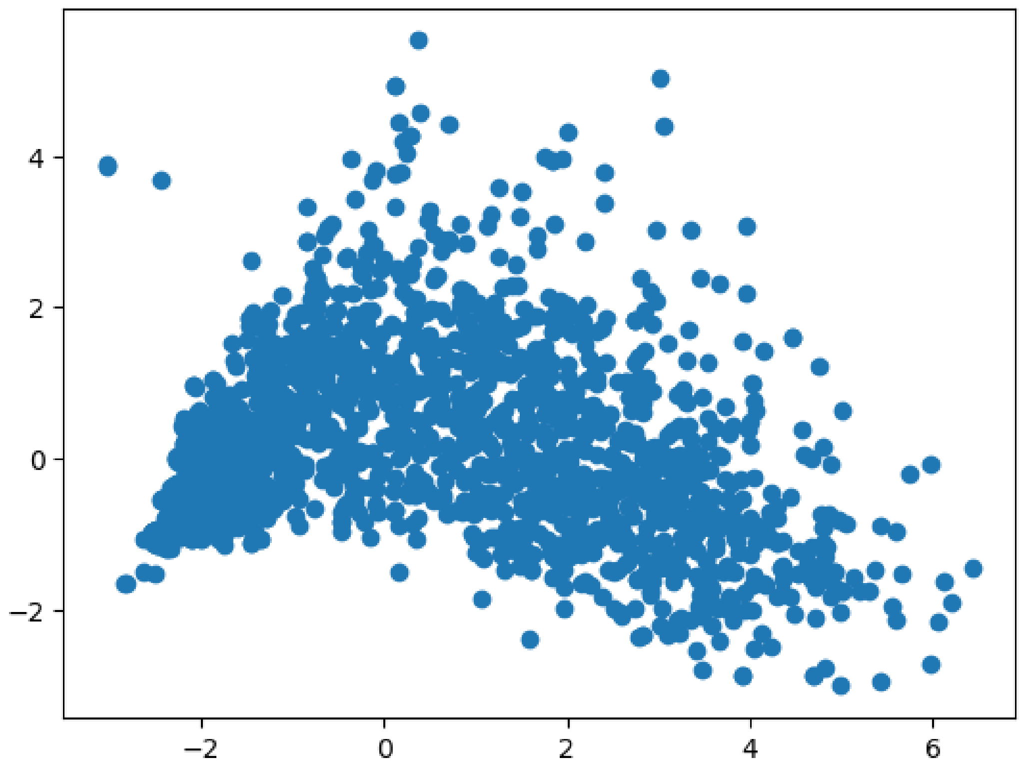
```
[0.48773352 0.13832059 0.07850407 0.06503529 0.05361044 0.04440343
 0.04082509 0.03921513]
```

[28]:
```
#scree plot using explained variance proportion (standarized data)

PC_values = np.arange(pca.n_components_) + 1
plt.plot(PC_values, pca.explained_variance_ratio_, 'o-', linewidth=2,␣
 ↪color='blue')
plt.title('Scree Plot')
plt.xlabel('Principal Component')
plt.ylabel('Variance Explained')
plt.show()
```



[117]:
```
pca_df = pd.DataFrame(data=pca_features,columns=['PC1', 'PC2', 'PC3'])
plt.scatter(pca_df['PC1'],pca_df['PC2'])
```

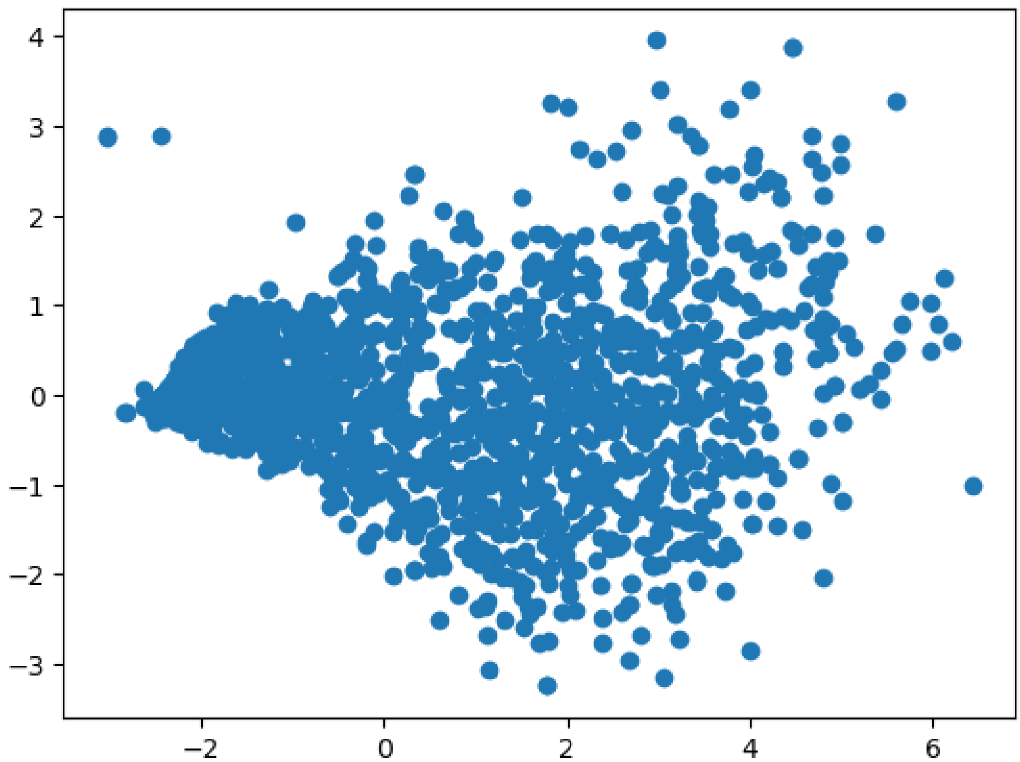
[117]: <matplotlib.collections.PathCollection at 0x241575fcd48>



[118]:

```
plt.scatter(pca_df['PC1'],pca_df['PC3'])
```

[118]: <matplotlib.collections.PathCollection at 0x24157531948>



3. Con los resultados de la Pregunta 2, mantenga los primeros 3 componentes principales y repita el analisis. Graficamente y estadisticamente indique si existen diferencias o relaciones significativas entre los valores de los PCA y las siguientes variables: Income, Kidhome, Education y Recency. Que puede concluir de los resultados?

