```
# Cargue de Librerías básicas
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Importar tensorflow
import tensorflow as tf
print("TF version   : ", tf.__version__)

# Necesitaremos GPU
print("GPU available: ", tf.config.list_physical_devices('GPU'))

# keras version is 2.11.0
import keras
print("Keras version   : ", keras.__version__)


```

```
TF version   :  2.15.0
GPU available:  []
Keras version   :  2.15.0
```

```
#-----------------------------------------#
#       debido a que estoy usando COLAB      #
#-----------------------------------------#

from google.colab import drive
drive.mount('/content/drive') #/content/drive/MyDrive/pec2/data/xl.pickle
print("GPU available: ", tf.config.list_physical_devices('GPU'))
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
GPU available:  []
```

```
import pandas as pd

home =  '/content/drive/MyDrive/TFM/'

file_path = home + "2017_2023DSTrabajo.xlsx"

dsXls = pd.read_excel(file_path)
dsXls.head(5)
dsXls.info()
```

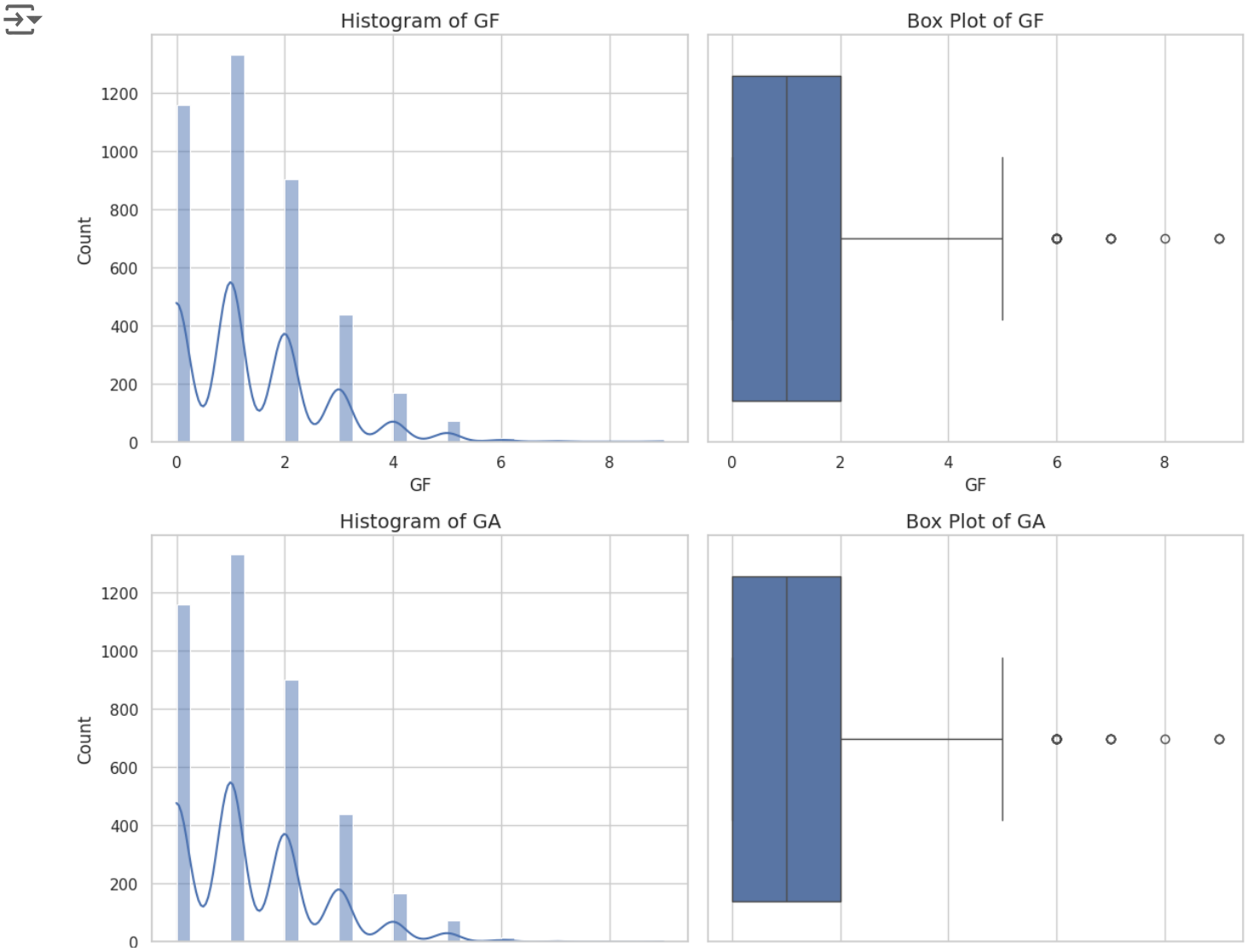
```
10
11  ####################################
12  # LIMPIEZA DE DATOS
13  ####################################
14  #1. validar duplicados
15  dsXls.nunique()
16
