Se importa las librerías

In **[**1]: import pandas **as** pd

**In** [2]=

import statsmodels. formula.api as smf

import statsmodels.api **as** sm

import matplotlib.pyplot **as** plt

**import** seaborn as sns

import scipy.stats as st

**import** numpy **as np**

import scipy.stats as ss

from itertools import combinations

import re

## preguntar *por* las funciones de la librería sm y smf

## https://yuasaavedraco.github.io/Docs/Regresi%C3%B3n\_Lineal\_Simple\_con\_Python.html

## https://local.coder.org/stepwise-regression-in-python

Pregunta 1:

Describa los datos, entregando gráficos y tablas que muestren las principales características de la base de datos. Ayuda: puede emplear el método .hist(bins-n\_bins, figsize = (ancho, alto)) para graficar un resu- men de histogramas de los datos.

Se hace una transformación al archivo original para que pase a ser **un** csv

In [3]: df=pd.read\_csv('/Users/milan/crime\_usa.csv',sep=';')

df.shape

Out [3]:

(50, 7)

Se revisa las correlaciones iniciales de las columnas del dataset

In [4]: plt.figure(figsize=(16, 6))

sns.heatmap(df.corr(), vmin=-1, vmax=**1,** annot=True**)**

<Axes Subplot:>

Out [4]:

In [5]:

- 100

M

1

0.76

0.53

-0.14

0.32

-0.18

-0.026

0.76

1

0.51

-0.18

0.29

-0.2

22

-0.75

-0.046

-0.50

♡ -

0.53

0.51

1

0.12

0.31

-0.28

0.12

-0.25

X4

C

-0.14

-0.18

0.12

1

-0.54

0.18

0.68

-0.00

-0.25

325

0.32

0.29

0.31.

-0.54

1

-0.63

-0.51

--0.50

-0.18

-0.2

-0.28

0.18

-0.63

1

0.59

- -0.75

-0.026

-0.046

0.12

0.68

-0.51

0.59

1

-1.00

X1

X2

X3

X4

X5

X6

X7

• Se muestra que las opciones X2,X3 y X5 tienen una correlación relativamente importante, por tanto estimo que serán significativas más adelante en el ejercicio

Se hace un modelo de regresión linear múltiple sin una limpieza previa para ver sus resultados generales

• Se ocupan 2 funciones para estimar regresión lineal a la espera de confirmar cual es la diferencia entre ambas

x\_corr-sm\_add\_constant(df [[ 'X2', 'X3','X4', 'X5', 'X6', 'X7"]])

y= df ['x1']

regressor\_OLS = sm.OLS (y,x\_corr).fit()

regressor\_OLS.summary()

C:\Users\milan\anaconda\lib\site-packages\statsmodels\tsa\tsatools.py:142: FutureWarning: In a future version of pandas all arguments of concat except for the argument 'objs' will be keyword-only

x = pd.concat(x[::order], 1)

OLS Regression Results

Out [5]:

Dep. Variable:

X1

R-squared: 0.613

Model**:**

OLS

Method**:**

Least Squares

**Adj**. R-squared**:** 0.559

**F-**statistic: 11.36

00:45:32

**Date:** Sat, 14 May 2022 Prob (**F**-statistic**)**: 1.42e-07

**Time:**

Log-Likelihood: -330.87

No. Observations:

50

AIC:

675.7

**Df** Residuals:

43

BIC:

689.1

Df Model**:**

Covariance Type:

6

nonrobust

coef std err

t P>|t|

[0.025 0.975]

const 100.3936 370.693 0.271 0.788 -647.180 847.968

X2

0.3323 0.060 5.574 0.000 0.212 0.453

X3

3.9982 2.682 1.490 0.143 -1.412 9.408

X4

1.8579

5.241 0.355 0.725 -8.711 12.427

**X5**

X6

7.8389

2.5588 3.427

7.760 1.010 0.318 -7.810 23.488

0.747 0.459

-4.352 9.470

X7

-3.2312 10.715 -0.302 0.764 -24.841 18.378

Omnibus: 23.696

Durbin**-**Watson**:**

1.860

Prob(Omnibus)**:**

0.000 Jarque**-Bera** (JB):

43.825

**Skew**: 1.411

**Prob(JB)**: 3.04e-10

Kurtosis: 6.615

Cond. No. 1.13e+04

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 1.13e+04. This might indicate that there are

strong multicollinearity or other numerical problems.