R: Mayores valores en PC1 se asocian a una menor probabilidad de tener hijos e ingreso mas alto. No se observan patrones obvios por educacion o Recency. En otras palabras, los componentes estimados correlacionan con algunas de las variables demograficas pero no se asocian a la latencia desde la ultima compra.

[113]:
```
pca = PCA(n_components=3)
pca_features = pca.fit_transform(df_st)
print(pca.explained_variance_ratio_)
```

```
[0.48773352 0.13832059 0.07850407]
```

[114]:
```
pca_vectors = pd.DataFrame(data = pca.components_)
pca_vectors.head()
```

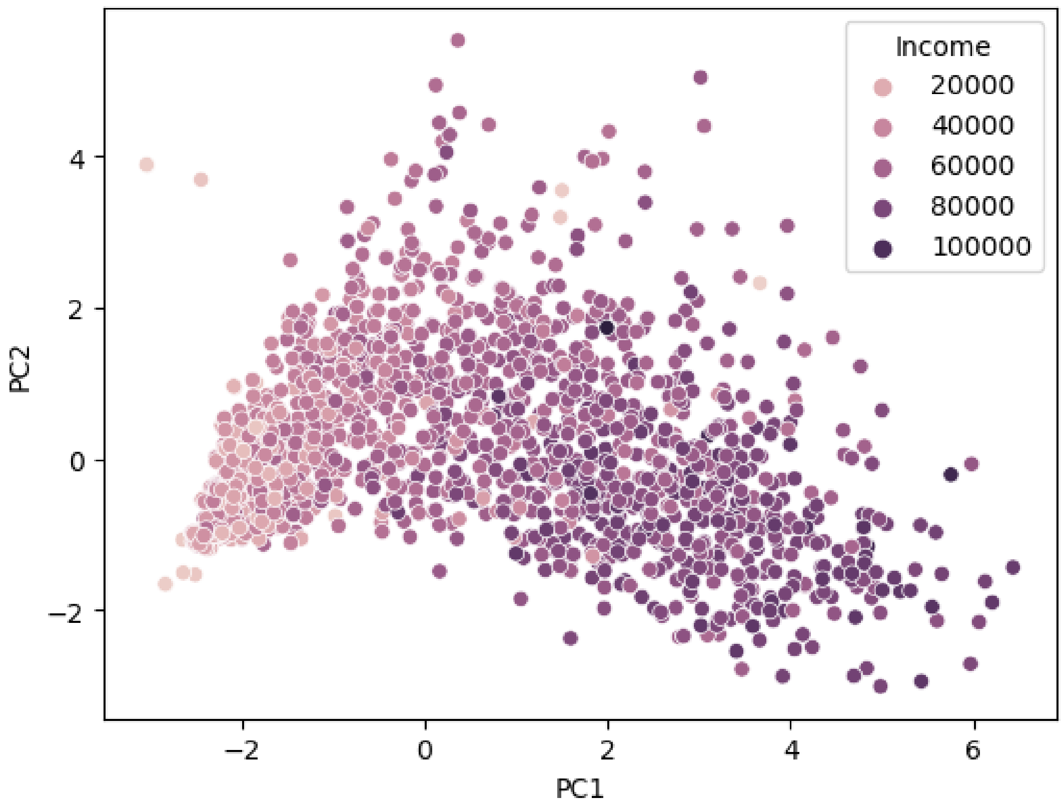
[114]:
```
          0         1         2         3         4         5         6  \
0  0.345652  0.329467  0.367261  0.339410  0.328118  0.279623 -0.026615
1  0.209308 -0.228660 -0.180425 -0.237768 -0.204254  0.145564  0.696183
2 -0.507871  0.322237 -0.197585  0.304310  0.279365  0.432519  0.385171
```

```
          7         8         9
0  0.273978  0.378615  0.341941
1  0.475368 -0.012596  0.202743
2 -0.033853 -0.212138 -0.216711
```

[119]:
```
pca_df = pd.DataFrame(data=pca_features,columns=['PC1', 'PC2', 'PC3'])
features = df_store[['Income','Kidhome','education_Master','Recency']]
pca_df = pd.concat([pca_df, features], axis=1, join='inner')
```

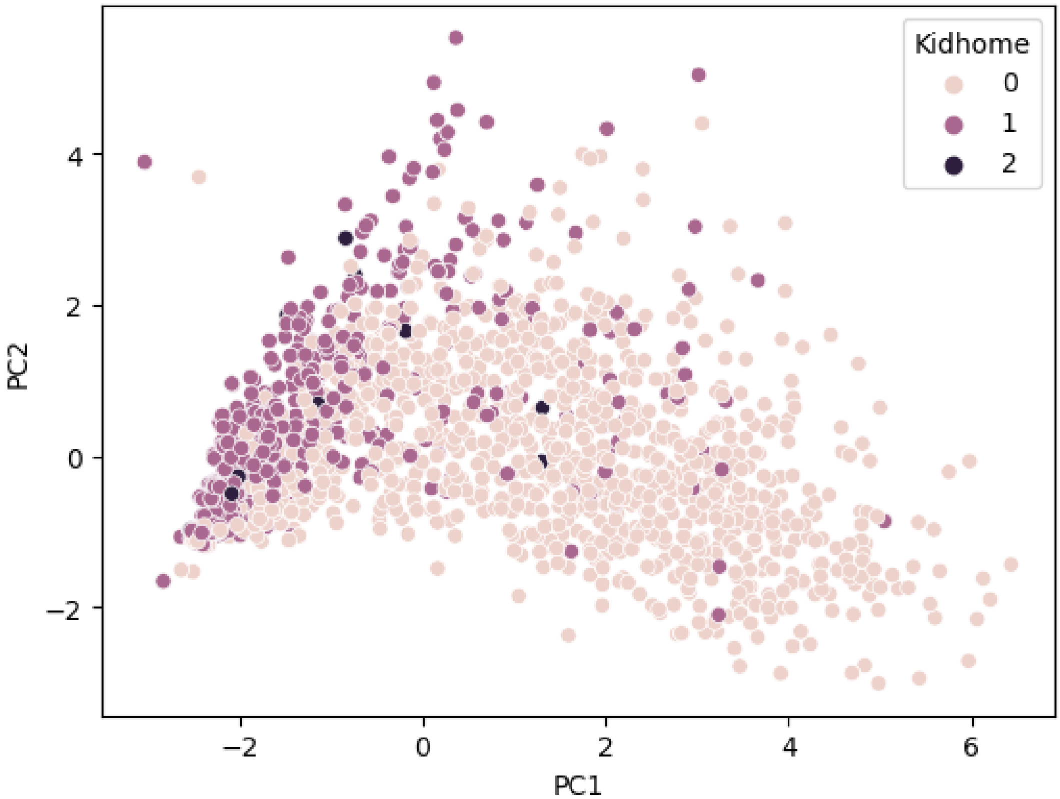
[39]:
```
sns.scatterplot(data=pca_df, y='PC2' , x='PC1', hue='Income')
```

