Semana 4 - Actividad 1

Ciencia y Analítica de datos

Profesor Titular: Jobish Vallikavungal Devassia

Maestría en Inteligencia Artificial Aplicada (MNA-V)

11/10/2022

Equipo 24

Victor Hugo Avila Felipe - A01794425

Andrés Eduardo Figueroa García - A01378536

Glosario

0.- Instrucciones

1.- Parte 1: Ejercicio guiado

Paso 1: Determine el número mínimo de componentes principales que representan la mayor parte de la variación en sus datos

Paso 2: Interprete cada componente principal en términos de las variables originales

Paso 3: Identifique valores atípicos

2.- Parte 2: Responde las siguientes preguntas en una celda de texto en Jupyter Notebook

0.- Instrucciones

Parte 1: Ejercicio guiado

Revise el ejercicio guiado para el análisis de componentes principales utilizando el conjunto de datos.

Paso 1: Determine el número mínimo de componentes principales que representan la mayor parte de la variación en sus datos

- Utilice la proporción acumulada de la varianza que explican los componentes para determinar la cantidad de varianza que explican los componentes principales.

Paso 2: Interprete cada componente principal en términos de las variables originales

- Examine la magnitud y la dirección de los coeficientes de las variables originales.
  Nota: Cuanto mayor sea el valor absoluto del coeficiente, más importante será la variable correspondiente en el cálculo del componente.

Paso 3: Identifique valores atípicos

- Realice alguna gráfica de valores atípicos o boxplot para identificar los valores atípicos. Cualquier punto que esté más alejado de la línea de referencia es un valor atípico.

Parte 2: Responde las siguientes preguntas en una celda de texto en Jupyter Notebook

1. ¿Cuál es el número de componentes mínimo y por qué?
2. ¿Cuál es la variación de los datos que representan esos componentes?
3. ¿Cuál es la pérdida de información después de realizar PCA?
4. De las variables originales, ¿Cuál tiene mayor y cuál tiene menor importancia en los componentes principales?
5. ¿Cuándo se recomienda realizar un PCA y qué beneficios ofrece para Machine Learning?

1.- Parte 1

Se importan las librerías a utlizar en el ejercicio

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns

from sklearn.decomposition import PCA
from sklearn import preprocessing
import matplotlib.pyplot as plt

pd.set_option('max_columns', None)
```

Se importa la Dataframe a utilizar en el análisis.

In [2]:
```
mypath = "https://raw.githubusercontent.com/PosgradoMNA/Actividades_Aprendizaje-/main/d