17  #2. validar nulos, rellenar valores faltantes con la mediana
18  #dsXls.isnull().sum()
19  dsXls['Dist'].fillna(dsXls['Dist'].median(), inplace=True)
20  dsXls['Attendance'].fillna(dsXls['Attendance'].median(), inplace=True)
21  dsXls.isnull().sum()
22
23
24  ####################################
25  # ESTADISTICAS
26  ####################################
27  #dsXls.describe().T
28  dsXls.iloc[:,1:].describe()
```

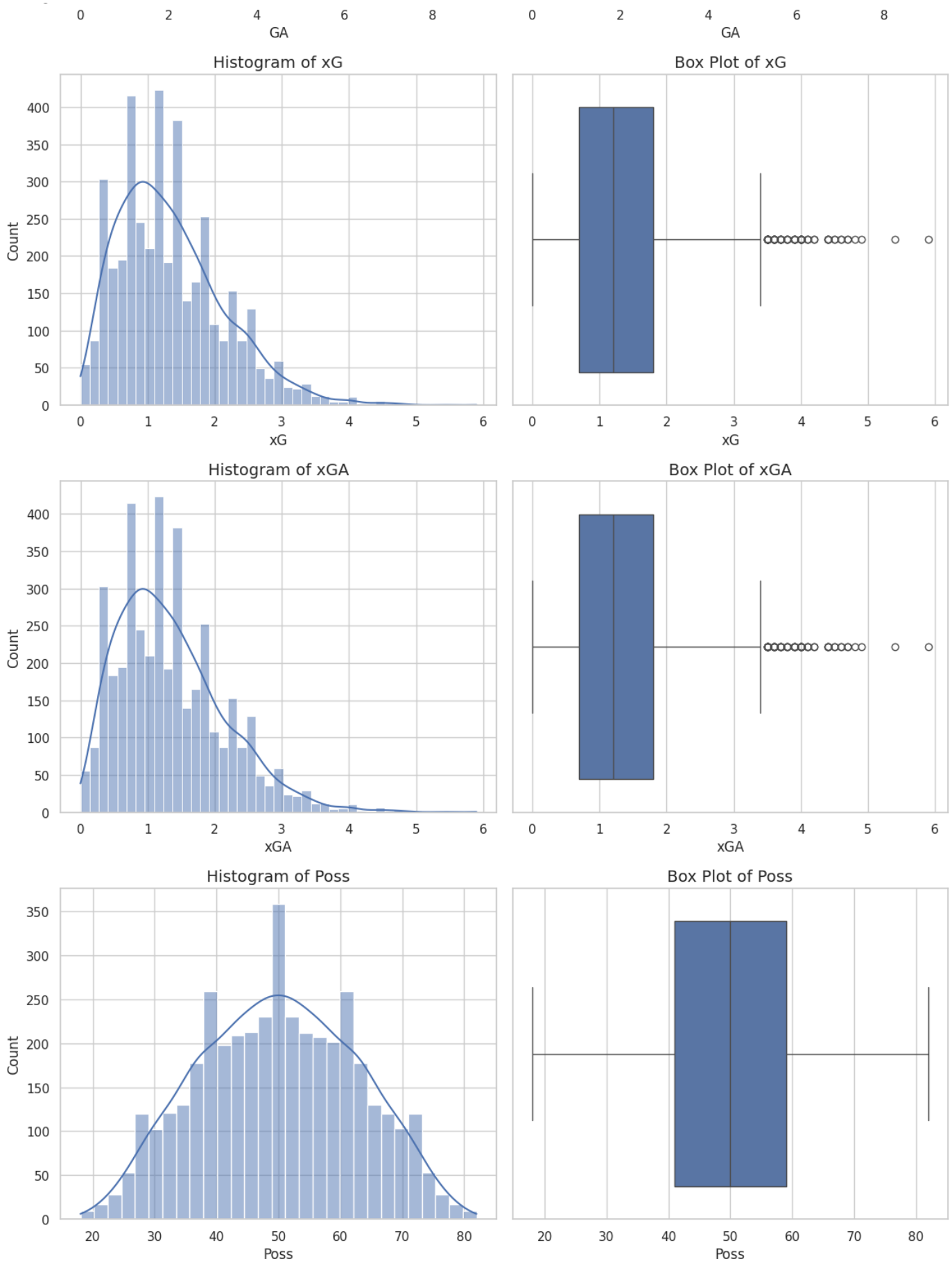
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4092 entries, 0 to 4091
Data columns (total 21 columns):
 #   Column      Non-Null Count  Dtype         
---  ------      --------------  -----         
 0   Date        4092 non-null   datetime64[ns]
 1   Round       4092 non-null   object        
 2   Day         4092 non-null   object        
 3   Venue       4092 non-null   object        
 4   Result      4092 non-null   object        
 5   GF          4092 non-null   float64       
 6   GA          4092 non-null   float64       
 7   Opponent    4092 non-null   object        
 8   xG          4092 non-null   float64       
 9   xGA         4092 non-null   float64       
 10  Poss        4092 non-null   float64       