[39]: <AxesSubplot:xlabel='PC1', ylabel='PC2'>



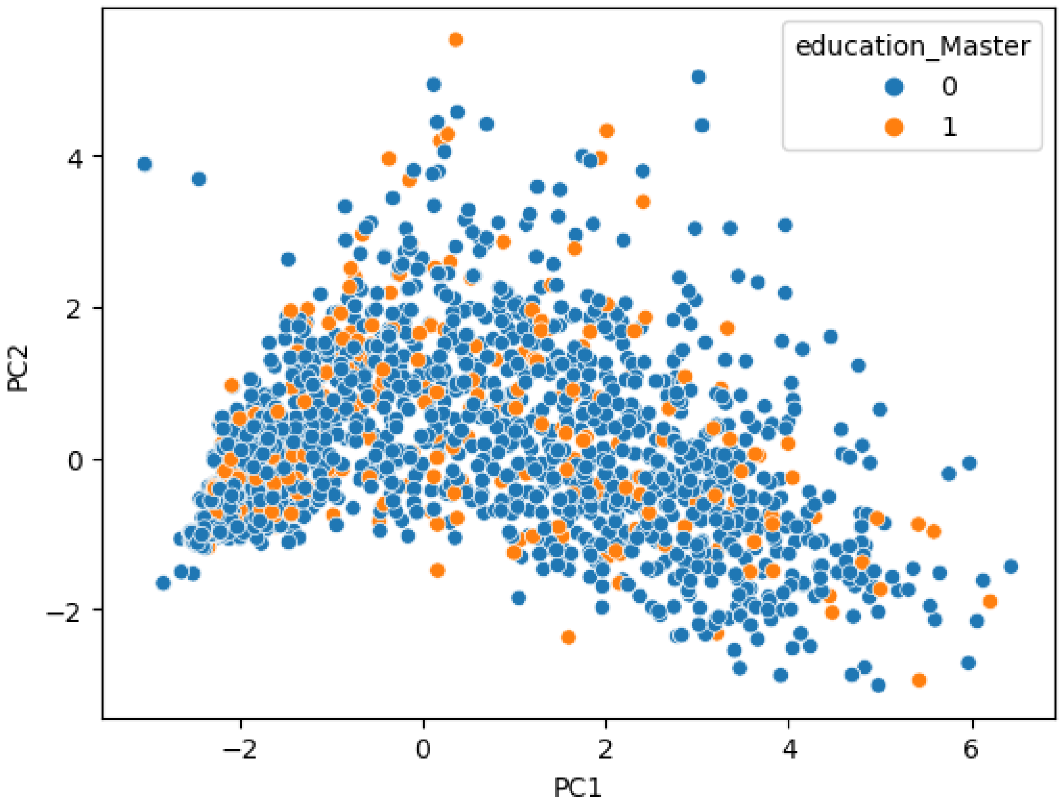
[40]:
```
sns.scatterplot(data=pca_df, y='PC2' , x='PC1', hue='Kidhome')
```

[40]: <AxesSubplot:xlabel='PC1', ylabel='PC2'>



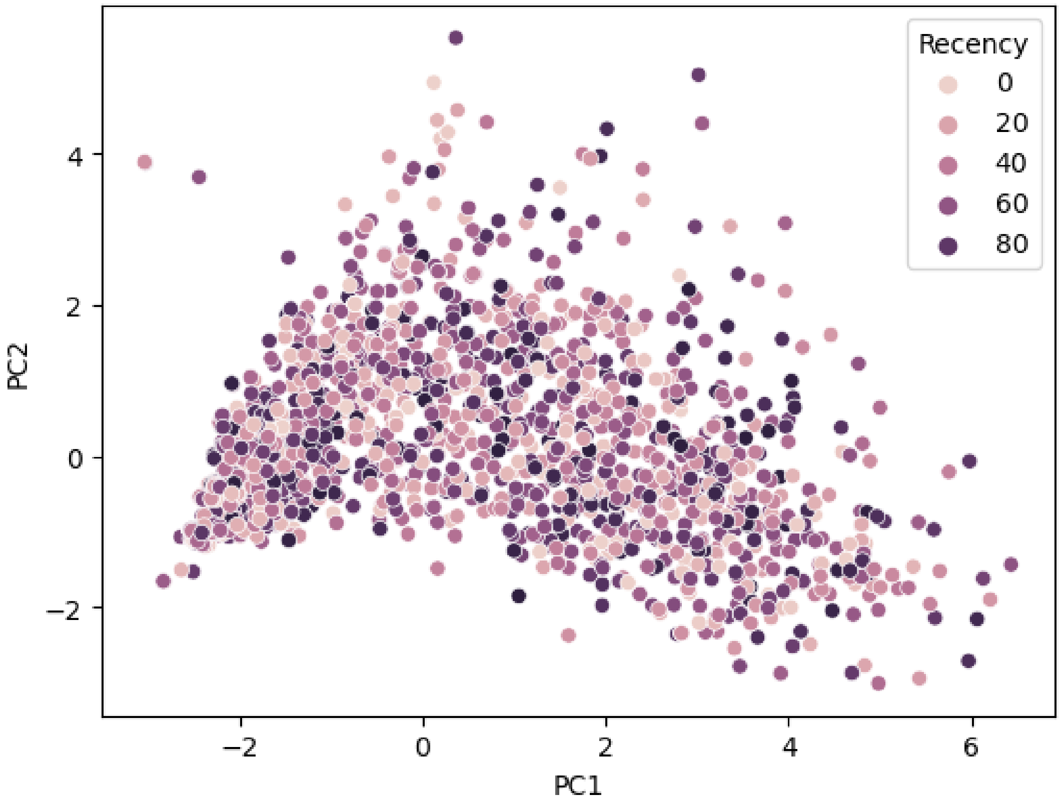

```
[41]: sns.scatterplot(data=pca_df, y='PC2' , x='PC1', hue='education_Master')
```

```
[41]: <AxesSubplot:xlabel='PC1', ylabel='PC2'>
```