• Se obtiene que la R2 0.559 para la formula X1-X2+X3+X4+X5+X5+X7

Se crea función para plotear rápidamente las columnas

In [6]: def graficas\_plot(nom,x):

data\_plt=x

fig = plt.figure(figsize=(16, 6))

fig.suptitle(nom)

ax1 = fig.add\_subplot(221**)**

ax1.set\_title('Gatito')

ax1.boxplot(x, flierprops-dict(markerfacecolor='red', marker='0'), showmeans=True);

ax2 = fig.add\_subplot(222)

ax2.hist(data\_plt, density=**True, bins** -8**,** label="Data")

mn

data\_plt.min()

mx = data\_plt.max()

rv = ss.norm(np.mean (data\_plt), np.std (data\_plt))

x = np.linspace (mn,mx)

ax2.set\_title('Distribución")

ax2.plot(x, rv.pdf(x), lw=2, color="blue");

Se valida los valores de las columnas

In [7]: graficas\_plot("X1",df[ **'X1'**])

graficas\_plot("X2",df[ 'X2']) graficas\_plot("X3", df[ 'X3'])

graficas\_plot("x4", **df**[ 'X4'])

graficas\_plot("x5", df['x5'])

graficas\_plot("X6",df[ 'X6']) graficas\_plot("x7", df[ 'x7'])

1750

1500

1250

1000

750

500

3000

2000

1000

80

8

60-

40

N

20

70

60

8 2 8 8

50

**0**

X1

Gatito

O

X2

Gatito

o

H

Gatito

X3

X4

Gatito

X5

Gatito

**35**

30

25

201

15

10

5

8 8 9

20-

30

25

20

15

སྐ ཋ གླ གླུ ཤཱ ཋ

10

X6

Gatito

8

X7

Gatito

0.0015

0.0010-

0.0005

Distribución

0.0000

400

600

800

1000

1200

1400

1600

1800

0.0012

**0.0010**

0.0008

0.0006

0.0004

0.0002

0.0000

Distribución

-

500

1000

1500

2000

2500 3000 3500

Distribución

0.04

0.03

0.02

0.01

0.00

20

30

40

50

60

70

80

0.04

0.03

0.02

0.01

Distribución

0.00

45

50

55

60

65

70

75

80

0.08

0.06

0.04-

0.02

Distribución

0.00

15

20

30

0.04

0.03

0.02

0.01

Distribución

**0.00**

10

20

30

40

50

60

70

80

0.10

0.08

0.06

0.04

0.02

0.00

10

15

Distribución

**25**

30

• Se presenta que la mayoría de las variables predictoras estudiados tienen a tener OUTLINER con número excesimante grandes a diferencia de otras variables, esto se debe expresar a que el crimen y la consecuente lucha que se lleva contra ellas en ciertas ciudades de estados unidos se

encuentra disparada.

**Se** eliminó una **sección del** código **donde** validaba **si** era conveniente eliminar **outliner, se** llega a la conclusión **de** que quitar las **outliner** empeorá la calidad del modelo**,** esto **puede** estar causado por la poca cantidad de **variables del** modelo

In [8]:

df [df ['x2']>2000]

Out [8]:

**X1** X2 **X3**

**X4** X5 X6 X7

40

1740 3545 86 62

22 18 15

• Por ejemplo, validando este registro con un OUTLINER en la columna "X2 - Crimen violento cada 100K" (Máximo APROX esperado: 1700/Valor real 3545), podemos ver que también presenta OUTLINER en "X1 - Crimen reportado cada 1M " (Máximo APROX esperado:1000 / Valor real 1740**)** y "X3"

Pregunta 2:

Efectúe un análisis de regresión lineal múltiple que explique la tasa de crímenes reportados totales por cada 1 millón de habitantes. Para esto, implemente la metodología de stepwise hacia atrás o hacia adelante, para encontrar la mejor combinación de variables explicatorias. Puede emplear cualquier métrica de desempeño de modelo: RMSE, R2, AIC, o BIC. Ayuda: Todas las métricas, parámetros, y otros resultados relevantes pueden ser revisados a partir del objeto 'model en donde quedan almacenados al crear algún modelo de regresión

• Modelo de regresión sin parametros (MO), en esta parte no sabía si el modelo nulo debía ser X1 ~ PromedioY o X1 ~ 1+PromedioY, los intentos al usar esto dieron R2 0 o 1 dependiendo de la librería, por lo mismo la descarté de la conclusión que debía obtener

In [9]: df['y\_mean']=np.mean (y)

y= df ['x1']

me.summary(

me = sm.OLS(y,df['y\_mean']).fit()

Out [9]:

In [10]:

OLS Regression Results

Dep. Variable:

X1

R**-squared**: 0.000

Model:

Adj. R-squared:

Method:

Least Squares

**F**-statistic:

Date: Sat, 14 May 2022 Prob (F-statistic):

OLS

**0.000**

nan.

nan.

Time**:**

00:45:33

Log-Likelihood: -354.61

No. Observations:

50

AIC: 711.2

**Df** Residuals:

49

BIC: 713.1

0

nonrobust

Df Model**:**

Covariance Type:

coef **std** err t **P>|t|** [0.025 0.975**]**

y\_mean 1.0000 0.058 17.271 0.000 0.884 1.116

Durbin-Watson**:**

1.178

Omnibus: 21.722

**Prob(**Omnibus): 0.000 Jarque-**Bera (**JB)**:** 31.319

Notes:

Skew: 1.483

**Prob**(**JB)**: 1.58e-07

Kurtosis: 5.498

Cond. No.

**1.00**

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

mo\_1 **=** sm.OLS(y, sm.add\_constant (df['y\_mean'])).fit()

mo\_1.summary()

C:\Users\milan\anaconda\lib\site-packages\statsmodels\tsa\tsatools.py:142: FutureWarning: In a future version of pandas all arguments of concat except for the argument 'objs' will be keyword-only

**x** = pd.concat(x[::order], 1)

OLS Regression Results

Out [10]:

Dep. Variable:

Model:

Method:

In []:

X1

**R-**squared**:** 0.000

Adj. R-**squared**: 0.000

OLS

Least Squares

**F**-statistic:

**Date:** Sat, 14 May 2022 Prob (F-statistic):

nan.

nan

**Time:**

00:45:33

Log-Likelihood: -354.61

No. Observations:

50

AIC: 711.2

**Df** Residuals:

49

BIC: 713.1

0

nonrobust

Df Model:

Covariance Type:

coef std err t P>|t| [0.025 0.975]

y\_mean 1.0000 0.058 17.271 0.000 0.884 1.116

Durbin-Watson:

1.178

Omnibus: 21.722

**Prob**(Omnibus): 0.000 Jarque**-**Bera (JB): 31.319

Notes:

**Skew**: 1.483

Prob(JB): 1.58e-07

Kurtosis:

5.498

Cond. No.

**1.00**

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Se trata de hacer stepwise con la siguiente página hacia adelante usando la siguiente función recuperada desde un sitio web

import statsmodels.formula.api as smf

def forward\_selected (data**,** response):

"""Linear model designed by forward selection**.**

Parameters:

data pandas DataFrame with all possible predictors and response

response: string, name of response column in data

Returns**:**

model: an "optimal" fitted statsmodels linear model

with **an** intercept

selected by forward selection

evaluated by adjusted R-squared

remaining set(data.columns)

remaining.remove(response)

selected = []

current\_score, best\_new\_score = 0.0, 0.0

while remaining and current\_score == best\_new\_score:

scores\_with\_candidates = []

for candidate in remaining:

formula = "{} ~ {} + 1".format(response,

'+'.join(selected + [candidate]))

score smf.ols (formula, data).fit().rsquared\_adj

scores\_with\_candidates.append((score**,** candidate))

scores\_with\_candidates.sort()

best\_new\_score, best\_candidate = scores\_with\_candidates.pop()

**if** current\_score < best\_new\_score**:**

remaining.remove(best\_candidate)

selected.append(best\_candidate)

current\_score = best\_new\_score

formula = "{} ~ {} + 1".format(response,

+ '.join(selected))

model = smf.ols (formula, data).fit()

return model

model forward\_selected (df, '**X1')**

print (model.model.formula)

print (model.rsquared\_adj)

Por medio de la función se llego a las siguientes conclusiones

• La formula ideal es X1 ~ X2 + X3 + 1

• El R2 ajustado que se consigue es 0.5848856605529651

Siguiendo las lógicas explicadas en clases creé el siguiente código que termina generando un DataFrame con las formulas

x=df[['x2', 'X3', 'x4', 'X5', 'X6', 'X7"**]]**

y=df['x1']

In []:

lista={}

r2={}

In []:

pos\_f=0

for pos,oi in enumerate(x):

temp list(combinations(x, 1+pos))

for posc,io in enumerate(temp):

rt=[]

for posh, ip in enumerate(io):

rt.append(ip)

apo sm.OLS (y, sm.add\_constant(x[rt])).fit()

apo.model.endog\_names apo.model.exog\_names

x\_concat=""

for i in apo.model.exog\_names:

x\_concat= x\_concat+i+"+"

**lista**[pos\_f]=apo.model.endog\_names+"~"+x\_concat

r2[pos\_f]=apo.rsquared\_adj

pos\_f=pos\_f+1

d = { 'Formula': lista, 'r2\_adj': r2}

dg= pd.DataFrame(data=d)

dg.sort\_values(by=['r2\_adj'], ascending=False).head()

Por medio del ejercicio llegue a concluir que la formula y el R2 Ajustado propuestos por el código es correcto, consiguiendo exactamente el mismo resultado.

Pregunta 3

Muestre el resumen (summary) del modelo escogido y describa sus principales características: - R2 - Estadístico F - Estadístico t

• A continuación se muestra el sumario de la formula ganadora:

fin **=** sm.OLS(y, sm.add\_constant(df[['x2', 'X3']])).fit()

fin.summary()

Conclusiones

• El mejor R2\_adj es 0.585

• EL estadístico F es los bastante bajo para no entregar preocupación por los parametros.

• Los estadístico t son adecuados, aunque para X3 esta un poco baja.

• El valor de P para X3 es un poco más alta que 0,05.

Pregunta 4:

Comente sus resultados en función del problema. ¿Cuáles son las variables que mejor explican el problema de la tasa de crimen en las ciudades de Estados Unidos?

• Me fijo que las variables del dataset las podríamos separar en 2 tipos:

• las variables de "Abandono social" asociadas a factores de crimen y tasas de no educación en la población(X1,X2,X3,X5).

■Las variables de "Diligencia social" asociadas principalmente a tasas sobre la educación en niveles medios y universitarios(X4,X6,X7).

• Habiendo una correlación alta entre X1 con **las** variables de abandono social (además que la combinación de variables en la regresión toman estas mismas variables) es claro que la falta de oportunidades para obtener una educación adecuada aumenta las tasas de criminalidad de las

ciudades.

■La falta de educación puede estar coartando las chances de superación de las personas.

In [ ]= df [df['x7' ]>30]

• Por ejemplo, en este registro donde vemos una tasa alta de personas de + **de** 25 años con estudios universitarios vemos que los valores de abandono social se encuentran dentro de los rangos esperados para su distribución, disponibles en el punto 1.

• Por el contrario puedo rescatar el mismo punto que vimos antes al inicio del documento y ver que los OUTLINER en variables de abandono social al ser altos dejan en rangos "aceptables/a la baja" en las variables de diligencia social.

In []: df [df['X2"]**>**2000]

• Sumado a que intente eliminar los OUTLINER dando como resultado el empobrecimiento del desempeño del dataset, da entender que estas variables son importantes.