df = pd.read_csv(mypath, index_col=0)
df.index.name = None
```

Se eliminarán las columnas que contengan valores NULL y la variable "Y"

In [3]:
```
ndf = df.dropna(axis=0)
ndf = ndf.drop(["Y"], axis=1)
```

In [4]:
```
# ndf.head()
```

In [5]:
```
ndf.describe()
```

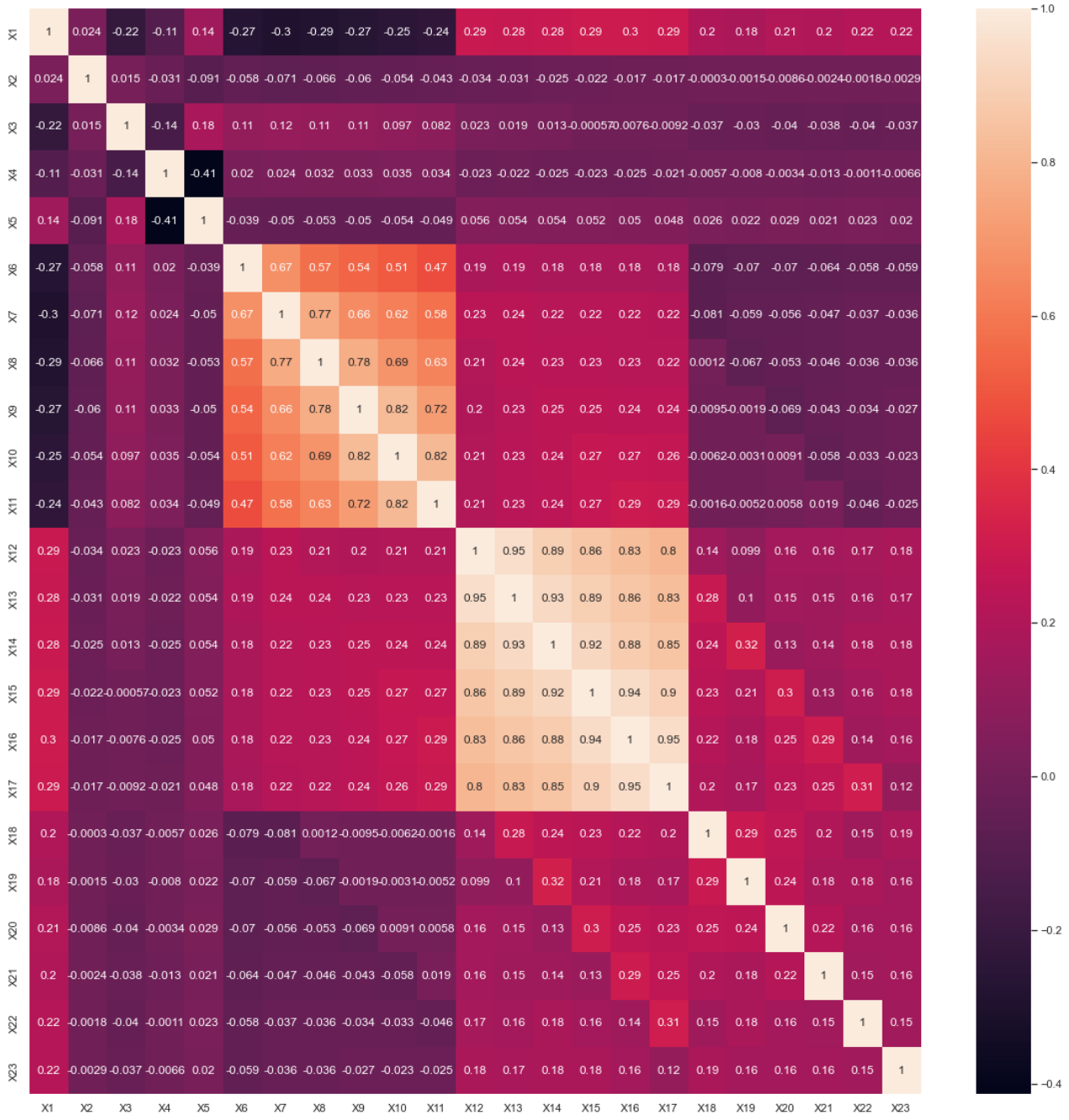
Out[5]:

	X1	X2	X3	X4	X5	X6	
count	29958.000000	29958.000000	29958.000000	29958.000000	29958.000000	29958.000000	29958.000
mean	167555.900928	1.604012	1.853094	1.551739	35.483443	-0.017124	-0.134
std	129737.299088	0.489070	0.790471	0.521952	9.214319	1.123989	1.197
min	10000.000000	1.000000	0.000000	0.000000	21.000000	-2.000000	-2.000
25%	50000.000000	1.000000	1.000000	1.000000	28.000000	-1.000000	-1.000
50%	140000.000000	2.000000	2.000000	2.000000	34.000000	0.000000	0.000
75%	240000.000000	2.000000	2.000000	2.000000	41.000000	0.000000	0.000
max	1000000.000000	2.000000	6.000000	3.000000	79.000000	8.000000	8.000

In [6]:
```
# ndf.dtypes
```

Se realizará una tabla de correlación entre las variables de la tabla para visuaizar cuáles variables con las que podrían formar parte del mismo Component

In [7]:
```
sns.set(rc={'figure.figsize':(20,20)})
CorrM = ndf.corr(method ='pearson')
sns.heatmap(CorrM, annot=True)
plt.show()
```



Se observa que las variabels entre X6 y X11 muestran una fuerte correlación positiva, al igual que X12 a X17. Se puede apreciar de igual forma una alta correlación negativa entre X4 y X5.

Paso 1: Determine el número mínimo de componentes principales que representan la mayor parte de la variación en sus datos

Se normalizan los datos para evitar que unos tengan mayor influencia que otros por la magnitud que presentan. Se escogió como opción de normalización MinMax, ya nos dio mejores resultados en la generación del PCA.

In [8]:

```
scaler = preprocessing.MinMaxScaler()
# scaler = preprocessing.StandardScaler()
scaler.fit(ndf)
sdf = pd.DataFrame(scaler.fit_transform(ndf.values), columns=ndf.columns, index=ndf.ind

pcs = PCA()
```

```
pcs.fit(sdf)
```

Out[8]:
```
PCA()
```

Se obtiene una tabal que resume las principales estadísticas para análisis de PCA.

In [9]:
```
pcsSummary_df = pd.DataFrame({'Standard deviation': np.sqrt(pcs.explained_variance_),
                              'Proportion of variance': pcs.explained_variance_ratio_,
                              'Cumulative proportion': np.cumsum(pcs.explained_variance_
                              })

pcsSummary_df = pcsSummary_df.transpose()
pcsSummary_df.columns = ['PC{}'.format(i) for i in range(1, len(pcsSummary_df.columns)
pcsSummary_df.round(4)
```

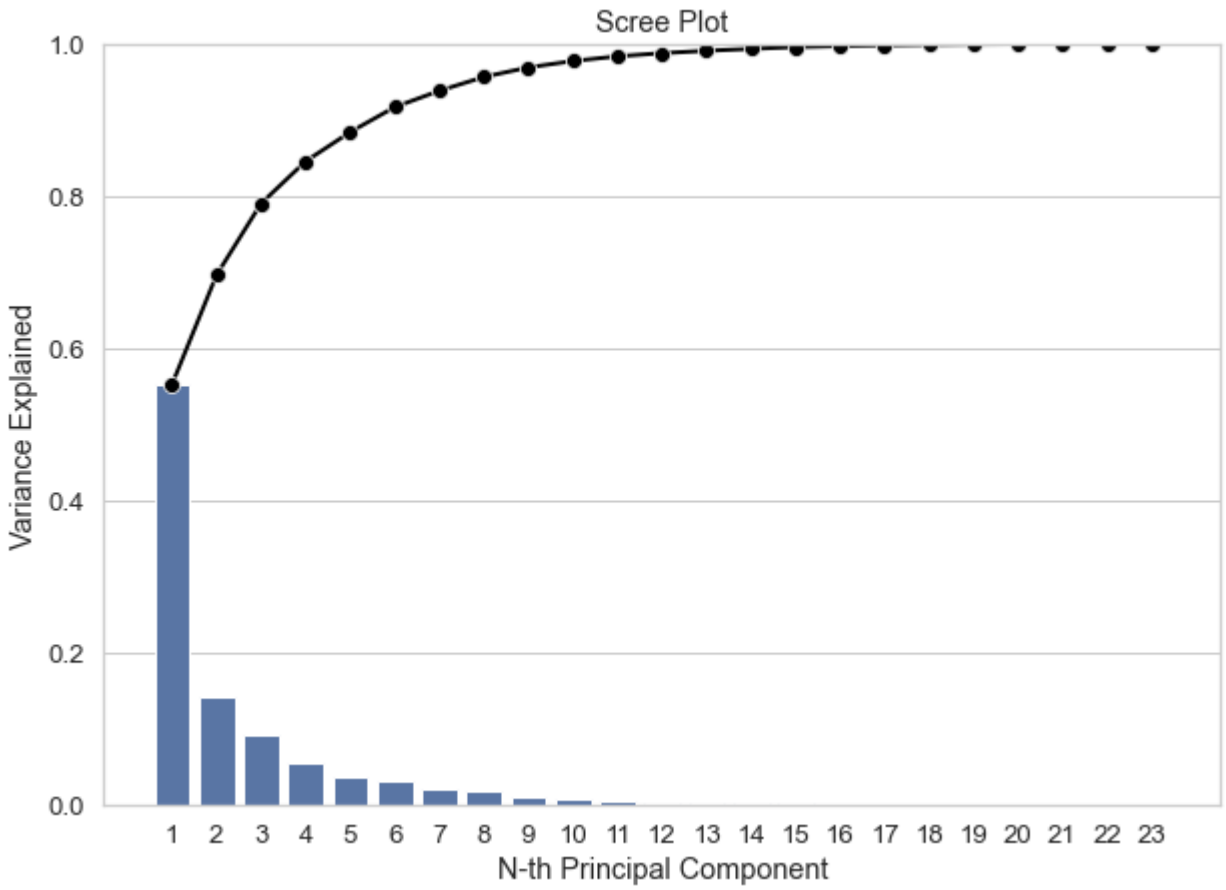
Out[9]:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12
Standard deviation	0.4898	0.2496	0.2018	0.1548	0.1290	0.1192	0.0963	0.0887	0.0738	0.0602	0.0519	0.0425
Proportion of variance	0.5529	0.1436	0.0939	0.0552	0.0384	0.0327	0.0214	0.0181	0.0126	0.0084	0.0062	0.0042
Cumulative proportion	0.5529	0.6964	0.7903	0.8455	0.8839	0.9167	0.9381	0.9562	0.9688	0.9771	0.9833	0.9875

Ya que se obtuvieron las proporciones acumulativas, se puede ver que el PC6 junta más de un 90%. Además, se incluye el siguiente Scree Plot para tener una representación gráfica.

In [10]:
```
# Scree Plot
PC_components = np.arange(pcs.n_components_) + 1