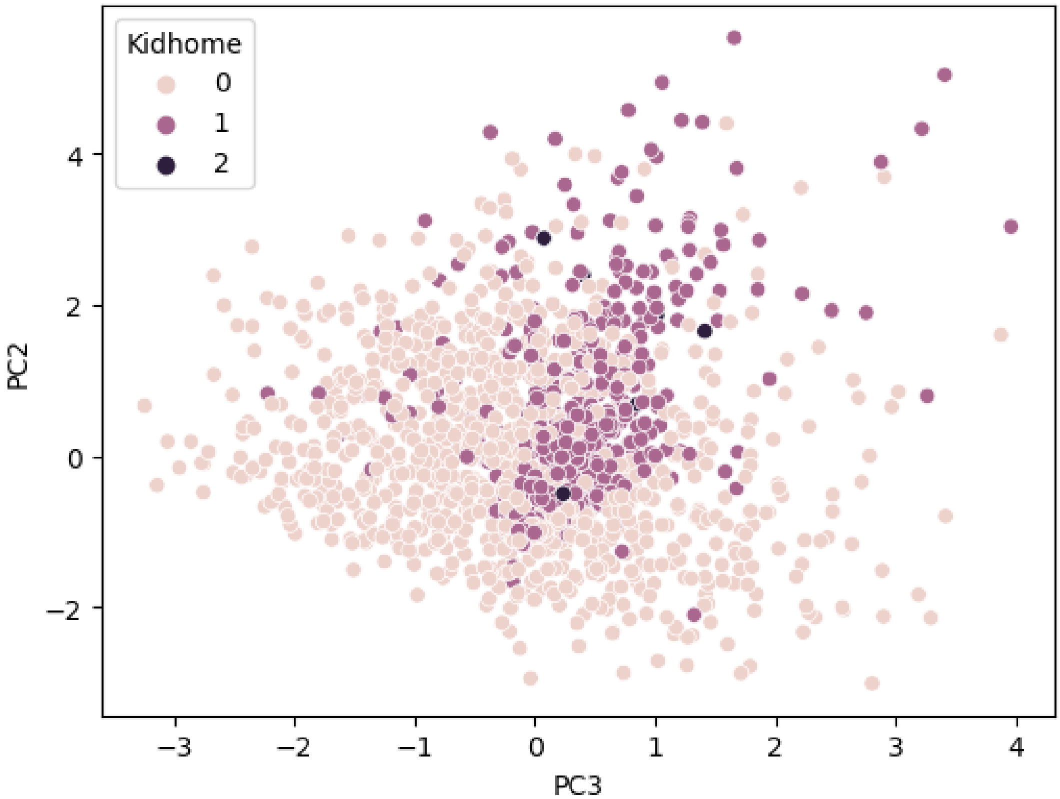
[42]:
```
sns.scatterplot(data=pca_df, y='PC2' , x='PC1', hue='Recency')
```

[42]: <AxesSubplot:xlabel='PC1', ylabel='PC2'>



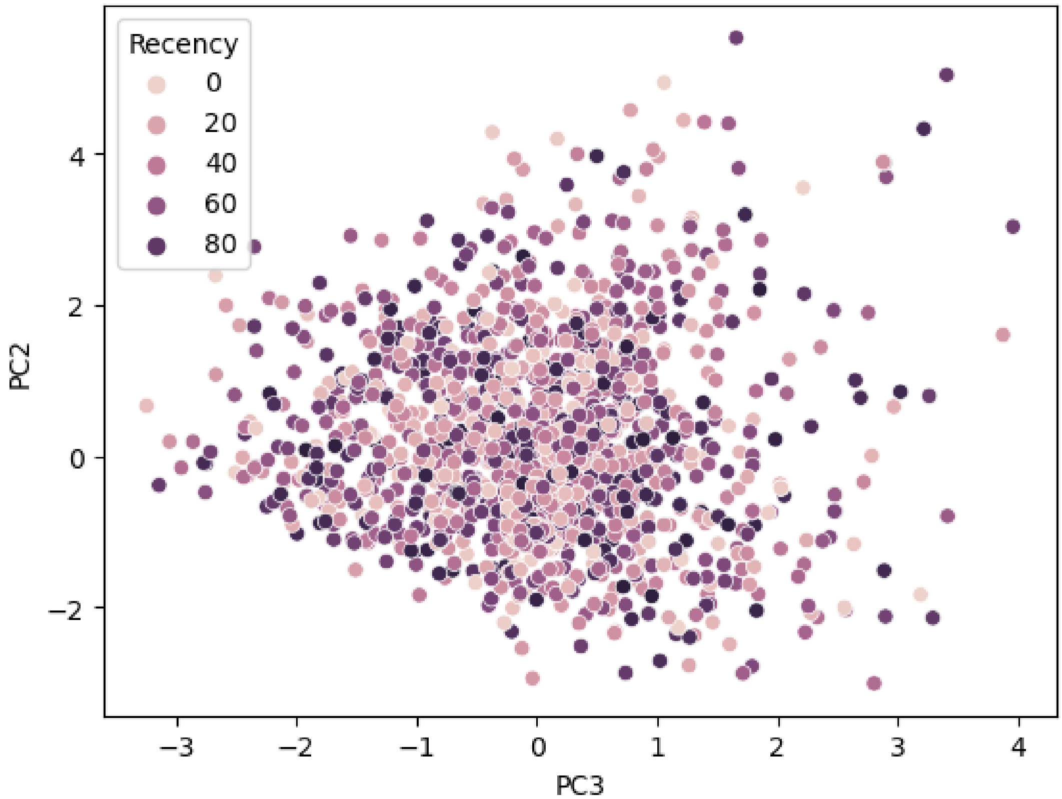

```
[124]: sns.scatterplot(data=pca_df, y='PC2' , x='PC3', hue='Kidhome')
```

