Tipologia y ciclo de vida de los datos

Alumna: Mabel Arroyo Adeba

PRA2: Limpieza y análisis de datos

Fecha: Enero 2021

Indice

1. Descripción del dataset (0,5 puntos)

Para la realización de esta práctica se utilizará el dataset generado en la PRA1. Se trata de un dataset que recopila los datos de los sorteos de la lotería primitiva que se han celebrado desde el jueves 21 de junio de 1991 hasta el 31 de octubre de 2020. Es un dataset en formato .csv (Historico_Primitiva.csv)

Los campos del fichero Historico_Primitiva.csv son:

Nombre Campo	Tipo	Descripción	Ejemplo
INDICE	Numérico	Número de línea	1
FEC_LARGA	Texto	Fecha descriptiva	Resultado del sábado 31 de octubre de 2020
FEC_SORTEO	Fecha	Fecha del sorteo en formato AAAMMDD	20201031
B1	Numérico	Valor Bola 1	2
B2	Numérico	Valor Bola 2	13
B3	Numérico	Valor Bola 3	21
B4	Numérico	Valor Bola 4	32
B5	Numérico	Valor Bola 5	41
B6	Numérico	Valor Bola 6	43
C	Numérico	Complementario	40
R	Numérico	Reintegro	9
RECAUDACION	Numérico	Importe recaudado	11.420.804,00 €
BOTE	Numérico	Importe del bote	7.500.000,00 €
ACE	Numérico	Acertantes de la categoría especial (6 aciertos + R)	0
PCE	Numérico	Premio de la categoría especial	0,00 €
AC1	Numérico	Acertantes de la categoría 1ª (6 aciertos)	1
PC1	Numérico	Premio categoría 1ª	1.450.282,32 €
AC2	Numérico	Acertantes de la categoría 2ª (5 aciertos + C)	9

PC2	Numérico	Premio categoría 2ª	24.171,37 €
AC3	Numérico	Acertantes de la categoría 3ª (5 aciertos)	206
PC3	Numérico	Premio categoría 3ª	2.288,07 €
AC4	Numérico	Acertantes de la categoría 4ª (4 aciertos)	10.098
PC4	Numérico	Premio categoría 4ª	75,40 €
AC5	Numérico	Acertantes de la categoría 5ª (3 aciertos)	189.207
PC5	Numérico	Premio categoría 5ª	8,00 €
ACR	Numérico	Acertantes del Reintegro	1.100.069
PCR	Numérico	Premio Reintegro	1,00 €

Para ello además contaremos con el fichero "variacion_del_pib.csv" en el que se presenta la variación anual del PIB de España desde 1971 hasta el tercer trimestre de 2020 extraído de:

https://www.epdata.es/datos/pib-espana-ine-contabilidad-nacional-trimestra/36/espana/106

Los campos de este fichero son:

Nombre Campo	Tipo	Descripción	Ejemplo
Año	Texto	Año	"1971"
Periodo	Texto	Periodo (trimestre)	"Trimestre 1"
Variación anual (en %)	Texto	Variación anual	"2,8"

Al observar el fichero podemos ver que los datos se presentan como de tipo texto. Para hacer los análisis deberemos transformarlos en numéricos.

Teniendo en cuenta que en ese periodo de tiempo hemos sufrido dos grandes crisis (la del 2008 y la actual producida por la pandemia del COVID-19), con la información extraída de estos dos ficheros, intentaremos responder a la pregunta:

¿Existe alguna relación entre la serie temporal de la recaudación de los años 2000 al 2020 y los índices económicos en ese periodo de tiempo? Es decir ¿La recaudación aumenta en tiempos de crisis o disminuye?

2. Integración y selección de los datos a analizar (0,5 ptos)

Seleccionaremos de cada dataset los datos correspondientes a los años 2000 a 2020. Puesto que los datos del PIB son trimestrales y los de la recaudación tienen información dos veces a la semana, habrá que establecer un criterio para poder homegeneizar los intervalos.

El criterio de homogeneización de datos consistirá en calcular la media de la recaudación de todos los sorteos celebrados por trimestre. Una vez hecho integraremos los dos datasets y seleccionaremos los datos en las fechas indicadas.

El dataset resultante tendrá la forma:

Nombre Campo	Tipo	Descripción	Ejemplo
AÑO	Numérico	Año	2000
TRIM	Numérico	Trimestre	1
PIB	Numérico	Valor trimestral de la variación anual del PIB	5,38
REC	Numérico	Recaudación media de los sorteos de la Primitiva en el trimestre que corresponda	14.894.774,94

Hay que tener en cuenta que este apartado no se completa hasta que los datos estén limpios. Es por eso que en el flujo del código, dejamos aparcada la integración y seguimos con el punto siguiente de limpieza de datos. Una vez tenemos los datos limpios finalizamos la unión de los datos.
Después de la limpieza y finalización de la integración, se hará el análisis en el punto 4.

3. Limpieza de datos (2 ptos)

-DATOS NULOS:

El fichero de los datos del **PIB**, no tiene datos nulos. Lo qué sí encontramos las siguientes líneas de texto adicionales, que eliminamos en el primer apartado.

;;
;;
"Métrica":"Variación anual (en %)"
"Fuente":"PIB"
"Clasificación":""
"Unidad":"%"
"Escala":"Unidades"
"EscalaFactorPotencia10":"0"
"SonDatosNumericos":"True"
;;
;;
"Url":"https://www.epdata.es/pib-tambien-pisa-acelerador-crece-24-variacion-interanual/c560967a-b1f7-40ef-b7b2-42f84a8f1088"
"Titulo":"Variación anual del PIB de España hasta el tercer trimestre de 2020"
"Subtitulo":"Cambio metodológico a partir del primer trimestre de 1996"

En el fichero de datos de **sorteos** se detectaron 16 filas con el campo recaudación a 'None'.