_ = sns.set(style = 'whitegrid',
            font_scale = 1.2
            )

fig, ax = plt.subplots(figsize=(10, 7))

_ = sns.barplot(x = PC_components,
                y = pcs.explained_variance_ratio_,
                color = 'b'
                )

_ = sns.lineplot(x = PC_components-1,
                 y = np.cumsum(pcs.explained_variance_ratio_),
                 color = 'black',
                 linestyle = '-',
                 linewidth = 2,
                 marker = 'o',
                 markersize = 8
                 )

plt.title('Scree Plot')
plt.xlabel('N-th Principal Component')
plt.ylabel('Variance Explained')
```

```
plt.ylim(0, 1)
plt.show()
```



Se muestra una tabla con los componentes principales que abarcan al menos el 90%, que sería desde PC1 hasta PC 6.

In [11]:
```
pcsComponents_df = pd.DataFrame(pcs.components_.transpose(),
                                columns=pcsSummary_df.columns,
                                index=ndf.columns
                                )
pcsComponents_df.iloc[:,:6]
# pcsComponents_df
```

Out[11]:

	PC1	PC2	PC3	PC4	PC5	PC6
X1	-0.008438	-0.208182	0.106591	0.637467	0.021764	0.059880
X2	-0.998288	0.044411	0.015108	0.005377	0.020535	-0.026461
X3	-0.002925	0.089291	0.201336	-0.377353	0.546826	0.662243
X4	0.011460	0.077536	-0.745903	0.101053	0.605515	-0.241777
X5	0.031781	-0.092906	0.605882	0.001958	0.533354	-0.575325
X6	0.017170	0.322965	0.037268	-0.024355	-0.023267	-0.032892
X7	0.022178	0.402770	0.046181	-0.005379	-0.029863	-0.035579
X8	0.021050	0.416770	0.043913	0.010324	-0.035509	-0.064012
X9	0.019120	0.411812	0.045270	0.032573	-0.035984	-0.079501

	PC1	PC2	PC3	PC4	PC5	PC6
X10	0.017178	0.391908	0.041108	0.052998	-0.036196	-0.074644
X11	0.014573	0.376123	0.039933	0.065327	-0.033356	-0.074281
X12	0.005522	0.070808	0.054126	0.295471	0.101859	0.184153
X13	0.005447	0.078923	0.056330	0.311315	0.104853	0.189668
X14	0.002554	0.044815	0.031988	0.175540	0.056319	0.103387
X15	0.003748	0.073440	0.050550	0.286582	0.086078	0.160013
X16	0.003136	0.073470	0.050022	0.283918	0.080843	0.153849
X17	0.002335	0.054636	0.036454	0.210599	0.060135	0.112420
X18	-0.000003	-0.002702	0.003659	0.031543	0.007581	0.012124
X19	0.000022	-0.002492	0.002201	0.018442	0.003989	0.006830
X20	0.000317	-0.004065	0.003583	0.032827	0.008028	0.011852
X21	0.000077	-0.005363	0.004652	0.039119	0.007191	0.015500
X22	0.000071	-0.008002	0.005462	0.057155	0.012910	0.020088
X23	0.000150	-0.006685	0.005625	0.052878	0.010061	0.019355

Paso 2: Interprete cada componente principal en términos de las variables originales

Se muestran las variables que forman parte de PC1 hasta PC4, donde es importante mencionar que X1 es compun para todos los PC# mostrados, a excepción de PC1.

PC1: Este PC coincide con la predicción realizada a partir de las correlaciones de las variables. Se puede observer en la gráfica con Heatmap que se muestra al inicio del programa. Se esperaba que las columnas X6 a X11 estuvieran fuertemente relacionadas.

In [12]:
```
pcsComponents_df.PC1.abs().nlargest(8)
```

Out[12]:
```
X2     0.998288
X5     0.031781
X7     0.022178
X8     0.021050
X9     0.019120
X10    0.017178
X6     0.017170
X11    0.014573
Name: PC1, dtype: float64
```

In [13]:
```
pcsComponents_df.PC1.abs().idxmax()
pcsComponents_df.PC1.abs().max()
print("La variable con mayor impacto es {} con un valor de {}".format(pcsComponents_df.
```

```
La variable con mayor impacto es X2 con un valor de 0.9982884967976403
```

In [14]:
```
pcsComponents_df.PC2.abs().idxmin()
```

```
pcsComponents_df.PC2.abs().min()
print("La variable con menor impacto es {} con un valor de {}".format(pcsComponents_df.
```

```
La variable con menor impacto es X19 con un valor de 0.0024916000144794524
```

PC2: Este PC coincide, de igual manera, con la predicción realizada a partir de las correlaciones de las variables. Es igual al PC1, pero se cambia el X2 por X1. Se esperaba que las columnas X6 a X11 estuvieran fuertemente relacionadas.

In [15]:
```
pcsComponents_df.PC2.abs().nlargest(7)
```

Out[15]:
```
X8     0.416770
X9     0.411812
X7     0.402770
X10    0.391908
X11    0.376123
X6     0.322965
X1     0.208182
Name: PC2, dtype: float64
```

PC3: Este PC muestra elementos que tiene una correlación media, entre parámetros X1, X3 y X5, mientras que X4 muestra una alta correlación con X5. Por tal motivo, las variables se relacionan de manera indirecta.

In [16]:
```
pcsComponents_df.PC3.abs().nlargest(4)
```

Out[16]:
```
X4    0.745903
X5    0.605882
X3    0.201336
X1    0.106591
Name: PC3, dtype: float64
```

PC4: Este PC coincide con la segunda predicción realizada a partir de las correlaciones de las variables, donde se esperaba que las columnas X12 a X17 estuvieran fuertemente relacionadas.

In [17]:
```
pcsComponents_df.PC4.abs().nlargest(7)
```

Out[17]:
```
X1     0.637467
X3     0.377353
X13    0.311315
X12    0.295471
X15    0.286582
X16    0.283918
X17    0.210599
Name: PC4, dtype: float64
```

PC5: Este PC tiene como variables principales, las mismas que se muestran en PC3, por lo que se puede hacer una extensión del análisis realizado previamente en ese PC.

In [18]:
```
pcsComponents_df.PC5.abs().nlargest(3)
```

Out[18]:
```
X4    0.605515
X3    0.546826
X5    0.533354
Name: PC5, dtype: float64
```

PC6: Este PC coincide tiene una aportación de la varianza cercano al 3%, por lo que un análisis realizado con base en los datos originales no es tan evidente. Sin embargo, se ve como una combinación del análisis de PC4 y PC5 de manera simultánea.

In [19]:
```
pcsComponents_df.PC6.abs().nlargest(11)
```

Out[19]:
```
X3     0.662243
X5     0.575325
X4     0.241777
X13    0.189668
X12    0.184153
X15    0.160013
X16    0.153849
X17    0.112420
X14    0.103387
X9     0.079501
X10    0.074644
Name: PC6, dtype: float64
```

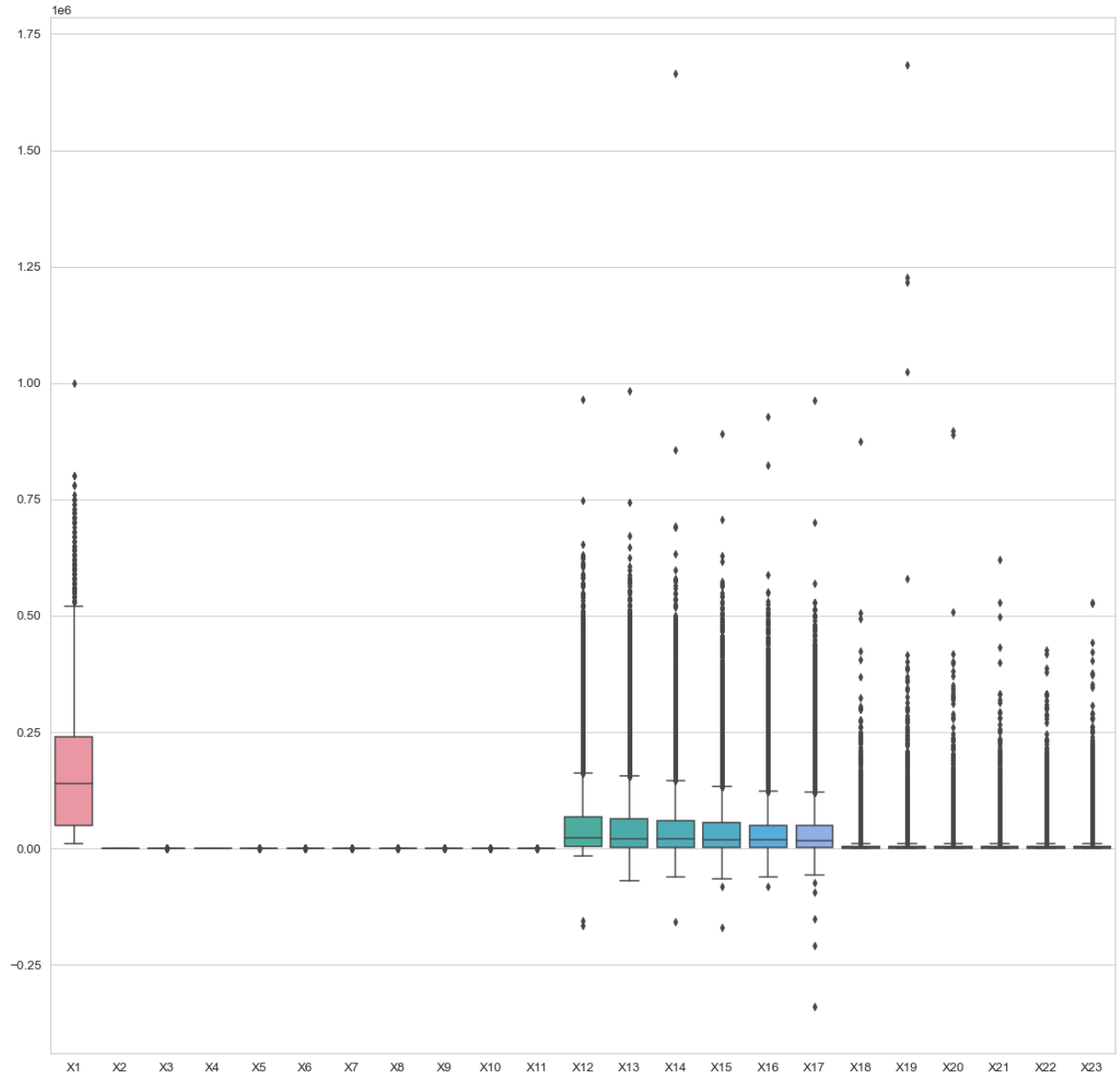
Paso 3: Identifique valores atípicos

Para la búsqueda de valores atípicos se realizarán dos gráficas de boxplot. La primera se hará con los datos originarles, mientras que la segunda se hará con los valores escalados de tal manera que se puedan apreciar de mejor manera los valores atípicos para todas las variables sin importar la escala.

De la gráfica de boxplot antes del escalamiento se pueden apreciar ciertos valores atípicos, como lo son el préstamo de 1 millón para X1, así com algún bill entre X12 y X17.

In [20]:
```
sns.boxplot(data=ndf)
```

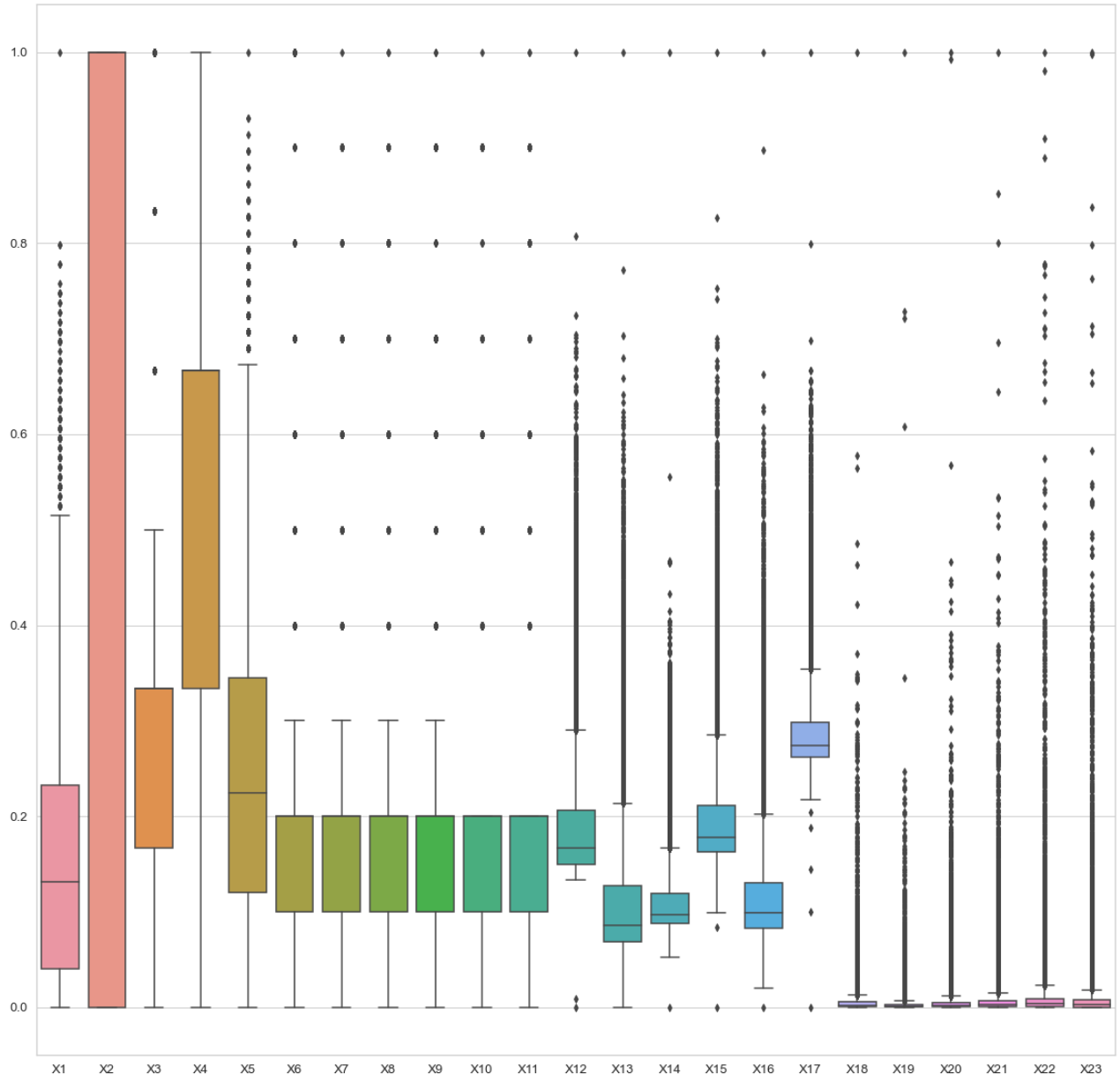
Out[20]:
```
<AxesSubplot:>
```



A continuación, se muestran el boxplot para los valores escalados. Se puede apreciar que prácticamente todas las columnas, menos X2 (género) y X3 (educación) tiene valores atípicos con base en los datos estadísticos de las mismas.

In [21]:
```
sns.boxplot(data=sdf)
```

Out[21]:
```
<AxesSubplot:>
```



2.- Parte 2

1. ¿Cuál es el número de componentes mínimo y por qué?

Basados en el análisis que se realizó y la discusión realizada en equipo, se llegó a la conclusión que el número mínimo de componentes a tener en cuenta serían 6. La razón es que un número menor a este nos daría un valor de varianza acumulado menor a 90%, lo cual indicaría que estamos perdiendo información.

2. ¿Cuál es la variación de los datos que representan esos componentes?

La variación total de los datos es de 0.9167 para el conjunto de componentes entre PC1 y PC6, que son los mínimos reportados en la pregunta anterior.

3. ¿Cuál es la pérdida de información después de realizar PCA?

Dado que la información a tener en cuenta se da por el acumulado de la varianza, los datos que se dejan de considerar estarían dados por el complemento de este porcentaje. Para el caso donde sólo

se toman los primero 6 PC, sería una pérdida de 8.33%.

4. De las variables originales, ¿Cuál tiene mayor y cuál tiene menor importancia en los componentes principales?

Para el PC1, se identificó que la variable con mayor impacto sería X2 y la de menor impacto sería X19. Se incluye a continuación, el resultado del print que se corrión para obtener esta información.

La variable con mayor impacto es X2 con un valor de 0.9982884967976403 La variable con menor impacto es X19 con un valor de 0.0024916000144794524

5. ¿Cuándo se recomienda realizar un PCA y qué beneficios ofrece para Machine Learning?

El PCA como un método de reducción de dimensionalidad, se recomienda cuando tienes una gran cantidad de datos, siendo que no todos estos o representan de manera favorable los atributos que se quieren estudiar. Funciona dejando aquellos que tienen un mayor impacto, por lo que en el caso de Machine Learning, se vería beneficiado un modelo cuyos datos pasen por esta clase de pre-procesamiento, en cuanto a que al reducir las variables a tener en cuenta, se puede obtener un modelo más simplificado.