[124]: <AxesSubplot:xlabel='PC3', ylabel='PC2'>



[122]:
```
sns.scatterplot(data=pca_df, y='PC2' , x='PC3', hue='Recency')
```

[122]: <AxesSubplot:xlabel='PC3', ylabel='PC2'>



4. A partir del mismo set de variables de la pregunta 2 realice un EFA. En particular determine el numero optimo de factores y las variables que se asocian a cada factor. Tambien discuta si existen variables que no son informativas.

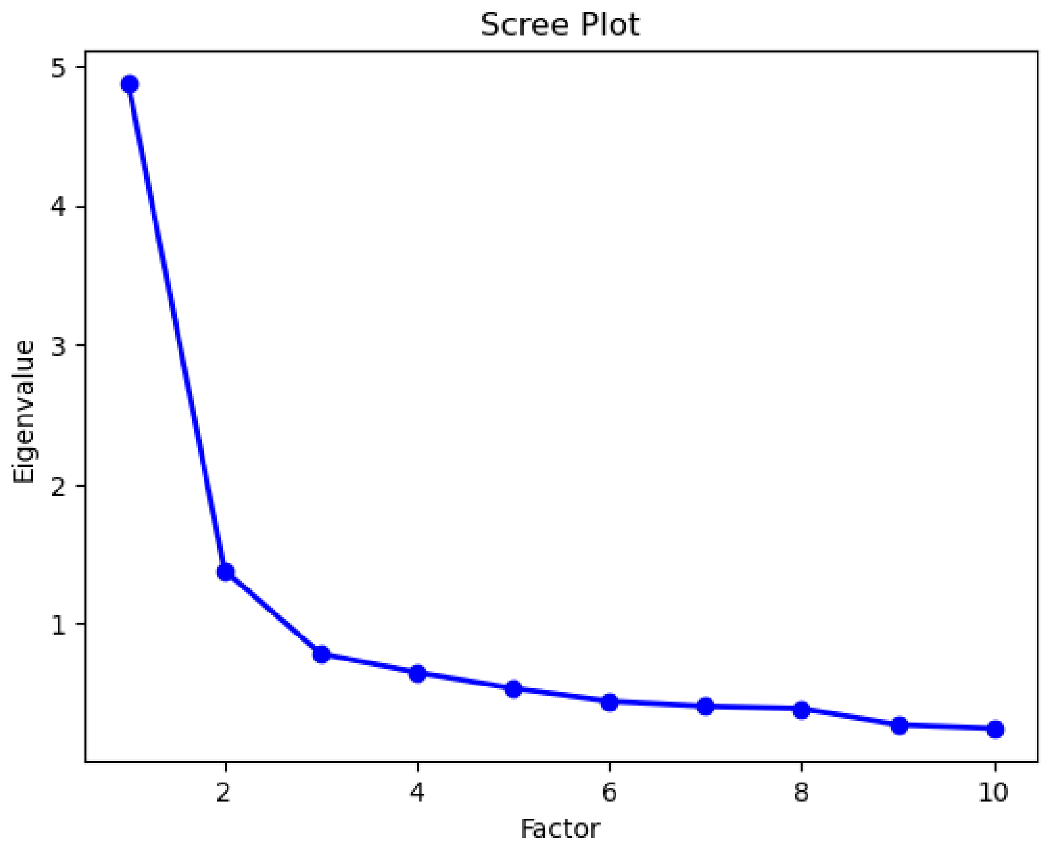
R: El analisis exploratorio sugiere la presencia de un factor dominante y al menos dos que podrian ser relevantes para resumir la data. Los resultados se confirman en ambos metodos usados (considerando tambien la data estandarizada). NumDealsPurchases no parece contribuir informacion relevante al modelo, lo cual es consistente con el analisis de la matriz de correlacion.

[126]:
```
# Create factor analysis object and perform factor analysis
fa = FactorAnalyzer(rotation='oblimax', method='ml')
fa.fit(df)
fa.get_factor_variance()
```

[126]:
```
(array([4.47466619, 0.86347937, 0.42355804]),
 array([0.44746662, 0.08634794, 0.0423558 ]),
 array([0.44746662, 0.53381456, 0.57617036]))
```

[127]:
```
values = np.arange(1,11)
eigenvalues = pd.DataFrame(data=fa.get_eigenvalues())
plt.plot(values, eigenvalues.loc[0], 'o-', linewidth=2, color='blue')
plt.title('Scree Plot')
```

```
plt.xlabel('Factor')
plt.ylabel('Eigenvalue')
plt.show()
```



[128]:
```
fa.loadings_
```

[128]:
```
array([[ 0.80043827,  0.34055811, -0.20095447],
       [ 0.65877455, -0.32850053,  0.17176946],
       [ 0.79909929, -0.21501984, -0.17281919],
       [ 0.6876218 , -0.36019325,  0.15478428],
       [ 0.6568856 , -0.29468963,  0.17865296],
       [ 0.54571807,  0.01609312,  0.21622698],
       [-0.04979688,  0.40588141,  0.2937617 ],
       [ 0.58839609,  0.42634552,  0.3323678 ],
       [ 0.82784434, -0.0359942 , -0.13257965],
       [ 0.7247583 ,  0.16943277,  0.08266383]])
```