Para rellenar este valor teníamos dos opciones:

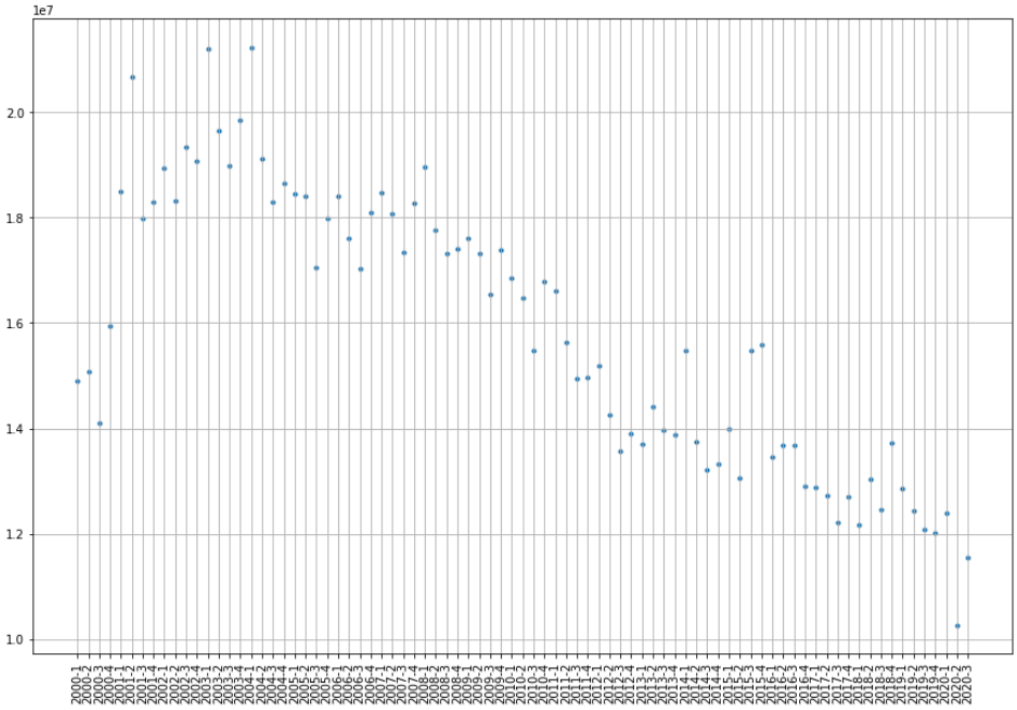
1. Rellenar con la media de los valores anterior y posterior. Se podría optar por esta solución en lugar de tomar la media total, porque es una sucesión temporal a lo largo de 10 años y la media total no tiene porqué estar en torno al valor perdido. Es más probable que sea más cercana a los valores anterior y posterior.
2. Eliminar las filas: puesto que después haremos media trimestral eliminar estas filas no supondría gran perdida de datos en el caso que nos ocupa. El número total de filas es de 2157.

Por practicidad, optamos por la segunda opción y limpiamos el dataframe eliminando las 16 filas que no contienen valor en el campo "Recaudación".

DETECCIÓN DE VALORES EXTREMOS

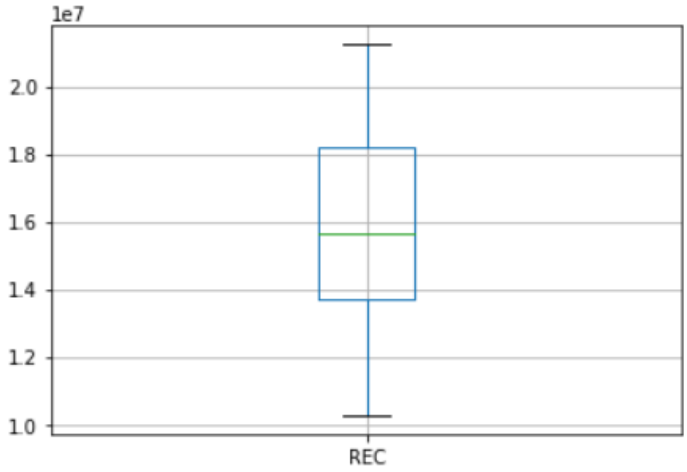
Después de la limpieza, y habiendo terminado la integración obtenemos nuestro dataset definitivo. En él hacemos la detección de outliers. Para ello realizamos una representación de los punto en el tiempo y también un diagrama de cajas para cada uno de los grupos de datos que estamos tratando: el de recaudación de los sorteos y el de los datos del PIB.

Para los datos de recaudación:

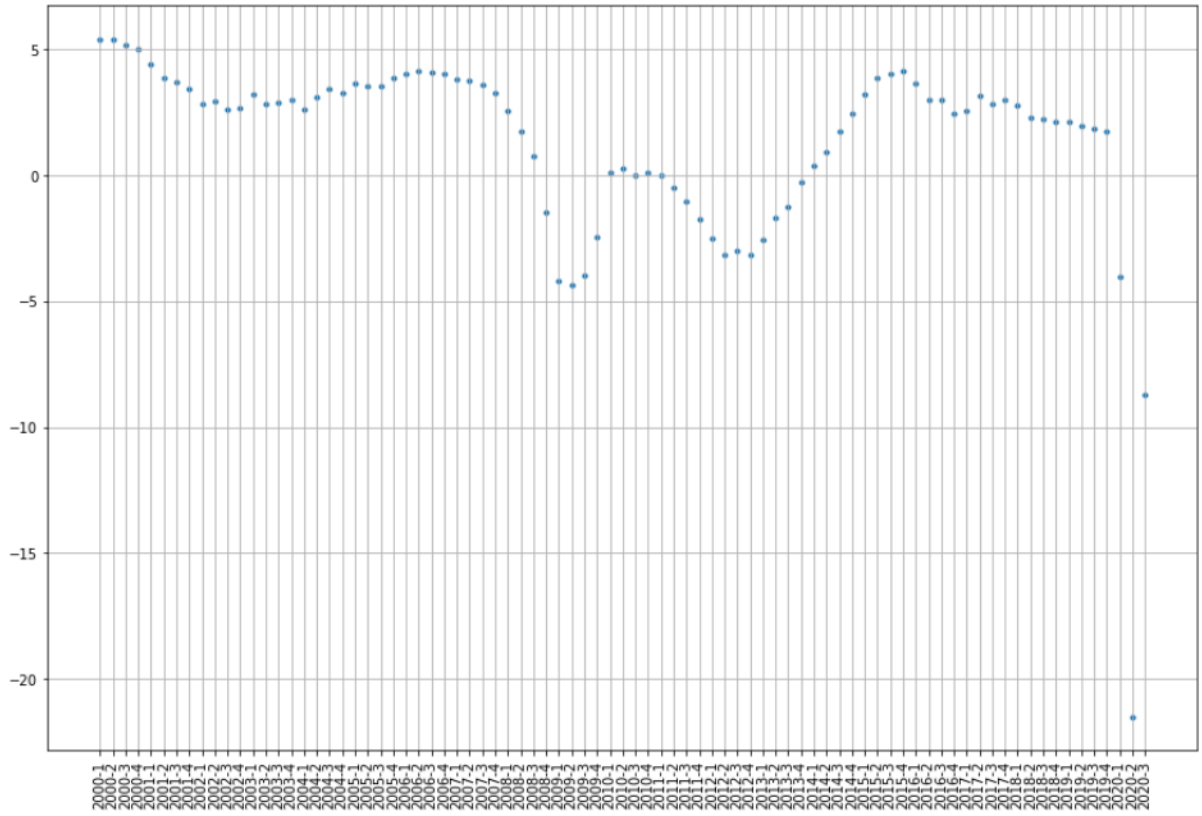


Podemos observar varios grupos que aparecen apartados, pero no queda claro que sean puntos extremos.

En el boxplot siguiente confirmamos que no existen outliers.

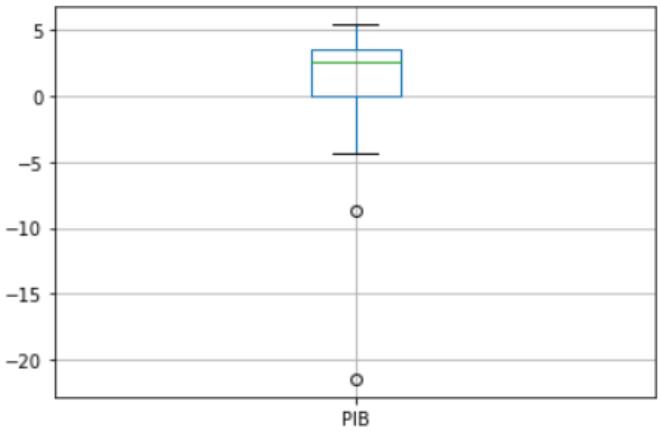


Para los datos del PIB:



Podemos ver que a finales de 2008 y principios de 2009 y también a principios de 2020 hay valores extremos. Estos valores no se descartan pues existen con un sentido. Se produjeron por las crisis del 2008 y del COVID-19.

En el diagrama de caja, también vienen identificados:



4. Análisis de los datos (2,5 ptos)

4.1. Selección de los grupos a analizar

Ya se ha estado comentando en los apartados anteriores. Los grupos que se analizarán son los resultados del PIB y de la recaudación de los sorteos.

4.2. Comprobación de la normalidad y homocedasticidad

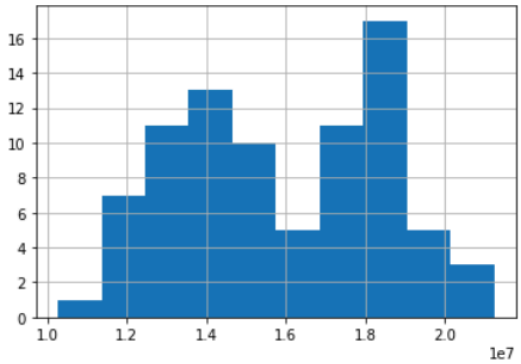
Para la comprobación de la normalidad, utilzaremos el test Shapiro-Wilk y también de forma gráfica con histogramas.

Como podemos observar a continuación el test en los dos casos arroja un p<0,05.

In [228]:
```
# Comprobación de la normalidad del Recaudación
stat, p = shapiro(df_PRI_PIB['REC'])
print('Estadisticos=%.3f, p=%.3f' % (stat, p))
# Interpretación
alpha = 0.05
if p > alpha:
   print('La muestra parece Gaussiana o Normal (no se rechaza la hipótesis nula H0)')
else:
   print('La muestra no parece Gaussiana o Normal(se rechaza la hipótesis nula H0)')

df_PRI_PIB['REC'].hist()
```

```
Estadisticos=0.959, p=0.009
La muestra no parece Gaussiana o Normal(se rechaza la hipótesis nula H0)
```

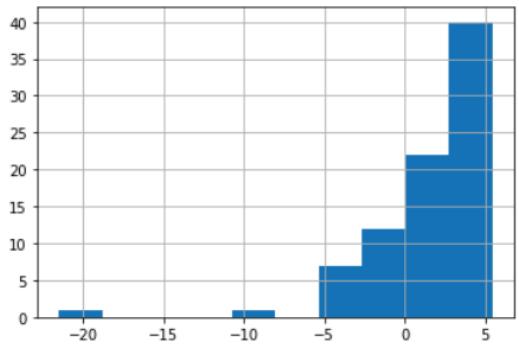
Out[228]: <AxesSubplot:>



In [227]:
```
# Comprobación de la normalidad del PIB
stat, p = shapiro(df_PRI_PIB['PIB'])
print('Estadisticos=%.3f, p=%.3f' % (stat, p))
# Interpretación
alpha = 0.05
if p > alpha:
   print('La muestra parece Gaussiana o Normal (no se rechaza la hipótesis nula H0)')
else:
   print('La muestra no parece Gaussiana o Normal(se rechaza la hipótesis nula H0)')

df_PRI_PIB['PIB'].hist()
```

```
Estadisticos=0.715, p=0.000
La muestra no parece Gaussiana o Normal(se rechaza la hipótesis nula H0)
```

Out[227]: <AxesSubplot:>



Por lo tanto, ni la recaudación ni el PIB siguen una distribución normal.

Para comprobar la homocedasticidad (variación de la varianza a lo largo del tiempo) utilizamos el test Fligner-Killen:

```
# Para la recaudación
stat, p = fligner(df_PRI_PIB['REC'],df_PRI_PIB['ANIO'])
print("p-valor",p)
```

```
p-valor 1.1404883871033193e-23
```

```
# Para el PIB
stat, p = fligner(df_PRI_PIB['PIB'],df_PRI_PIB['ANIO'])
print("p-valor",p)
```

En ambos casos da un valor inferior a 0,05 lo cual nos indica que hay variación de la varianza a lo largo del tiempo.

4.1. Aplicación de pruebas estadísticas

-PRUEBAS ESTADÍSTICAS BÁSICAS:

	ANIO	TRIM	PIB	REC
count	83.000000	83.000000	83.000000	8.300000e+01
mean	2009.879518	2.481928	1.366867	1.587300e+07
std	6.027169	1.119249	3.741771	2.618920e+06
min	2000.000000	1.000000	-21.500000	1.026409e+07
25%	2005.000000	1.500000	0.065000	1.368353e+07
50%	2010.000000	2.000000	2.650000	1.562836e+07
75%	2015.000000	3.000000	3.555000	1.818591e+07
max	2020.000000	4.000000	5.430000	2.124120e+07

Aquí podemos observar que el PIB tiene una media de 1,36 con una desviación estándar de 3,74, siendo su valor mínimo de -21,5 y el máximo de 5,43

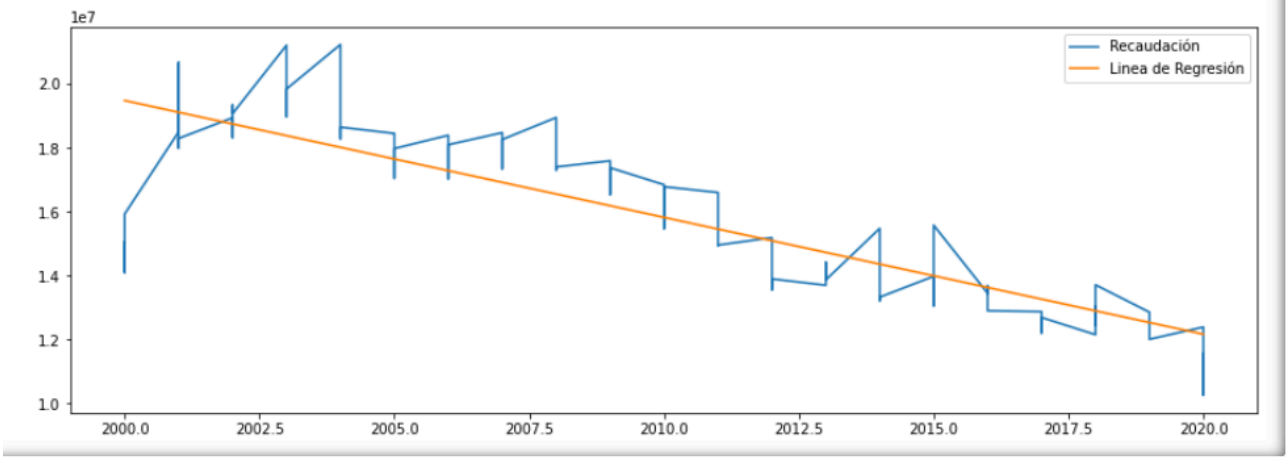
En el caso de la recaudación, la media es de 15,87 millones de euros con una desviación standard de 2,62 millones de euros. El mínimo es de 10.26 millones de euros y el máximo de 21,24 millones de euros.

-MODELO DE REGRESIÓN:

Para la recaudación: En el siguiente modelo vemos como se crea una línea de regresión con tendencia descendente.

```
# Creamos el modelo de regresión
regresion_REC = LinearRegression()

# Entrenamos el modelo con los datos
# El "reshape" se ha de poner porque si no da error.
# El fit necesita un array de dos dimensiones
X = df_PRI_PIB['ANIO'].array.to_numpy()
X = X.reshape(-1,1)
print(X.shape)
regresion_REC.fit(X, df_PRI_PIB['REC'])

# Hacemos la predicción. Con esto se crea la recta que muestra la tendencia
tendencia_REC = regresion_REC.predict(X)

# Visualizamos los datos junto a la tendencia calculada con el modelo de regresión
fig, ax = plt.subplots(1, 1, figsize=(15, 5))
ax.plot(X, df_PRI_PIB['REC'], label = 'Recaudación')
ax.plot(X, tendencia_REC, label = 'Linea de Regresión')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x1a36e75110>
```

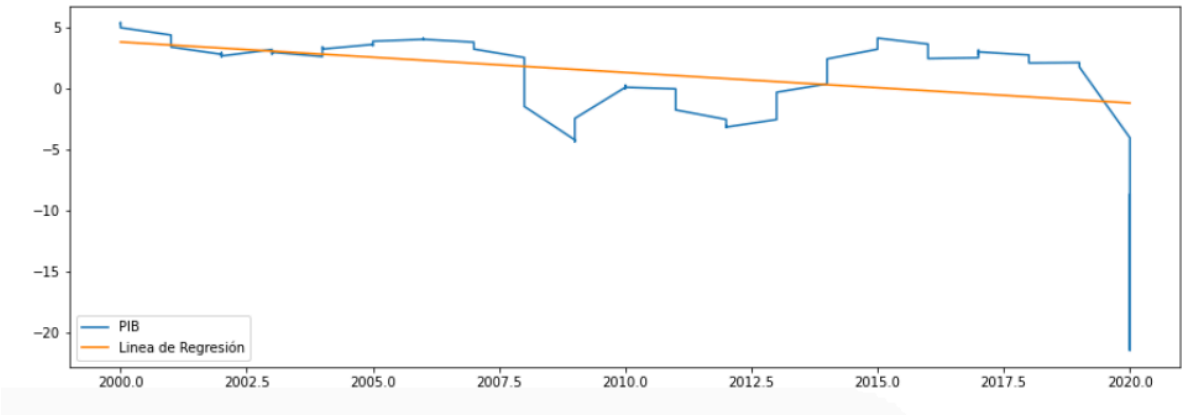


Para el PIB: El modelo de regresión en este caso también tiene una tendencia descendente.

```
# Creamos el modelo de regresión
regresion_PIB = LinearRegression()

# Entrenamos el modelo con los datos
# El "reshape" se ha de poner porque si no da error.
# El fit necesita un array de dos dimensiones
X = df_PRI_PIB['ANIO'].array.to_numpy()
X = X.reshape(-1,1)
print(X.shape)
regresion_PIB.fit(X, df_PRI_PIB['PIB'])

# Hacemos la predicción. Con esto se crea la recta que muestra la tendencia
tendencia_PIB = regresion_PIB.predict(X)

# Visualizamos los datos junto a la tendencia calculada con el modelo de regresión
fig, ax = plt.subplots(1, 1, figsize=(15, 5))
ax.plot(X, df_PRI_PIB['PIB'], label = 'PIB')
ax.plot(X, tendencia_PIB, label = 'Linea de Regresión')
plt.legend()
```

```
(83, 1)
```

```
<matplotlib.legend.Legend at 0x1a372d38d0>
```

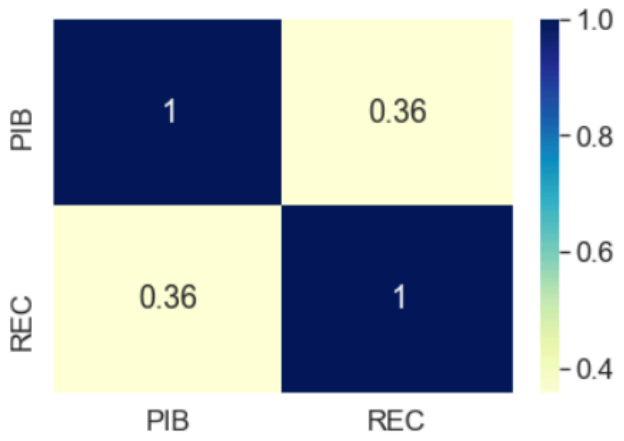


-CORRELACIÓN:

A continuación realizamos una prueba de correlación entre los dos grupos de datos y podemos observar un índice de correlación muy bajo. Esto nos indica que estas dos variables están poco correlacionadas, es decir que si una aumenta o disminuye la otra no tiene el mismo comporamiento.

```
# Calculamos las correlaciones
correlacion = datos_reducidos.corr()
sb.set(font_scale=1.5) # Aumenta el tamaño de letra de los valores de correlación
sb.heatmap(correlacion, annot=True, cmap="YlGnBu")
```

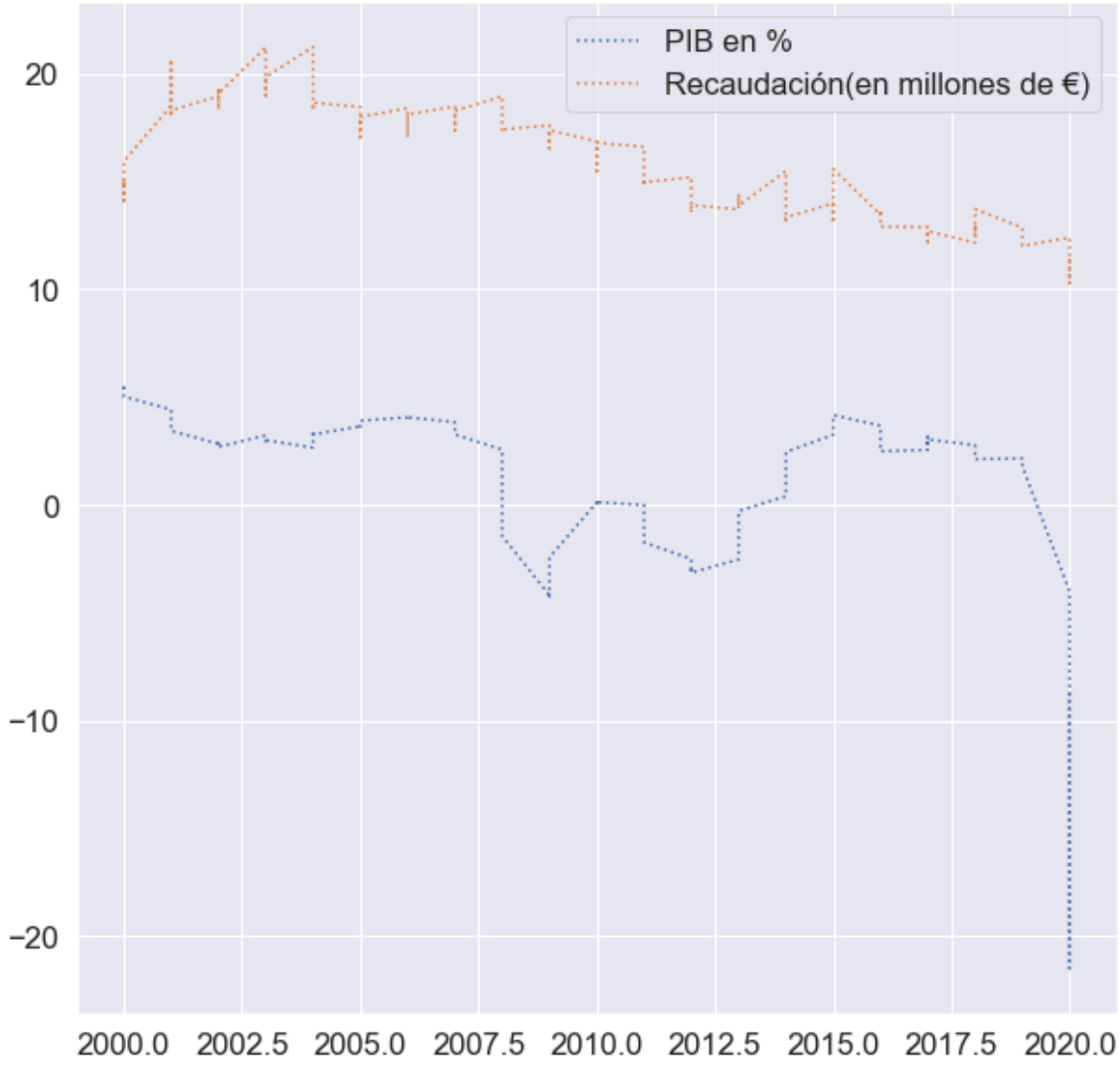
```
: <AxesSubplot:>
```



5. Representación de los resultados (2 ptos)

En este apartado se muestran los histogramas descriptivos y las series temporales donde se verá visualmente la relación entre la recaudación y el PIB.

La mayoría de lo resultados gráficos se han ido presentando a lo largo de todos los puntos anteriores. Aquí veremos una última gráfica donde se representa a la vez la recaudación y el PIB. Se ha ajustado la escala de la recaudación a millones de € .



6. Resolución del problema (0,5 ptos)

Después del trabajo realizado en los puntos anteriores, ya podemos responder a la pregunta inicial:

¿Existe alguna relación entre la serie temporal de la recaudación de los años 2000 al 2020 y los índices económicos en ese periodo de tiempo? Es decir ¿La recaudación aumenta en tiempos de crisis o disminuye?

La respuesta es: No. No existe una relación clara entre el nivel de recaudación que se obtiene en los sorteos de la Primitiva y la variación del índice PIB en los meses de grandes crisis económicas.
Sí se puede afirmar que con el tiempo, el nivel de recaudación disminuye, igual que lo ha hecho el PIB. Como vimos en los modelos de regresión ambos valores muestran una tendencia descendente.

Es una pena, que con en los últimos tiempos tanto la economía como la ilusión de las personas, tienda a disminuir. Contra esto último la ciencia de datos aún no puede hacer nada.

7. Código (2 ptos)

A continuación se incluye el código en Python del proceso de limpieza y análisis.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import shapiro
from scipy.stats import fligner
from sklearn.linear_model import LinearRegression
import seaborn as sb

# Cargamos el fichero con los datos de los sorteos
# y hacemos una previsualización de la estructura y
# de las primeras líneas de datos
df_prim = pd.read_csv('Historico_Primitiva.csv',sep  = ';')
print("Número de filas y columnas:", df_prim.shape)
print("Nombres de las columnas:", list(df_prim.columns))
df_prim.head(5)

# Cargamos el fichero con los datos del PIB
# y hacemos una previsualización de la estructura y
# de las primeras líneas de datos
df_pib = pd.read_csv('variacion_anual_del_pib.csv',sep  = ';')
print("Número de filas y columnas:", df_pib.shape)
print("Nombres de las columnas:", list(df_pib.columns))
df_pib.head(5)

# Miramos las últimas y podemos observar que hay texto adicional
# Este texto carece de sentido como datos, así que borramos manualmente
# El resultado lo grabamos en variacion_anual_del_pib2.csv
df_pib.tail(5)

# Volvemos a cargar el fichero con los datos del PIB
# y hacemos una previsualización de la estructura y
```

```
# de las primeras líneas de datos
df_pib = pd.read_csv('variacion_anual_del_pib2.csv',sep  = ';')
print("Número de filas y columnas:", df_pib.shape)
print("Nombres de las columnas:", list(df_pib.columns))
df_pib.head(5)

# Miramos las últimas y podemos observar que ahora ya está correcto
df_pib.tail(5)

# FICHERO DE DATOS SORTEOS
# Selección de los registros correspondientes a las fechas entre enero de 2000 y
octubre de 2020
df_prim = df_prim.iloc[:len(df_prim[df_prim.Fecha_AAAAMMDD>=20000101])]

print("Número de filas y columnas:", df_prim.shape)
print("Nombres de las columnas:", list(df_prim.columns))

# Vemos las 5 primeras filas
df_prim.head(5)

# Vemos las últimas 5 filas para comprobar que ha hecho bien la selección.
df_prim.tail(5)

# FICHERO DE DATOS PIB
# Selección de los registros correspondientes a las fechas entre enero de 2000 y
octubre de 2020
# Se eliminan las filas de los años anteriores al 2000
df_pib['Año'] = df_pib['Año'].astype('int')
df_pib = df_pib.iloc[len(df_pib[(df_pib.Año)<2000]):]

print("Número de filas y columnas:", df_pib.shape)
df_pib.head(5)

df_pib.tail(5)

# Ordenamos en orden ascendente
df_prim.sort_values('Fecha_AAAAMMDD', inplace=True)
df_prim.head(5)

# Integramos los datos de los dos datasets anteriores en uno nuevo
# que contiene los datos que necesitamos para el análisis
# Año, trimestre, PIB y recaudación

df_PRI_PIB = pd.DataFrame()

# Recogemos el año y lo convertimos a tipo entero
df_PRI_PIB['ANIO'] = df_pib['Año'].astype('int')

# Recogemos el trimestre quedándonos con el número que aparece en la cadena
# y lo transformamos en numérico (int)
df_PRI_PIB['TRIM'] = df_pib['Periodo'].str.extract(r"([\d])").astype('int')
```

```
# Recogemos el valor del PIB, sustituimos el símbolo decimal coma (,) por punto
(.)
# para que lo trate de forma correcta
df_PRI_PIB['PIB'] = df_pib['Variación anual (en
%)'].str.replace(',','.').astype('float64')

df_PRI_PIB.head(5)
# Limpieza de datos.Como son pocas filas, lo comprobamos de forma visual
pd.options.display.max_rows = None # Con esto, se visualizan todas las filas
df_pib

# Comprobamos cuántas filas tienen la recaudación sin informar
df_prim.loc[df_prim['Recaudación'] == 'None'].shape


# El total de filas es:
df_prim.shape

def Trimestre(mes):
    '''
      Calcula el trimestre correspondiente al año

      Parámetro:
          mes: Mes del que queremos saber el trimestre
      Resultado:
          Valor del trimestre en función del mes recibido como parámetro
    '''
    if mes >= 1 and mes <= 3:
        return 1
    elif mes >= 4 and mes <= 6:
        return 2
    elif mes >= 7 and mes <= 9:
        return 3
    elif mes >= 10 and mes <= 12:
        return 4


## Creamos un dataset temporal con los campos 'Año' 'Trimestre' y 'Recaudación'
# Año y Trimestre lo generamos a partir del campo 'Fecha_AAAAMMDD'
temp_rec = pd.DataFrame()
temp_rec['FEC']= pd.to_datetime(df_prim['Fecha_AAAAMMDD'].astype(str))

temp_rec['REC']= df_prim['Recaudación']

# Convertimos a numérico el valor de la recaudación
# Primero eliminamos el punto de los miles
# y luego cambiamos la coma de los decimales por un punto
# Por último convertimos en float
temp_rec['REC'] = temp_rec.apply(lambda row: row['REC'].replace('.',''), axis=1)
temp_rec['REC'] = temp_rec.apply(lambda row: row['REC'].replace(',','.'), axis=1)
temp_rec['REC'] = temp_rec['REC'].astype('float64')

# Recogemos el año y el trimestre
temp_rec['ANIO'] = temp_rec.apply(lambda row: row['FEC'].year, axis=1)
```

```
temp_rec['TRIM'] = temp_rec.apply(lambda row: Trimestre(row['FEC'].month),
axis=1)


temp_rec

# Hacemos la media por trimestre y lo incluimos en nuestro dataset de salida
# en el campo df_PRI_PIB['REC'] (tarea de integración que se había quedado
pendiente)
rec_tri = [] # Recaudación por trimestre en cada año
for anio in range(2000, 2021):
    for trim in range(1,5):
        rec_tri.append(temp_rec.loc[(temp_rec['TRIM'] == trim) &
(temp_rec['ANIO'] == anio)].REC.mean())

# Debemos eliminar el último resultado porque los valores del PIB llegan
# Hasta el tercer trimestre de 2020 y los datos de los sorteos hasta el
# 31 de Octubre que ya correspondería al cuarto trimestre.
rec_tri.pop()

# Comprobar que el tamaño es el mismo que el de PIB
print(len(rec_tri))

# Integramos el valor en nuestro dataset resultado
df_PRI_PIB['REC'] = rec_tri
df_PRI_PIB['AT']= df_PRI_PIB.apply(lambda row: str(int(row['ANIO'])) + '-' +
str(int(row['TRIM'])), axis=1)

#df_PRI_PIB
# Representamos los valores de la recaudación para detectar valores extremos y
decidir
# cómo los tratamos
fig = plt.figure(figsize=(15, 10))
plt.xticks(rotation='vertical')
plt.grid()
plt.scatter(df_PRI_PIB['AT'], df_PRI_PIB['REC'],s=10)

df_PRI_PIB.boxplot("REC")

# Representamos los valores del PIB para detectar valores estremos y decidir cómo
los tratamos
fig = plt.figure(figsize=(15, 10))
plt.xticks(rotation='vertical')
plt.grid()
plt.scatter(df_PRI_PIB['AT'], df_PRI_PIB['PIB'],s=10)

df_PRI_PIB.boxplot("PIB")

# Guardamos el dataset resultado en un csv.
df_PRI_PIB.to_csv('PRI_PIB.csv', index=False, sep=';')

# Análisis de datos
# Vemos las primeras filas del dataset a analizar
df_PRI_PIB.head(5)
```

```
# Comprobación de la normalidad del Recaudación
stat, p = shapiro(df_PRI_PIB['REC'])
print('Estadisticos=%.3f, p=%.3f' % (stat, p))
# Interpretación
alpha = 0.05
if p > alpha:
   print('La muestra parece Gaussiana o Normal (no se rechaza la hipótesis nula
H0)')
else:
   print('La muestra no parece Gaussiana o Normal(se rechaza la hipótesis nula
H0)')

df_PRI_PIB['REC'].hist()

# Comprobación de la normalidad del PIB
stat, p = shapiro(df_PRI_PIB['PIB'])
print('Estadisticos=%.3f, p=%.3f' % (stat, p))
# Interpretación
alpha = 0.05
if p > alpha:
   print('La muestra parece Gaussiana o Normal (no se rechaza la hipótesis nula
H0)')
else:
   print('La muestra no parece Gaussiana o Normal(se rechaza la hipótesis nula
H0)')

df_PRI_PIB['PIB'].hist()


# Comprobación de homocedasticidad
# Para la recaudación
stat, p = fligner(df_PRI_PIB['REC'],df_PRI_PIB['ANIO'])
print("p-valor",p)

# Para el PIB
stat, p = fligner(df_PRI_PIB['PIB'],df_PRI_PIB['ANIO'])
print("p-valor",p)

# Mostramos las estadísticas descriptivas básicas.
df_PRI_PIB.describe()

# Creamos el modelo de regresión
regresion_REC = LinearRegression()

# Entrenamos el modelo con los datos
# El "reshape" se ha de poner porque si no da error.
# El fit necesita un array de dos dimensiones
X = df_PRI_PIB['ANIO'].array.to_numpy()
X = X.reshape(-1,1)
print(X.shape)
regresion_REC.fit(X, df_PRI_PIB['REC'])

# Hacemos la predicción. Con esto se crea la recta que muestra la tendencia
```

```
tendencia_REC = regresion_REC.predict(X)

# Visualizamos los datos junto a la tendencia calculada con el modelo de
regresión
fig, ax = plt.subplots(1, 1, figsize=(15, 5))
ax.plot(X, df_PRI_PIB['REC'], label = 'Recaudación')
ax.plot(X, tendencia_REC, label = 'Linea de Regresión')
plt.legend()

# Creamos el modelo de regresión para el PIB
regresion_PIB = LinearRegression()

# Entrenamos el modelo con los datos
# El "reshape" se ha de poner porque si no da error.
# El fit necesita un array de dos dimensiones
X = df_PRI_PIB['ANIO'].array.to_numpy()
X = X.reshape(-1,1)
print(X.shape)
regresion_PIB.fit(X, df_PRI_PIB['PIB'])

# Hacemos la predicción. Con esto se crea la recta que muestra la tendencia
tendencia_PIB = regresion_PIB.predict(X)

# Visualizamos los datos junto a la tendencia calculada con el modelo de
regresión
fig, ax = plt.subplots(1, 1, figsize=(15, 5))
ax.plot(X, df_PRI_PIB['PIB'], label = 'PIB')
ax.plot(X, tendencia_PIB, label = 'Linea de Regresión')
plt.legend()

# Nos quedamos con las columnas de PIB y recaudación
datos_reducidos = df_PRI_PIB
datos_reducidos = datos_reducidos.drop(['ANIO','TRIM','AT'], axis=1)

# Calculamos las correlaciones
correlacion = datos_reducidos.corr()
sb.set(font_scale=1.5) # Aumenta el tamaño de letra de los valores de correlación
sb.heatmap(correlacion, annot=True, cmap="YlGnBu")

# Para poder representar los datos, bajamos la escala de la recaudación para que
esté
# en la misma escala que el PIB
datos_recaudacion = datos_reducidos.apply(lambda row: row['REC']/1e6, axis=1)
# Visualizamos los datos conjuntamente para ver la relación
fig, ax = plt.subplots(1, 1, figsize=(10, 10))
ax.plot(df_PRI_PIB['ANIO'], datos_reducidos['PIB'], label = 'PIB en %', linestyle
= 'dotted')
ax.plot(df_PRI_PIB['ANIO'], datos_recaudacion, label = 'Recaudación(en millones
de €)', linestyle = 'dotted')
plt.legend()
```

Firma de participantes:

Contribuciones	Firma
Investigación previa	MAA
Redacción de las respuestas	MAA
Desarrollo código	MAA

Referencias Bibliográficas:

Teguayco Gutiérrez González Práctica 2: Limpieza y validación de los datos 6 de diciembre de 2017

Pregunta 4:

https://machinelearningparatodos.com/como-saber-si-una-variable-sigue-una-distribucion-normal-en-python/