 11  Attendance  3212 non-null   float64       
 12  Season      4092 non-null   int64         
 13  Team        4092 non-null   object        
 14  Sh          4092 non-null   float64       
 15  SoT         4092 non-null   float64       
 16  Dist        4089 non-null   float64       
 17  SCA         4092 non-null   float64       
 18  KP          4092 non-null   float64       
 19  PPA         4092 non-null   float64       
 20  CrsPA       4092 non-null   float64       
dtypes: datetime64[ns](1), float64(13), int64(1), object(6)
memory usage: 671.5+ KB
```

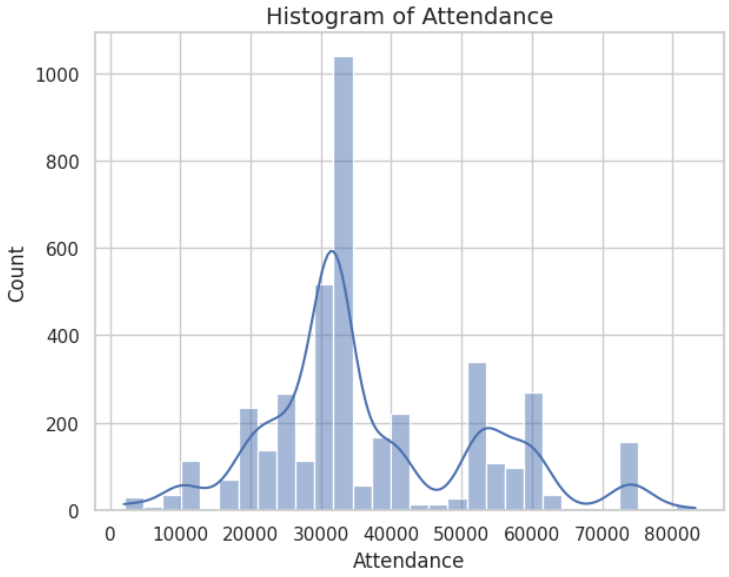
	GF	GA	xG	xGA	Poss	Attendance	
count	4092.000000	4092.000000	4092.000000	4092.000000	4092.000000	4092.000000	4(
mean	1.377810	1.377810	1.346163	1.346163	50.001222	36912.650049	2(
std	1.277631	1.277631	0.796551	0.796551	12.726702	15301.262664	
min	0.000000	0.000000	0.000000	0.000000	18.000000	2000.000000	2(
25%	0.000000	0.000000	0.700000	0.700000	41.000000	29296.000000	2(
50%	1.000000	1.000000	1.200000	1.200000	50.000000	32092.500000	2(
75%	2.000000	2.000000	1.800000	1.800000	59.000000	51237.000000	2(
max	9.000000	9.000000	5.900000	5.900000	82.000000	83222.000000	2(