[129]:
```
print(semopy.efa.explore_cfa_model(df, pval=0.05))
```

```
eta1 =~ MntMeatProducts + MntFishProducts + MntFruits + MntSweetProducts +
```

```
MntWines + MntGoldProds
```

[130]:
```
df_st = pd.DataFrame(df_st)
df_st.columns=df.columns.values
print(semopy.efa.explore_cfa_model(df_st, pval=0.05))
```

```
eta1 =~ NumCatalogPurchases + MntMeatProducts + MntWines + NumStorePurchases +
MntFishProducts + MntFruits + MntSweetProducts + MntGoldProds + NumWebPurchases
```

5. Con los resultados obtenidos en la Pregunta 4, proponga un CFA donde cada variable solo se asocia con un factor. Entregue un nombre a cada factor que representa el concepto comun entre todas las variables. Reporte la importancia de cada medida (variable) a cada factor e indique la correlacion entre factores.

R: Existen dos posibles modelos, uno con un solo factor asociado a todas las variables, o separarlas entre volumen de ventas ($) y numero de compras. Ambos modelos presentan indices de ajuste bastante similar, sin embargo el modelo con un factor es marginalmente mejor.

[133]:
```
mod1 = """
# measurement model
eta1 =~ NumCatalogPurchases + MntMeatProducts + MntWines + NumStorePurchases +␣
 ↪MntFishProducts + MntFruits + MntSweetProducts + MntGoldProds +␣
 ↪NumWebPurchases
    """
model1 = semopy.Model(mod1)
out1=model1.fit(df_st)

mod2 = """
# measurement model
eta1 =~ NumCatalogPurchases + NumStorePurchases + NumWebPurchases
eta2 =~ MntSweetProducts + MntFruits + MntGoldProds + MntMeatProducts +␣
 ↪MntWines + MntFishProducts
    """
model2 = semopy.Model(mod2)
out2=model2.fit(df_st)

semopy.calc_stats(model1)
```

[133]:
```
       DoF  DoF Baseline         chi2  chi2 p-value  chi2 Baseline      CFI  \
Value   27            36  1348.869551           0.0    9995.859047  0.86728

            GFI      AGFI       NFI      TLI     RMSEA        AIC         BIC  \
Value  0.865057  0.820076  0.865057  0.82304  0.149041  34.776536  137.349226

         LogLik
Value  0.611732
```

[134]:
```
semopy.calc_stats(model2)
```

[134]:
```
       DoF  DoF Baseline        chi2  chi2 p-value  chi2 Baseline      CFI  \
Value   26            36  1348.26636           0.0    9995.859047  0.86724

            GFI      AGFI       NFI       TLI     RMSEA        AIC         BIC  \
Value  0.865118  0.81324  0.865118  0.816179  0.151903  36.777083  145.048256

         LogLik
Value  0.611459
```

[86]:
```
model2.inspect(mode='list', what="names", std_est=True)
```

[86]:
```
                   lval  op                rval      Estimate  Est. Std  \
0   NumCatalogPurchases   ~                eta1      1.000000  0.826345
1     NumStorePurchases   ~                eta1      0.992432  0.708492
2       NumWebPurchases   ~                eta1      0.645663  0.546314
3      MntSweetProducts   ~                eta2      1.000000  0.673737
4             MntFruits   ~                eta2      0.965670  0.686074
5          MntGoldProds   ~                eta2      1.072912  0.629672
6       MntMeatProducts   ~                eta2      6.069227  0.741230
7              MntWines   ~                eta2      8.842049  0.720203
8       MntFishProducts   ~                eta2      1.383461  0.705803
9                  eta2  ~~                eta2    784.665136  1.000000
10                 eta2  ~~                eta1     65.479051  1.008228
11                 eta1  ~~                eta1      5.375298  1.000000
12  NumCatalogPurchases  ~~  NumCatalogPurchases      2.496613  0.317155
13            MntFruits  ~~           MntFruits    822.818274  0.529302
14      MntFishProducts  ~~     MntFishProducts   1512.926117  0.501842
15      NumWebPurchases  ~~     NumWebPurchases      5.267242  0.701541
16         MntGoldProds  ~~        MntGoldProds   1374.897581  0.603513
17             MntWines  ~~            MntWines  56925.138297  0.481308
18      MntMeatProducts  ~~     MntMeatProducts  23703.668964  0.450578
19     MntSweetProducts  ~~    MntSweetProducts    943.973238  0.546079
20    NumStorePurchases  ~~   NumStorePurchases      5.252896  0.498040

     Std. Err    z-value p-value
0           -          -       -
1    0.027294    36.3609     0.0
2    0.024451  26.406824     0.0
3           -          -       -
4    0.033319  28.982536     0.0
5    0.039987  26.831396     0.0
6    0.195641  31.022294     0.0
7    0.292272  30.252839     0.0
8     0.04655   29.71997     0.0
9    45.47588  17.254534     0.0
```

```
10     2.736358  23.929273     0.0
11     0.238965  22.494039     0.0
12      0.11048  22.597923     0.0
13    27.397846  30.032225     0.0
14    51.021995   29.65243     0.0
15     0.166339   31.66566     0.0
16    44.528594  30.876734     0.0
17  1940.307054  29.338211     0.0
18   822.743501  28.810521     0.0
19    31.211541  30.244365     0.0
20     0.180242  29.143524     0.0
```

6. Finalmente, implemente un SEM completo usando la estructura propuesta en la Pregunta 5. En particular, estime un modelo donde los factores explican la variable Response, junto con otras variables demograficas que existen en la base de datos. Ademas utilice dichas variables relevantes para explicar los factores latentes si lo considera apropiado. Las variables a incluir en el modelo final deben tener sustento teorico y el modelo final debe optimizar el ajuste a los datos, en base a los criterios vistos en clase. Que puede concluir en base a sus resultados?

R: Siguiendo la estructura de la Pregunta 5, se presentan dos modelos donde se relacionan las variables demograficas y Recency para explicar la variable Repsonse. Se permite que las variables demograficas tambien puedan afectar los factores latentes. Se estima un modelo con un factor y uno con dos factores. Los resultados muestran diferencias marginales entre ambos modelos, sin embargo los coeficientes del modelo de dos factores no son correctos, lo cual favorece al modelo de un factor.

Adicionalmente, se observa que el mejor modelo aun tiene un importante error de prediccion, segun los indices relevantes (RMSEA, CFI, TLI), lo cual puede deberse a la naturaleza binaria de la variable Response. En su interpretacion, podriamos considerar *eta1* como *intensidad de compra*, siendo una combinacion de frecuencia y volumen. La intensidad de compra crece con el ingreso pero disminuye con el numero de hijos y nivel educacional. Asimismo, la probabilidad de responder a las publicidades (Response=1) depende positivamente de la intensidad de compra y del numero de hijos, mientras que decrece con el ingreso y Recency (tiempo desde la ultima compra).

[106]:
```
df_cfa = pd.concat([df, features], axis=1, join='inner')
df_cfa = pd.concat([df_cfa, df_store['Response']], axis=1, join='inner')
df_cfa.describe()
```

[106]:
```
          MntWines    MntFruits  MntMeatProducts  MntFishProducts  \
count  2205.000000  2205.000000      2205.000000      2205.000000
mean    306.164626    26.403175       165.312018        37.756463
std     337.493839    39.784484       217.784507        54.824635
min       0.000000     0.000000         0.000000         0.000000
25%      24.000000     2.000000        16.000000         3.000000
50%     178.000000     8.000000        68.000000        12.000000
75%     507.000000    33.000000       232.000000        50.000000
max    1493.000000   199.000000      1725.000000       259.000000
```

```
       MntSweetProducts  MntGoldProds  NumDealsPurchases  NumWebPurchases  \
count       2205.000000   2205.000000        2205.000000      2205.000000
mean          27.128345     44.057143           2.318367         4.100680
std           41.130468     51.736211           1.886107         2.737424
min            0.000000      0.000000           0.000000         0.000000
25%            1.000000      9.000000           1.000000         2.000000
50%            8.000000     25.000000           2.000000         4.000000
75%           34.000000     56.000000           3.000000         6.000000
max          262.000000    321.000000          15.000000        27.000000

       NumCatalogPurchases  NumStorePurchases         Income      Kidhome  \
count          2205.000000        2205.000000    2205.000000  2205.000000
mean              2.645351           5.823583   51622.094785     0.442177
std               2.798647           3.241796   20713.063826     0.537132
min               0.000000           0.000000    1730.000000     0.000000
25%               0.000000           3.000000   35196.000000     0.000000
50%               2.000000           5.000000   51287.000000     0.000000
75%               4.000000           8.000000   68281.000000     1.000000
max              28.000000          13.000000  113734.000000     2.000000

       education_Master      Recency    Response
count       2205.000000  2205.000000  2205.00000
mean           0.165079    49.009070     0.15102
std            0.371336    28.932111     0.35815
min            0.000000     0.000000     0.00000
25%            0.000000    24.000000     0.00000
50%            0.000000    49.000000     0.00000
75%            0.000000    74.000000     0.00000
max            1.000000    99.000000     1.00000
```

[107]:

```
mod3 = """
# measurement model
eta1 =~ NumCatalogPurchases + NumStorePurchases + NumWebPurchases
eta2 =~ MntSweetProducts + MntFruits + MntGoldProds + MntMeatProducts +␣
 ↪MntWines + MntFishProducts

# regressions
Response ~ eta1 + eta2 + Recency + Income + Kidhome + education_Master
eta1 ~ Income + Kidhome
eta2 ~ eta1 + Income + Kidhome + education_Master
    """
model3 = semopy.Model(mod3)
out3=model3.fit(df_cfa)

mod4 = """
# measurement model
```

```
eta1 =~ NumCatalogPurchases + MntMeatProducts + MntWines + NumStorePurchases +␣
 ↪MntFishProducts + MntFruits + MntSweetProducts + MntGoldProds +␣
 ↪NumWebPurchases

# regressions
Response ~ eta1 + Recency + Income + Kidhome
eta1 ~ Income + Kidhome + education_Master
    """
model4 = semopy.Model(mod4)
out4=model4.fit(df_cfa)

semopy.calc_stats(model3)
```

[107]:

```
       DoF  DoF Baseline         chi2  chi2 p-value  chi2 Baseline       CFI  \
Value   74            95  2077.457114           0.0   15935.027153  0.873519

           GFI      AGFI      NFI        TLI     RMSEA        AIC         BIC  \
Value  0.86963  0.832632  0.86963  0.837626  0.110833  60.115685  236.768652

         LogLik
Value  0.942157
```

[108]:

```
model3.inspect(mode='list', what="names", std_est=True)
```

[108]:

```
                   lval  op               rval    Estimate  Est. Std  \
0                  eta1   ~             Income    0.000084  0.762670
1                  eta1   ~            Kidhome   -1.073052 -0.253249
2                  eta2   ~               eta1   15.938540  1.297938
3                  eta2   ~             Income   -0.000328 -0.242856
4                  eta2   ~            Kidhome    7.155884  0.137529
5                  eta2   ~   education_Master   -3.809386 -0.050614
6   NumCatalogPurchases   ~               eta1    1.000000  0.812977
7     NumStorePurchases   ~               eta1    1.047646  0.735502
8       NumWebPurchases   ~               eta1    0.668904  0.556204
9      MntSweetProducts   ~               eta2    1.000000  0.692821
10            MntFruits   ~               eta2    0.964941  0.691950
11          MntGoldProds   ~              eta2    1.044451  0.623624
12      MntMeatProducts   ~               eta2    6.127596  0.743577
13             MntWines   ~               eta2    9.016369  0.726070
14      MntFishProducts   ~               eta2    1.370668  0.702868
15             Response   ~               eta1    0.107789  0.684357
16             Response   ~               eta2   -0.002123 -0.165498
17             Response   ~            Recency   -0.002554 -0.206150
18             Response   ~             Income   -0.000004 -0.208832
19             Response   ~            Kidhome    0.111146  0.166545
20             Response   ~   education_Master   -0.011955 -0.012384
21                 eta2  ~~               eta2    1.308827  0.001676
```