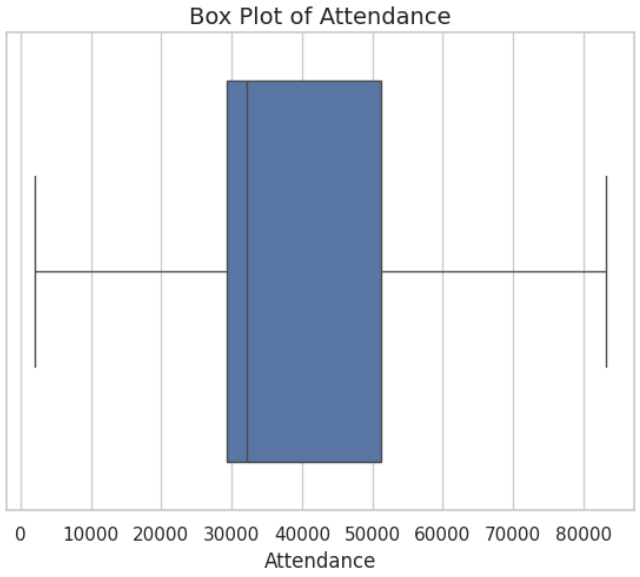
```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
```

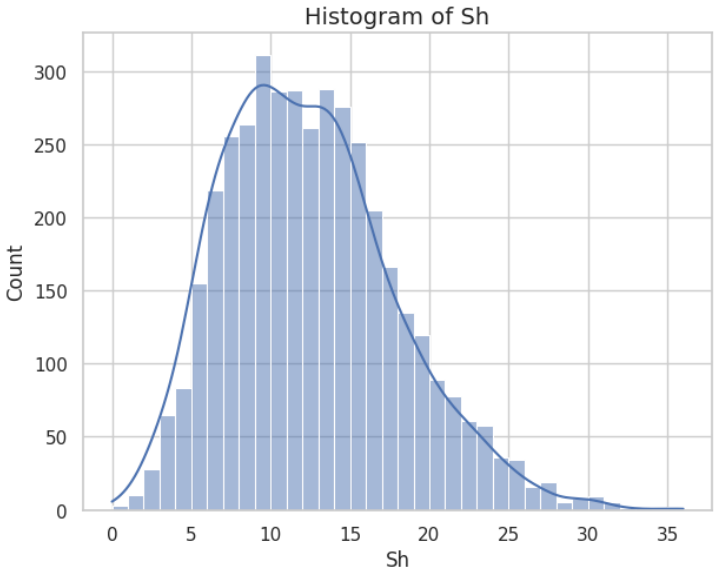
```
 3
 4 # Configuración de los estilos de los gráficos
 5 sns.set(style="whitegrid")
 6
 7 # Lista de columnas numéricas para visualizar
 8 numeric_columns = ['GF', 'GA', 'xG', 'xGA', 'Poss', 'Attendance', 'Sh', 'SoT',
 9
10 # Creación de histogramas y diagramas de caja para cada variable numérica
11 fig, axes = plt.subplots(len(numeric_columns), 2, figsize=(12, 5 * len(numeric
12 for i, col in enumerate(numeric_columns):
13     sns.histplot(dsXls[col], kde=True, ax=axes[i, 0])
14     axes[i, 0].set_title(f'Histogram of {col}', fontsize=14)
15     sns.boxplot(x=dsXls[col], ax=axes[i, 1])
16     axes[i, 1].set_title(f'Box Plot of {col}', fontsize=14)
17
18 plt.tight_layout()
19 plt.show()
```

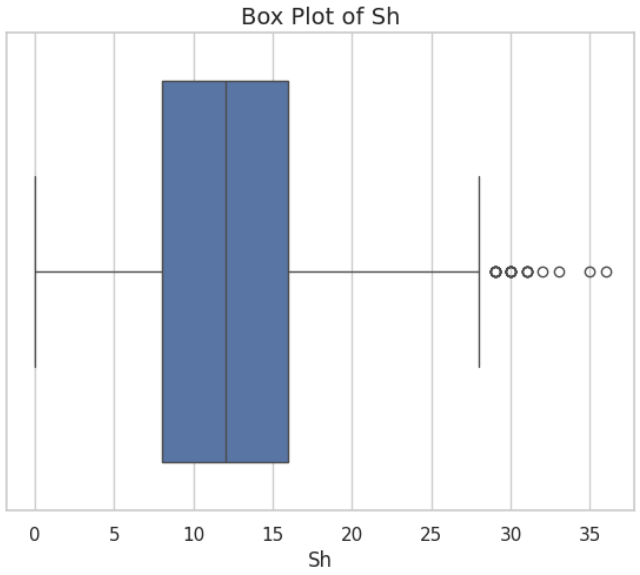


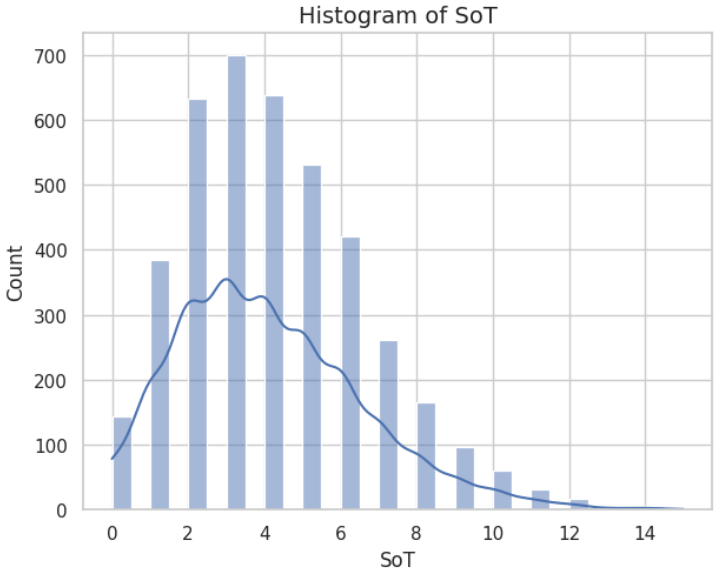

GA
GA
Histogram of xG
Box Plot of xG
Count
xG
xG
Histogram of xGA
Box Plot of xGA
Count
xGA
xGA
Histogram of Poss
Box Plot of Poss
Count
Poss
Poss

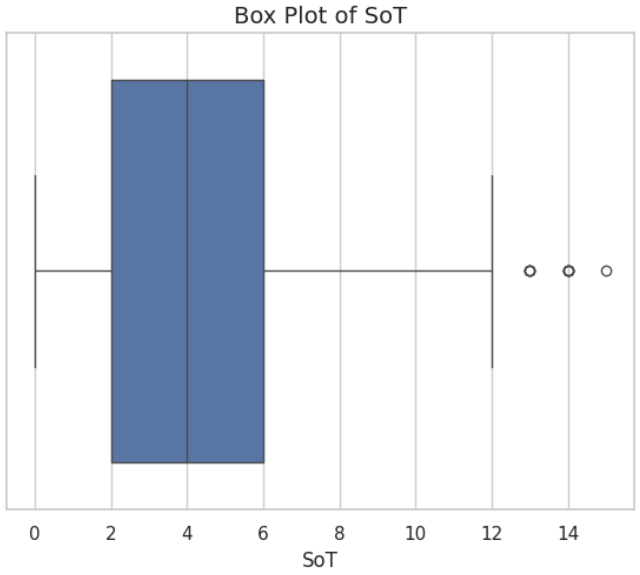

Histogram of Attendance
Count
Attendance
0
200
400
600
800
1000
0
10000
20000
30000
40000
50000
60000
70000
80000


Box Plot of Attendance
Attendance
0
10000
20000
30000
40000
50000
60000
70000
80000


Histogram of Sh
Count
Sh
0
50
100
150
200
250
300
0
5
10
15
20
25
30
35

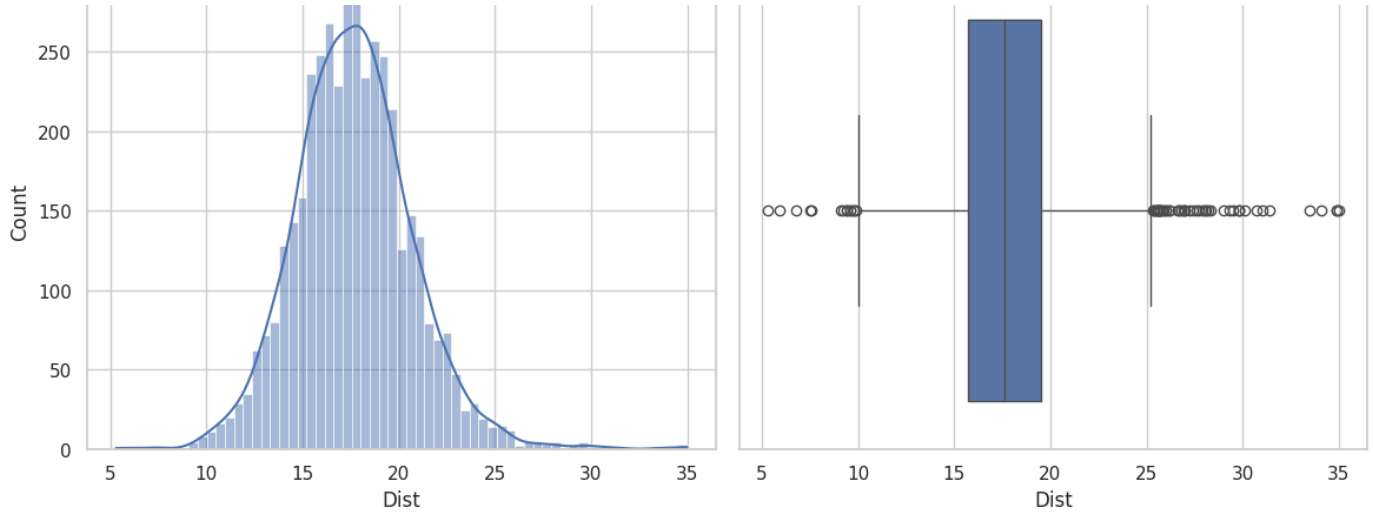

Box Plot of Sh
Sh
0
5
10
15
20
25
30
35

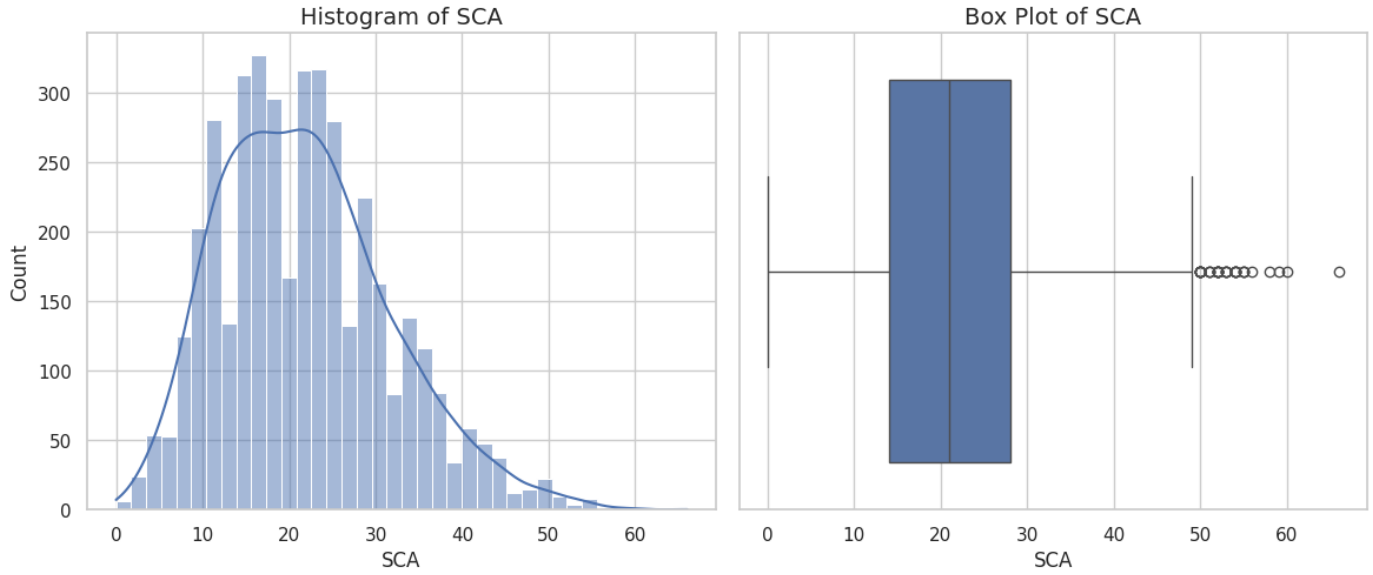

Histogram of SoT
Count
SoT
0
100
200
300
400
500
600
700
0
2
4
6
8
10
12
14

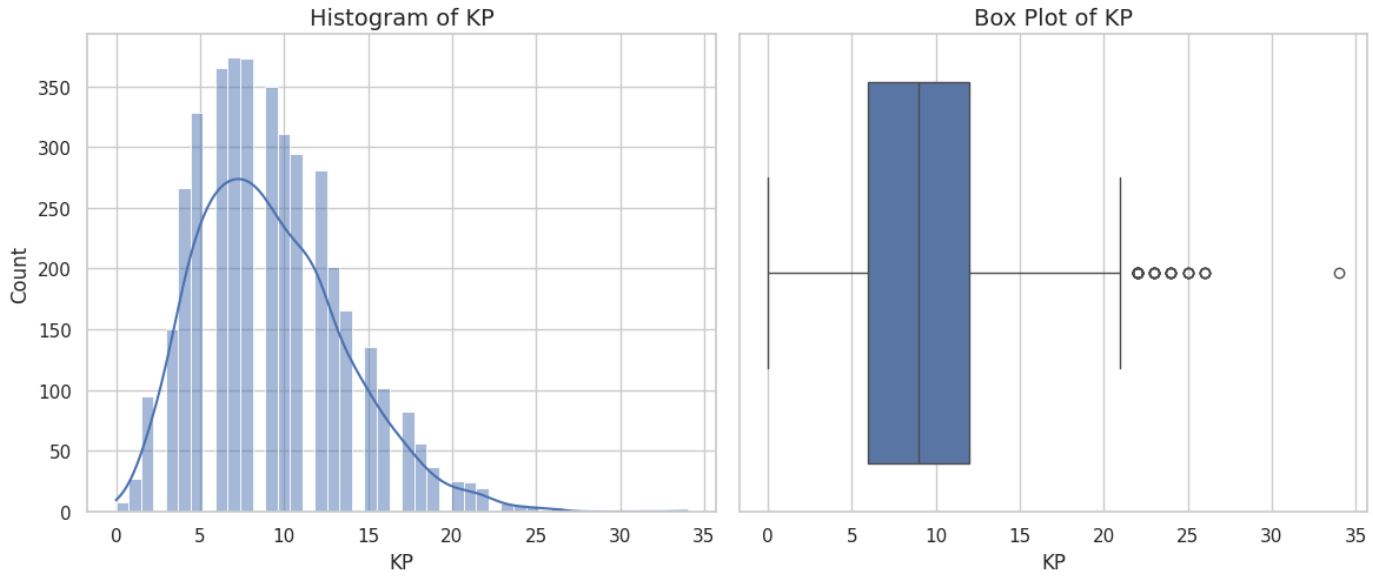

Box Plot of SoT
SoT
0
2
4
6
8
10
12
14

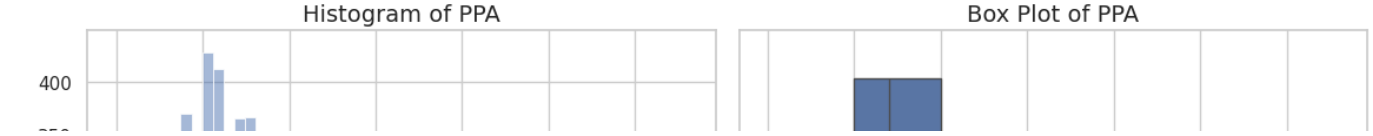

Histogram of Dist
300

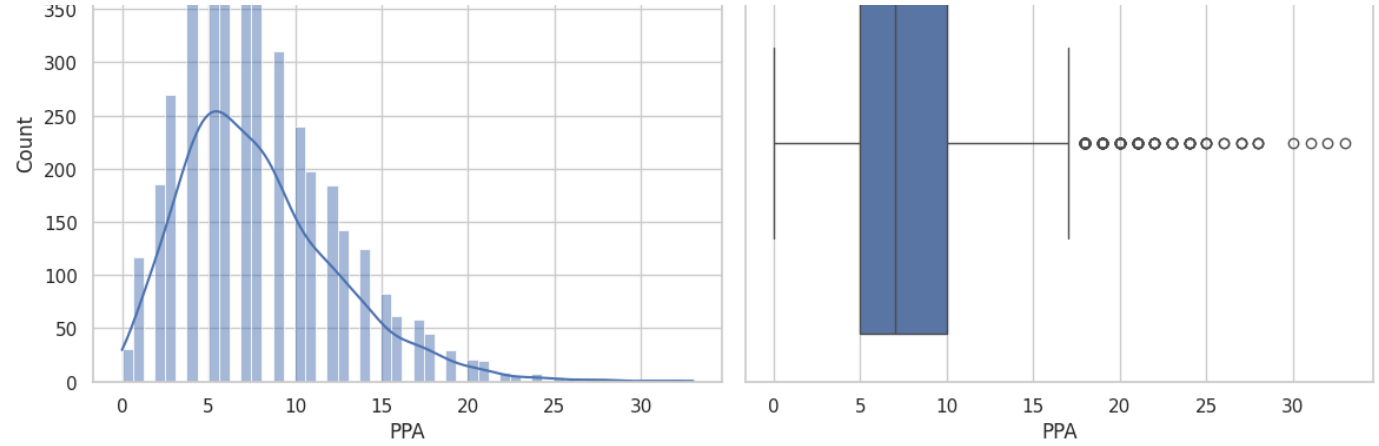

Box Plot of Dist

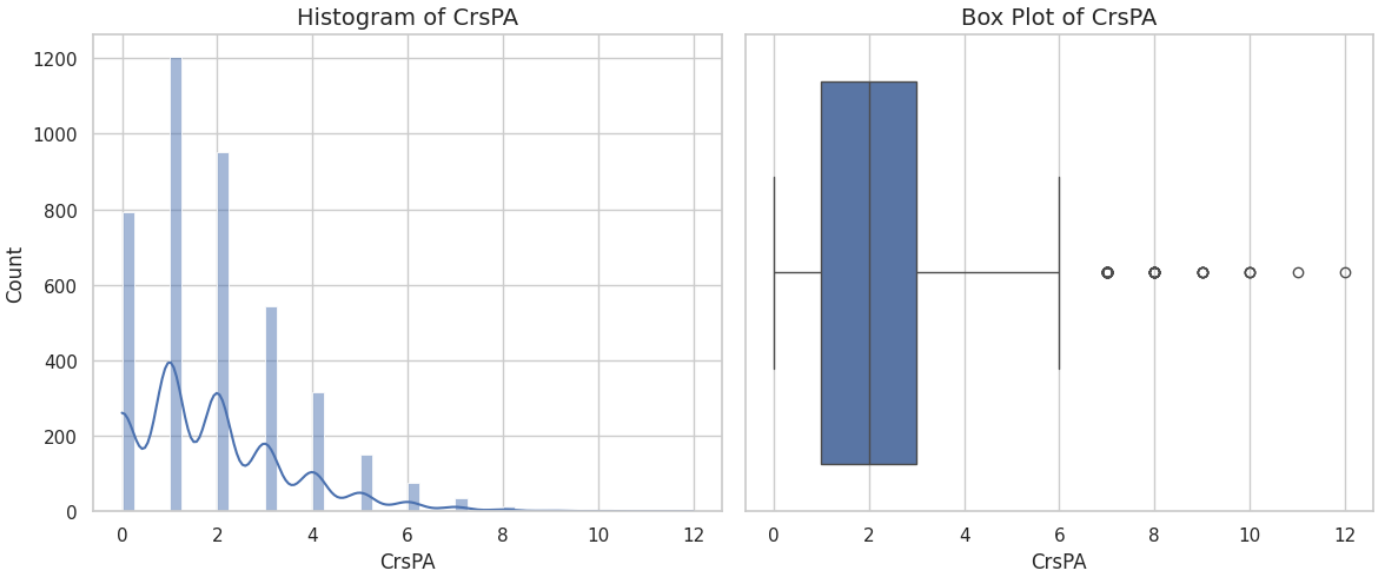

Count
Dist
Dist


Histogram of SCA
Box Plot of SCA
Count
SCA
SCA


Histogram of KP
Box Plot of KP
Count
KP
KP


Histogram of PPA
Box Plot of PPA

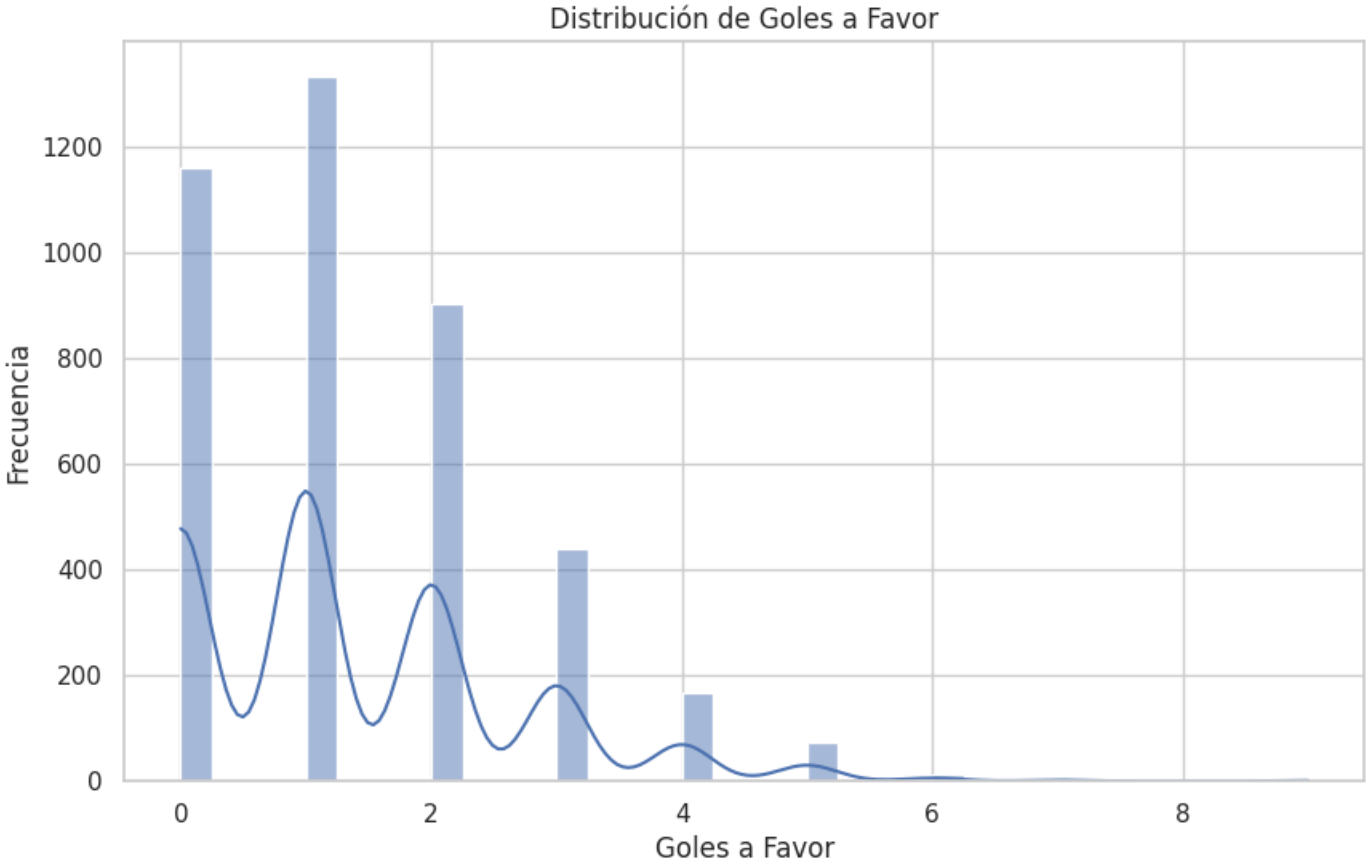

Count
PPA
PPA

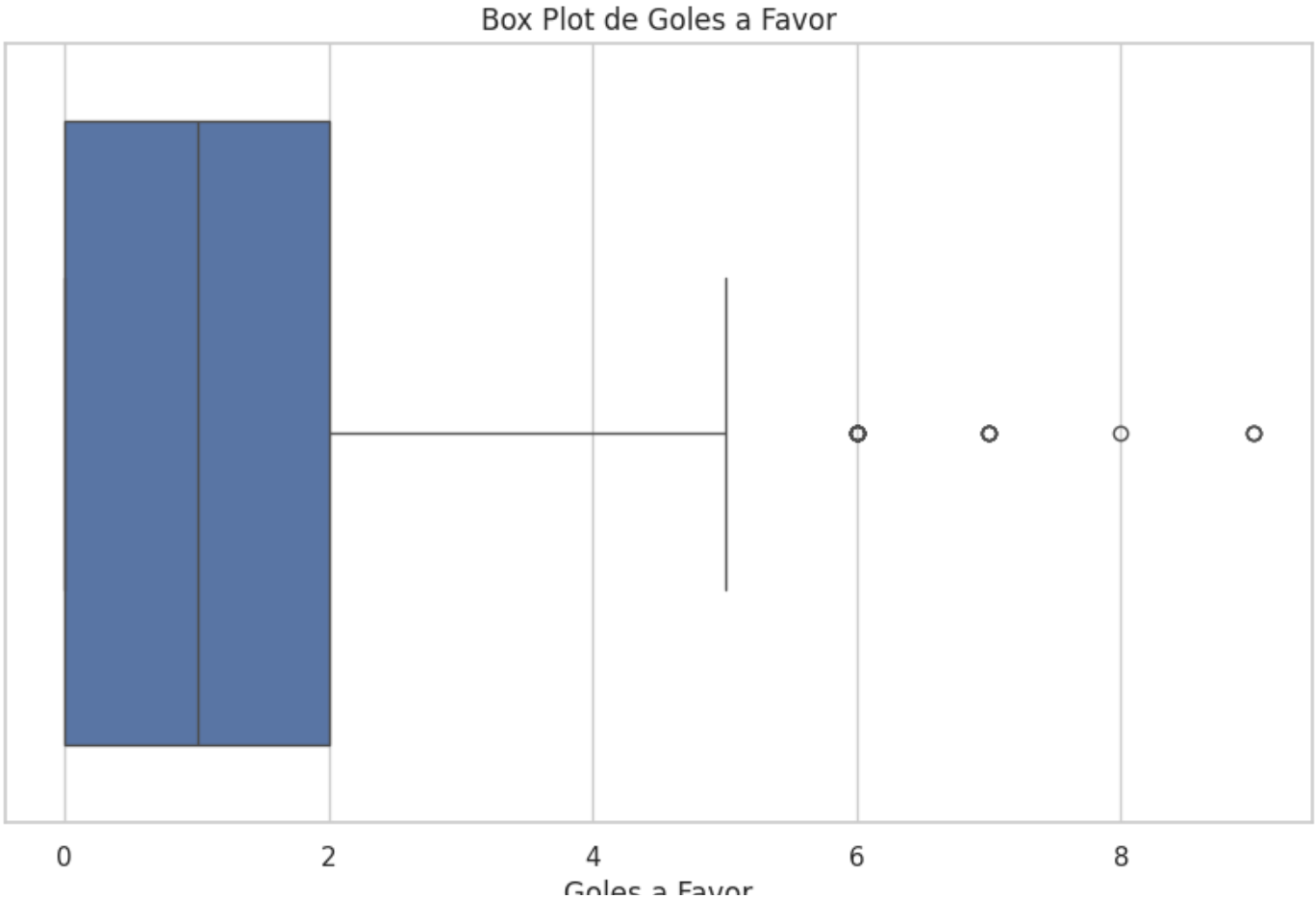

Histogram of CrsPA
Box Plot of CrsPA
Count
CrsPA
CrsPA

```
#analisis de goles a favor
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Suponiendo que 'data' es tu DataFrame
# Estadísticas descriptivas
######dsXlsNumeric = dsXls.loc[:, ['GF', 'GA', 'xG', 'xGA', 'Poss', 'Attendanc
print(dsXlsNumeric.describe())
print(dsXlsNumeric.skew())  # Asimetría
print(dsXlsNumeric.kurt())  # Curtosis

# Histograma para la variable 'GF' (Goles a favor)
plt.figure(figsize=(10, 6))
sns.histplot(dsXlsNumeric['GF'], kde=True)
plt.title('Distribución de Goles a Favor')
plt.xlabel('Goles a Favor')
plt.ylabel('Frecuencia')
plt.show()

# Diagrama de caja para 'GF'
plt.figure(figsize=(10, 6))
sns.boxplot(x=dsXlsNumeric['GF'])
plt.title('Box Plot de Goles a Favor')
plt.xlabel('Goles a Favor')
plt.show()

```

```
                GF           GA           xG           xGA         Poss  \
count  4092.000000  4092.000000  4092.000000  4092.000000  4092.000000
mean      1.377810     1.377810     1.346163     1.346163    50.001222
std       1.277631     1.277631     0.796551     0.796551    12.726702
min       0.000000     0.000000     0.000000     0.000000    18.000000
25%       0.000000     0.000000     0.700000     0.700000    41.000000
50%       1.000000     1.000000     1.200000     1.200000    50.000000
75%       2.000000     2.000000     1.800000     1.800000    59.000000
max       9.000000     9.000000     5.900000     5.900000    82.000000
```

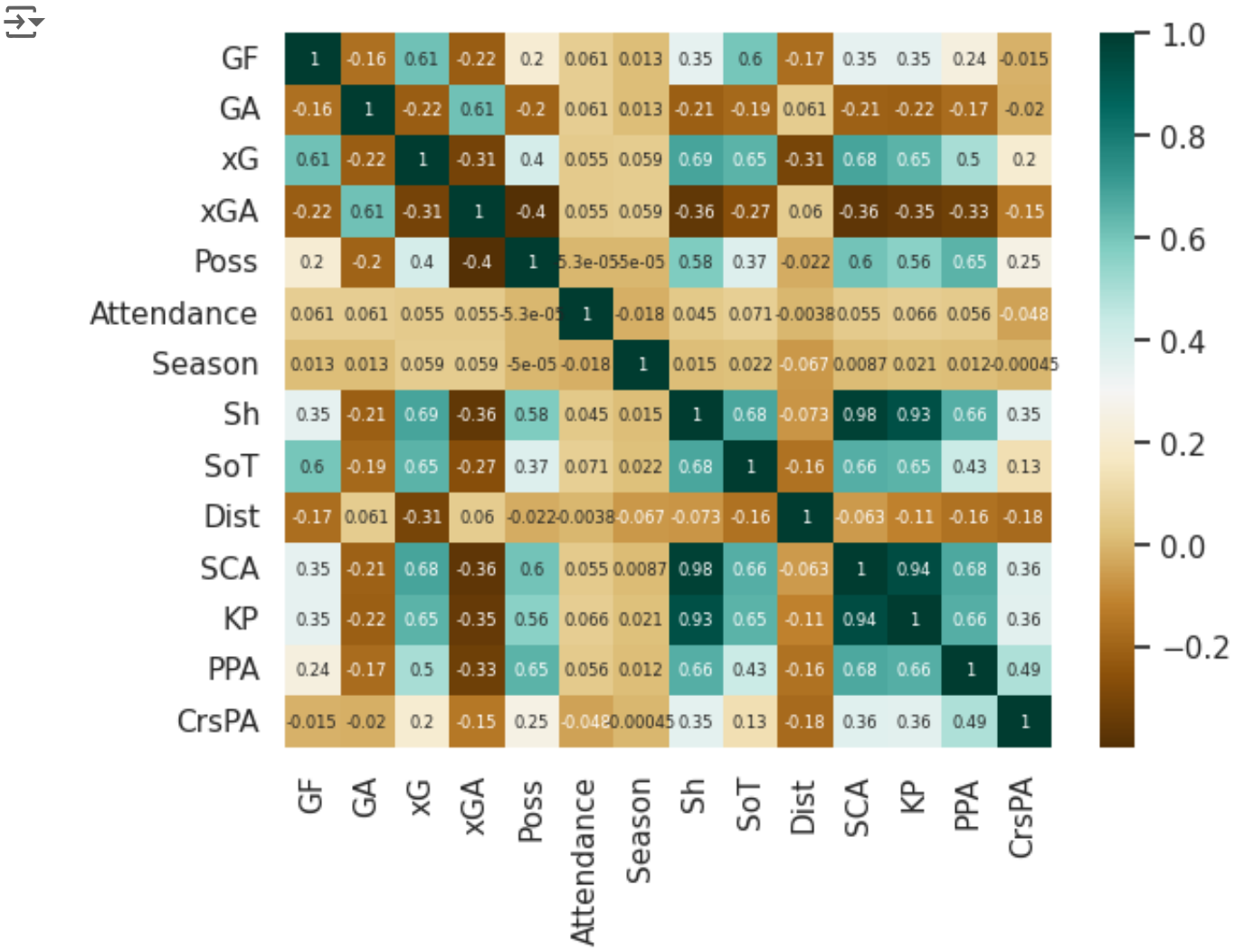
```
max       9.000000     9.000000     34.000000     34.000000     82.000000

          Attendance       Season           Sh          SoT         Dist  \
count   4092.000000  4092.000000  4092.000000  4092.000000  4092.000000
mean   36912.650049  2019.214076    12.317693     4.102884    17.675318
std    15301.262664     1.566615     5.427259     2.437269     3.038064
min     2000.000000  2017.000000     0.000000     0.000000     5.300000
25%    29296.000000  2018.000000     8.000000     2.000000    15.700000
50%    32092.500000  2019.000000    12.000000     4.000000    17.600000
75%    51237.000000  2021.000000    16.000000     6.000000    19.500000
max    83222.000000  2022.000000    36.000000    15.000000    35.000000

                SCA           KP          PPA         CrsPA
count  4092.000000  4092.000000  4092.000000  4092.000000
mean     21.840176     9.215054     7.901760     1.869501
std       9.897661     4.401972     4.553291     1.613642
min       0.000000     0.000000     0.000000     0.000000
25%      14.000000     6.000000     5.000000     1.000000
50%      21.000000     9.000000     7.000000     2.000000
75%      28.000000    12.000000    10.000000     3.000000
max      66.000000    34.000000    33.000000    12.000000
GF            1.099282
GA            1.099282
xG            0.919804
xGA           0.919804
Poss         -0.000095
Attendance    0.635124
Season        0.088767
Sh            0.577021
SoT           0.716879
Dist          0.457835
SCA           0.586169
KP            0.666184
PPA           0.998439
CrsPA         1.215871
dtype: float64
GF            1.712004
GA            1.712004
xG            1.066495
xGA           1.066495
Poss         -0.670897
Attendance    0.035171
Season       -1.151943
Sh            0.228083
SoT           0.497607
Dist          1.744887
SCA           0.206447
KP            0.448685
PPA           1.463371
CrsPA         2.104929
dtype: float64
```


Distribución de Goles a Favor
Frecuencia
0
200
400
600
800
1000
1200
0
2
4
6
8
Goles a Favor


Box Plot de Goles a Favor
0
2
4
6
8
Goles a Favor

Goles a favor

```
#CORRELACIÓN ENTRE VARIABLES
dsXlsNumeric = dsXls[['GF','GA','xG','xGA','Poss','Attendance','Season','Sh',
c= dsXlsNumeric.corr()

sns.heatmap(c,cmap="BrBG", annot=True, annot_kws={"size": 6})

plt.figure(figsize=(19, 9))

plt.rcParams["font.size"] = "2"
```




```
<Figure size 1900x900 with 0 Axes>
```