22	eta1	~~	eta1	0.770445	0.148809
23	NumCatalogPurchases	~~	NumCatalogPurchases	2.656081	0.339068
24	MntFruits	~~	MntFruits	791.337561	0.521205
25	MntFishProducts	~~	MntFishProducts	1502.270611	0.505976
26	NumWebPurchases	~~	NumWebPurchases	5.171563	0.690637
27	Response	~~	Response	0.114804	0.893846
28	MntGoldProds	~~	MntGoldProds	1338.257662	0.611093
29	MntWines	~~	MntWines	56925.213068	0.472822
30	MntMeatProducts	~~	MntMeatProducts	23704.272982	0.447093
31	MntSweetProducts	~~	MntSweetProducts	845.787886	0.519999
32	NumStorePurchases	~~	NumStorePurchases	4.821912	0.459036

	Std. Err	z-value	p-value
0	0.000002	41.590226	0.0
1	0.070662	-15.185748	0.0
2	1.425787	11.178764	0.0
3	0.000118	-2.772412	0.005564
4	1.813481	3.945938	0.000079
5	0.93531	-4.072859	0.000046
6	-	-	-
7	0.026826	39.052803	0.0
8	0.024267	27.563856	0.0
9	-	-	-
10	0.031895	30.253497	0.0
11	0.038103	27.411375	0.0
12	0.189335	32.363816	0.0
13	0.284857	31.652304	0.0
14	0.044643	30.702721	0.0
15	2.384728	0.0452	0.963948
16	0.148846	-0.014261	0.988622
17	0.000251	-10.170146	0.0
18	0.00005	-0.072549	0.942165
19	1.078524	0.103054	0.91792
20	0.567363	-0.021071	0.983189
21	18.066195	0.072446	0.942247
22	0.078886	9.76661	0.0
23	0.103915	25.560022	0.0
24	26.091834	30.328936	0.0
25	49.841841	30.140753	0.0
26	0.160723	32.176808	0.0
27	0.004007	28.652473	0.0
28	42.844621	31.235138	0.0
29	1917.635871	29.6851	0.0
30	809.570134	29.280073	0.0
31	27.900467	30.314471	0.0
32	0.163684	29.458682	0.0

[109]:

```
semopy.calc_stats(model4)
```

[109]:

```
       DoF  DoF Baseline         chi2  chi2 p-value  chi2 Baseline       CFI  \
Value   79            95  2144.235044           0.0   15935.027153  0.869619

            GFI      AGFI       NFI       TLI     RMSEA        AIC  \
Value  0.865439  0.838186  0.865439  0.843213  0.108909  50.055116

              BIC    LogLik
Value  198.215668  0.972442
```

[110]:

```
model4.inspect(mode='list', what="names", std_est=True)
```

[110]:

```
                  lval  op                 rval      Estimate  Est. Std  \
0                 eta1   ~               Income      0.000084  0.755056
1                 eta1   ~              Kidhome     -0.927183 -0.215658
2                 eta1   ~     education_Master     -0.246102 -0.039573
3   NumCatalogPurchases   ~                 eta1      1.000000  0.825750
4       MntMeatProducts   ~                 eta1     73.868689  0.742276
5              MntWines   ~                 eta1    111.569640  0.733645
6     NumStorePurchases   ~                 eta1      1.030893  0.734352
7       MntFishProducts   ~                 eta1     16.233881  0.695108
8             MntFruits   ~                 eta1     11.404750  0.683068
9      MntSweetProducts   ~                 eta1     11.857536  0.685201
10          MntGoldProds   ~                 eta1     12.460519  0.617811
11      NumWebPurchases   ~                 eta1      0.665228  0.561173
12             Response   ~                 eta1      0.061148  0.393914
13             Response   ~              Recency     -0.002559 -0.206515
14             Response   ~               Income     -0.000002 -0.107546
15             Response   ~              Kidhome      0.073538  0.110188
16                 eta1  ~~                 eta1      1.118095  0.209756
17  NumCatalogPurchases  ~~  NumCatalogPurchases      2.487033  0.318137
18            MntFruits  ~~            MntFruits    792.637949  0.533418
19      MntFishProducts  ~~      MntFishProducts   1502.611301  0.516824
20      NumWebPurchases  ~~      NumWebPurchases      5.131654  0.685085
21             Response  ~~             Response      0.114820  0.893917
22             MntWines  ~~             MntWines  56925.194867  0.461765
23         MntGoldProds  ~~         MntGoldProds   1340.699116  0.618310
24      MntMeatProducts  ~~      MntMeatProducts  23704.194816  0.449026
25     MntSweetProducts  ~~     MntSweetProducts    846.840144  0.530499
26    NumStorePurchases  ~~    NumStorePurchases      4.839775  0.460726

    Std. Err    z-value   p-value
0   0.000002  43.403761       0.0
1   0.061988 -14.957427       0.0
2   0.074258  -3.314137  0.000919
3          -          -         -
```

4	1.865846	39.58992	0.0
5	2.863827	38.958238	0.0
6	0.026427	39.009694	0.0
7	0.448059	36.231539	0.0
8	0.322086	35.409047	0.0
9	0.33351	35.553792	0.0
10	0.399823	31.165061	0.0
11	0.023987	27.733428	0.0
12	0.008471	7.218445	0.0
13	0.000251	-10.192787	0.0
14	0.000001	-2.267452	0.023363
15	0.017839	4.122268	0.000038
16	0.05947	18.800924	0.0
17	0.08947	27.797535	0.0
18	25.550305	31.022641	0.0
19	48.673494	30.871244	0.0
20	0.160054	32.061942	0.0
21	0.003521	32.608093	0.0
22	1879.380363	30.289342	0.0
23	42.335907	31.668133	0.0
24	786.627204	30.133963	0.0
25	27.320322	30.996712	0.0
26	0.15985	30.277007	0.0

[111]:
```
semopy.semplot(mod4, "semtarea3.png")
```

[111]: