| DATOS PERSONALES | FIRMA |
| --- | --- |
| |  |  | | --- | --- | | Nombre: Vanessa Alexandra | DNI:0603584061 | |  |
| Apellidos: Vásconez Nuñez |

| ESTUDIO | ASIGNATURA | CONVOCATORIA |
| --- | --- | --- |
| MÁSTER UNIVERSITARIO EN INGENIERÍA MATEMÁTICA Y COMPUTACIÓN (PLAN 2016) | 4391020006.- TÉCNICAS MULTIVARIANTES | Ordinaria Número periodo 1823 |

| FECHA | MODELO | CIUDAD DEL EXAMEN |
| --- | --- | --- |
| 14-16/01/2022 | Modelo - D | Riobamba - Ecuador |

| Etiqueta identificativa |
| --- |
|  |

**INSTRUCCIONES GENERALES**

1. Ten disponible tu documentación oficial para identificarte, en el caso de que se te solicite.
2. Si tu examen consta de una parte tipo test, indica las respuestas en la plantilla según las características de este.
3. Debes contestar en el documento adjunto, respetando en todo momento el espaciado indicado para cada pregunta. Si este es en formato digital, los márgenes, el interlineado, fuente y tamaño de letra vienen dados por defecto y no deben modificarse. En cualquier caso, asegúrate de que la presentación es suficientemente clara y legible. Entrega toda la documentación relativa al examen, revisando con detenimiento que los archivos o documentos son los correctos. El envío de archivos erróneos o un envío incompleto supondrá una calificación de “no presentado”.
4. Durante el examen y en la corrección por parte del docente, se aplicará el Reglamento de Evaluación Académica de UNIR que regula las consecuencias derivadas de las posibles irregularidades y prácticas académicas incorrectas con relación al plagio y uso inadecuado de materiales y recursos.
5. No está permitido el uso de Internet ni ningún tipo de comunicación con otra persona.Durante todo el examen tu teléfono móvil debe estar en modo avión.
6. La parte principal de cada pregunta consiste en interpretar y comentar los resultados obtenidos. Si te limitas a hacer los cálculos no vas a poder superar el examen.
7. Es fundamental que las respuestas estén debidamente redactadas, de forma clara y precisa y sin faltas de ortografía.
8. Para hacer el examen puedes utilizar los apuntes del curso y los scripts que hayas preparado y Python para hacer los cálculos.

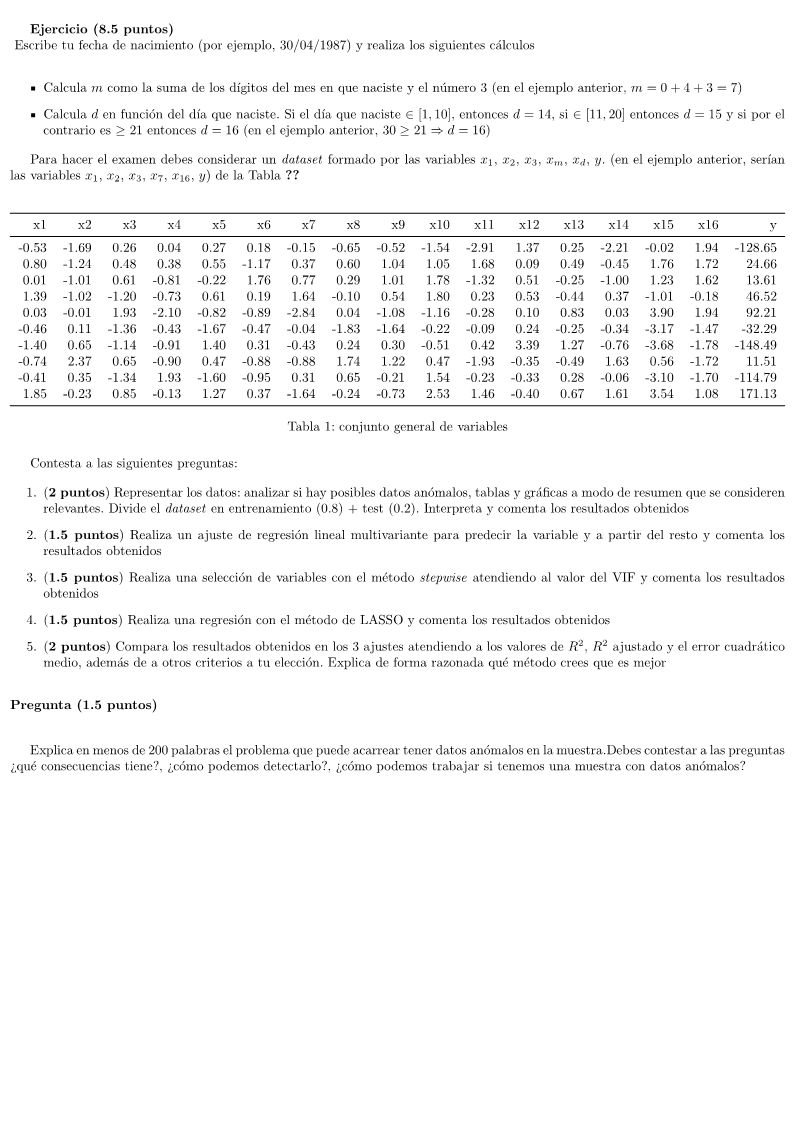
**Puntuación**

**Preguntas**

* Puntuación máxima 10.00 puntos

El examen constará de un ejercicio práctico (8,5 puntos) y una pregunta teórica (1,5 puntos). Los enunciados están en la página 14 y el espacio para responder el examen está entre las práginas 4 y 13.  
  
**1.** Pregunta

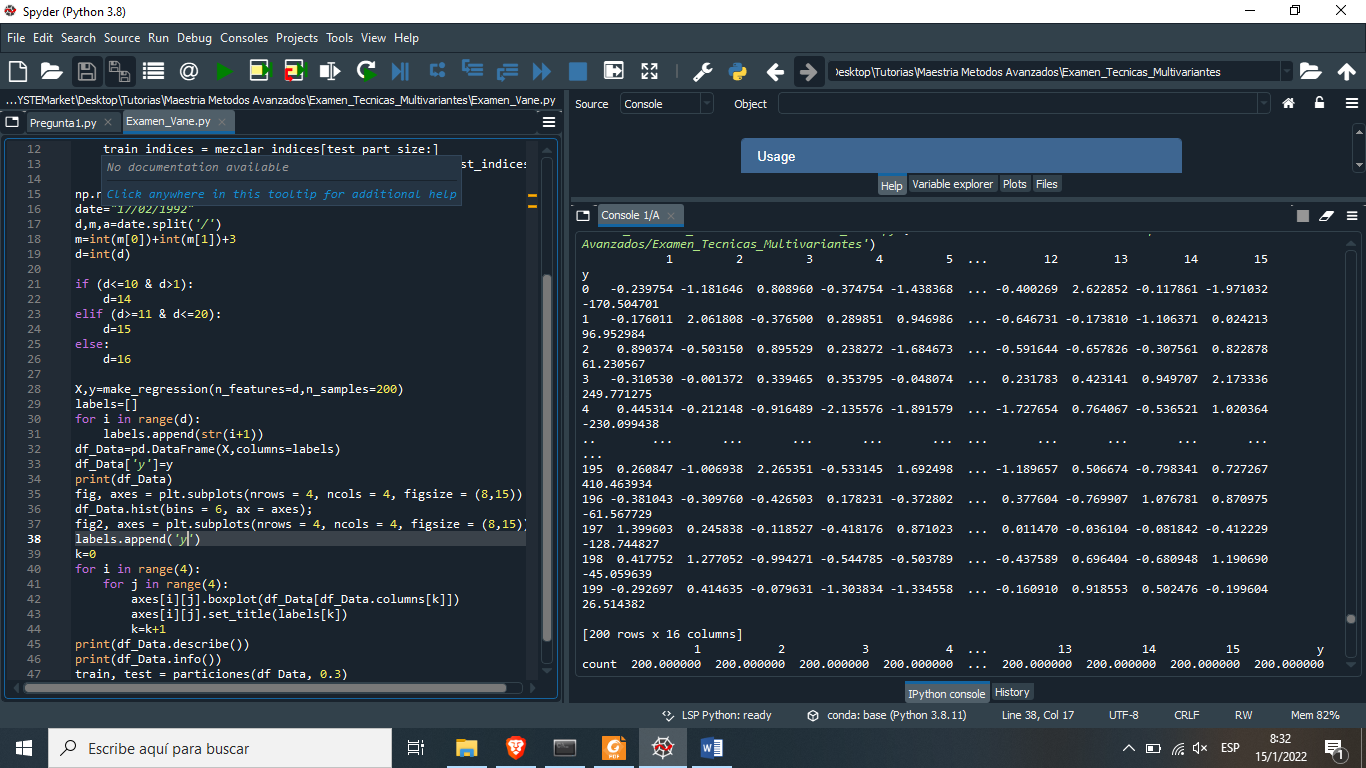
 (Responder en 10 caras)



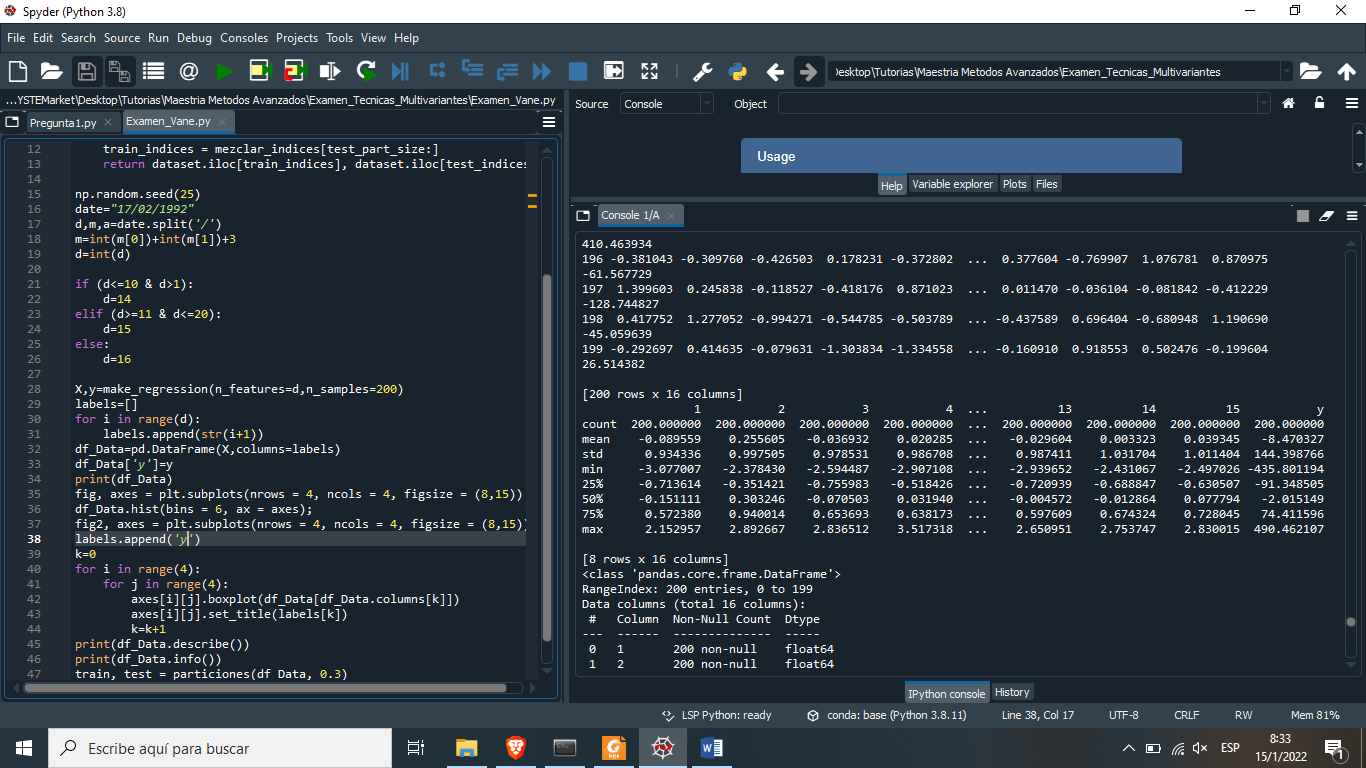
**PREGUNTA 1**

Para realizar este apartado se crean variables utilizando la función make\_regression con 200 observaciones y en este caso con 14 variables para la regresión lineal. Se genera un dataset con 14 variables, de los cuales graficamos los histogramas, los boxplots con los cuales se pueden observar los datos anómales y para describir las variables usamos los métodos describe e info.

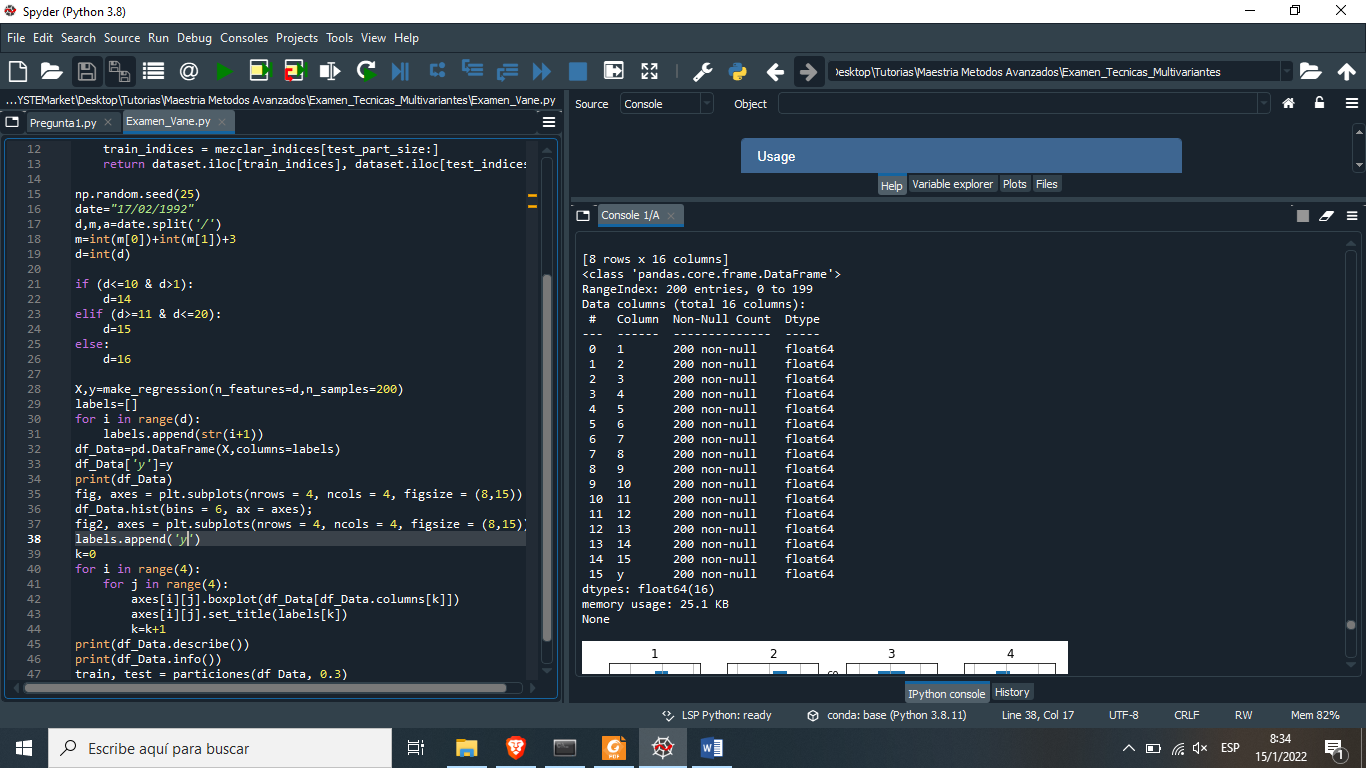
**Dataset generado**



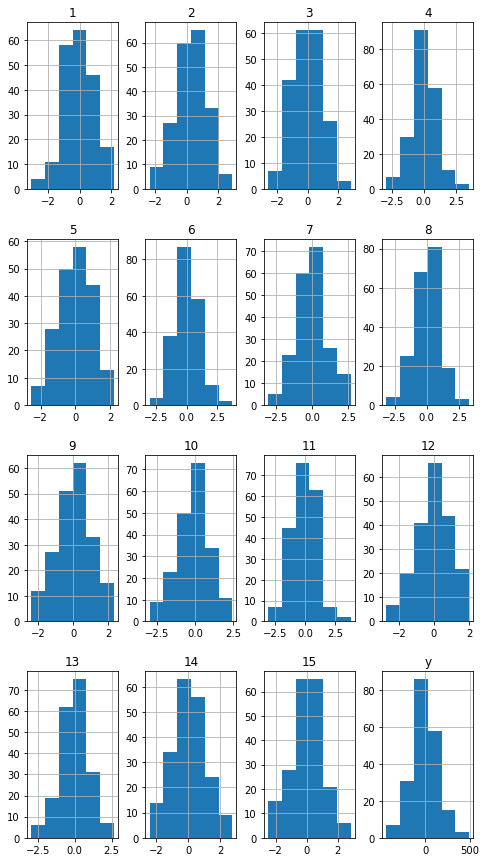
**Describe()**



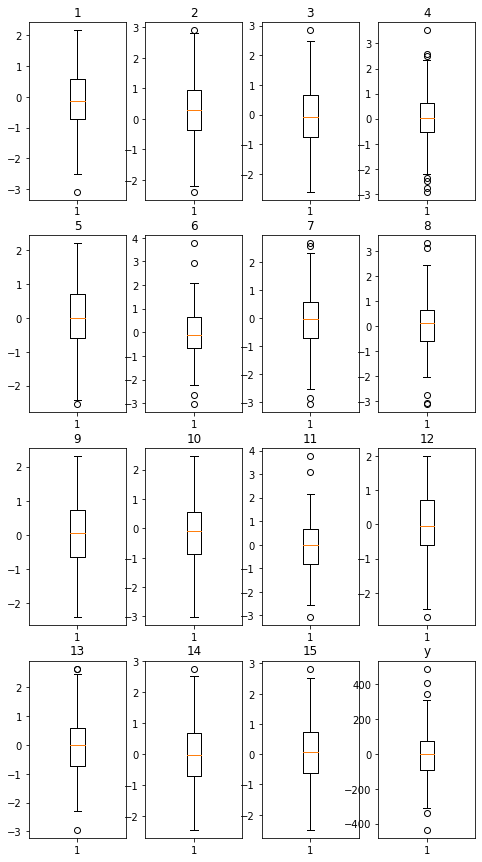
**Info()**



**Histogramas**



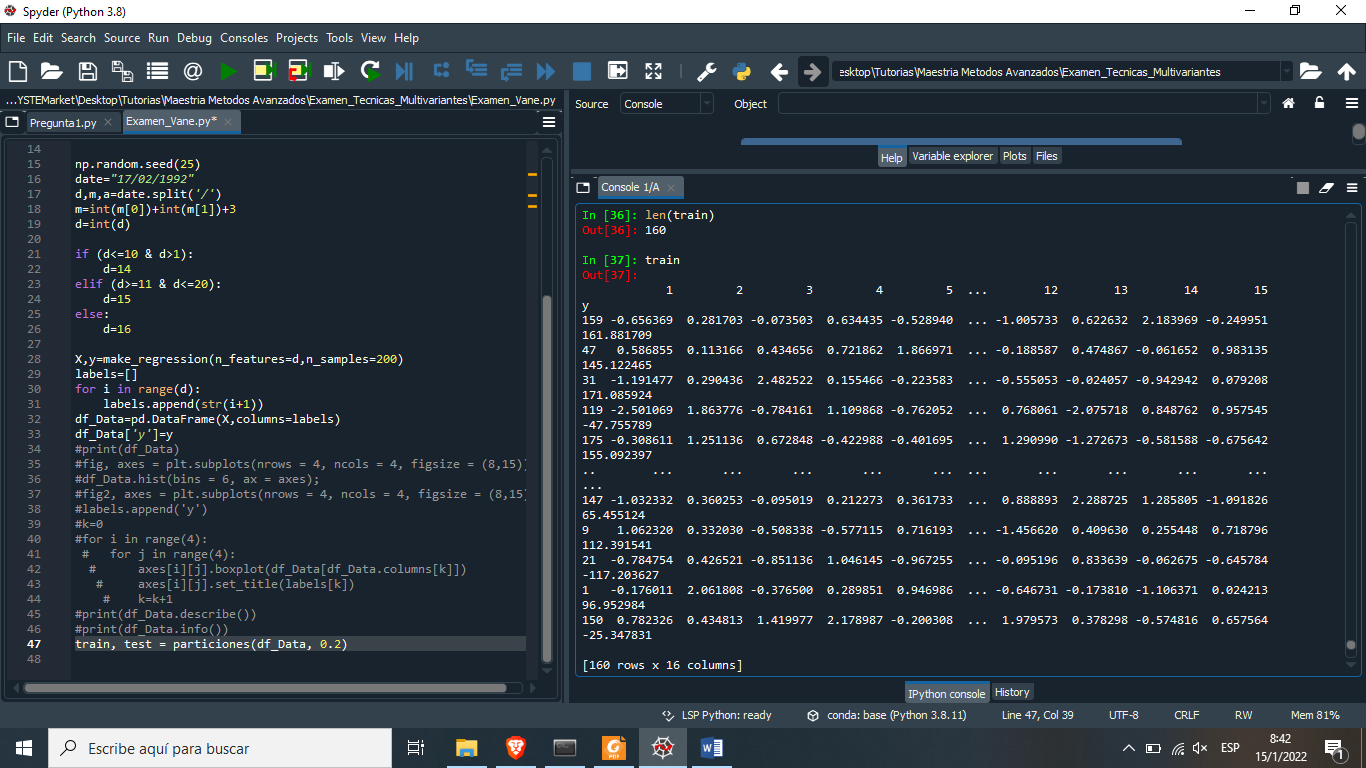
**Boxplots**



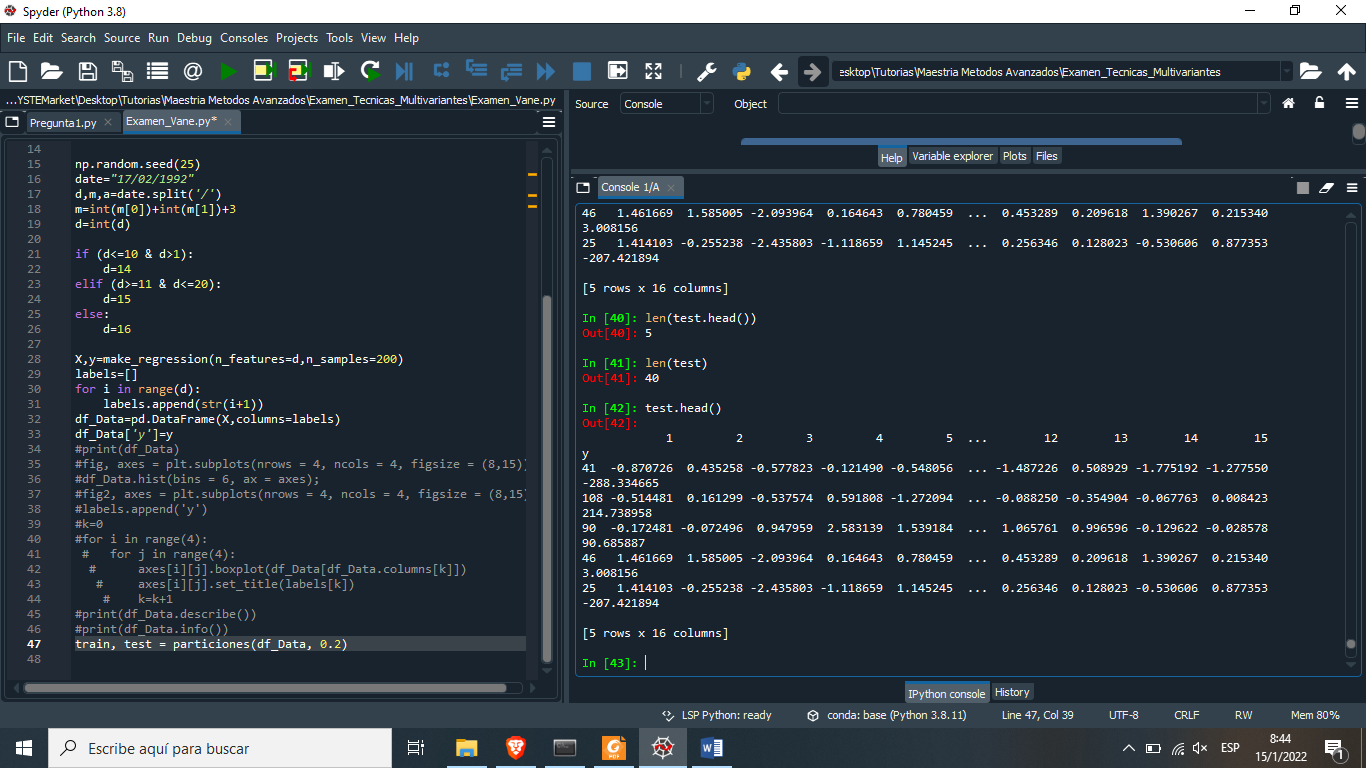
Se puede observar que las varialbes de entrada del modelo tienen una media cercana a 0 mientras que la variable de respuesta tiene una media alrededor de -8. En la tabla de info() tenemos que todas las variables tiene conteo no-Null lo cual es consistenete y ademas tiene las misma cantidad de 200, por lo que no hay ninguna variable con datos faltantes, asi que no hace falta usar un imputador. Vemos que todos los parámetros y la varaibles de respeusta tiene una distribución noraml gracias a los histogramas y por otro lado con los boxplots podemos observar de mejor manera los datos anómalos que este caso la variable x4 es la quemas presenta este tipo de datos.

Realizamos la división del dataset

**Data\_train**

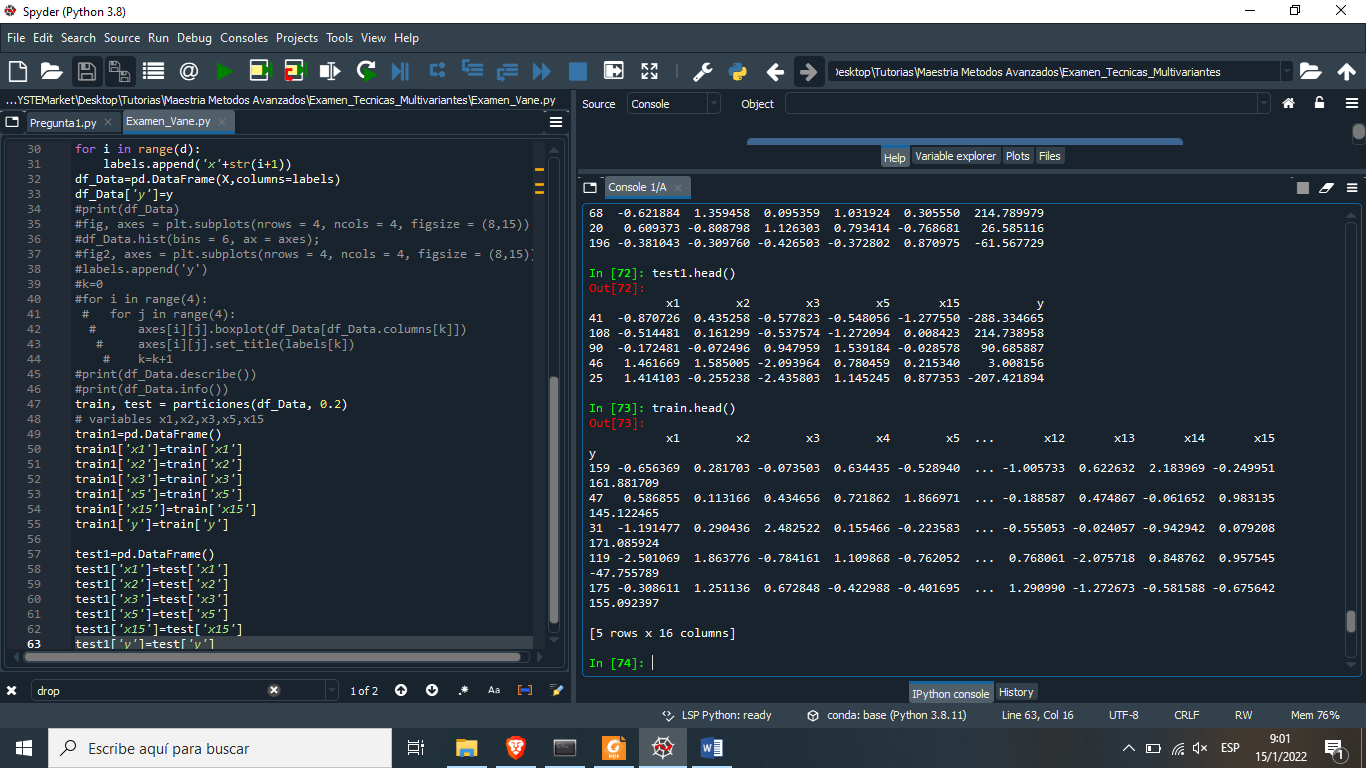


**Data\_test**

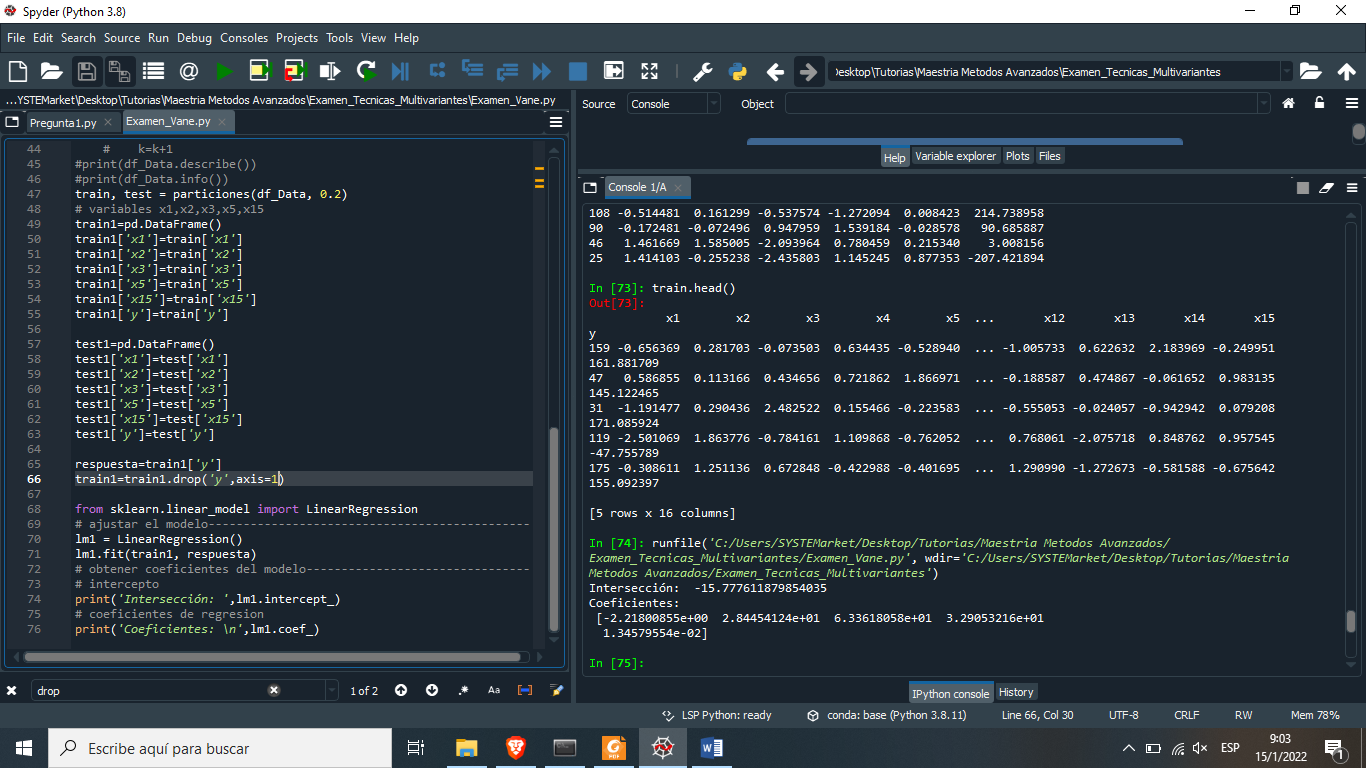


**PREGUNTA 2**

Escogemos las variables x1,x2,x3,x5,x15



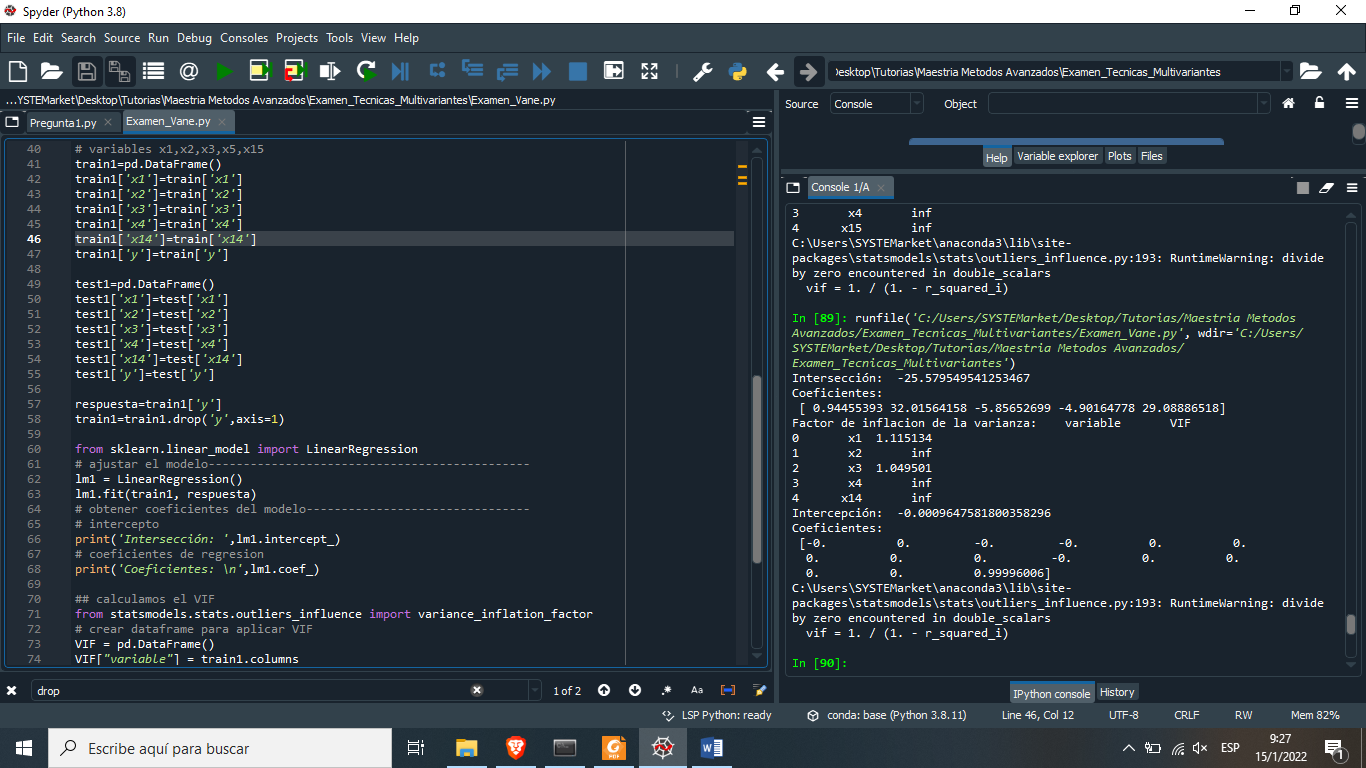
Realizamos la regresion lineal usando el método Liner\_Regressión y obtenemos los siguientes resultados



Entonces la ecuación que ajusta el modelo será

**PREGUNTA 3**

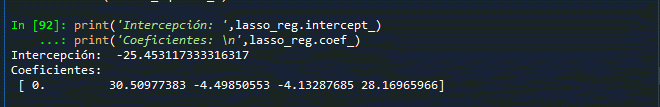
Calculamos el Vif para cada varialbe y nos dieron los siguientes resultados:



Se observa que las ´´unicas variables que no tiene colinealidad son las varialbes x1 y x2, por lo que solo escogemos estos valores para realizar el ajuste de esta manera nuestra ecuación lineal del modelo será

**PREGUNTA 4**

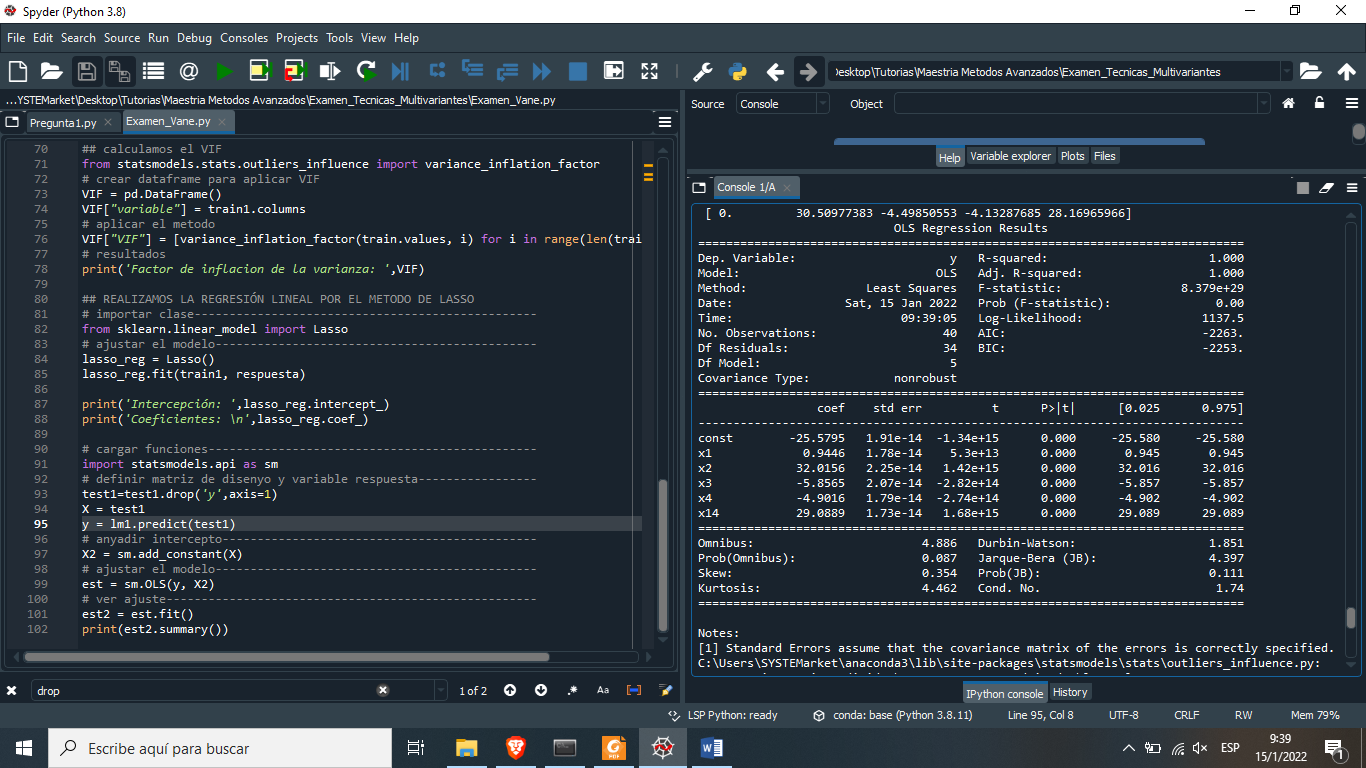
Realizamo el ajuste de regresión lienal multiple usando el método de LASSO y nos queda los siguientes resultados



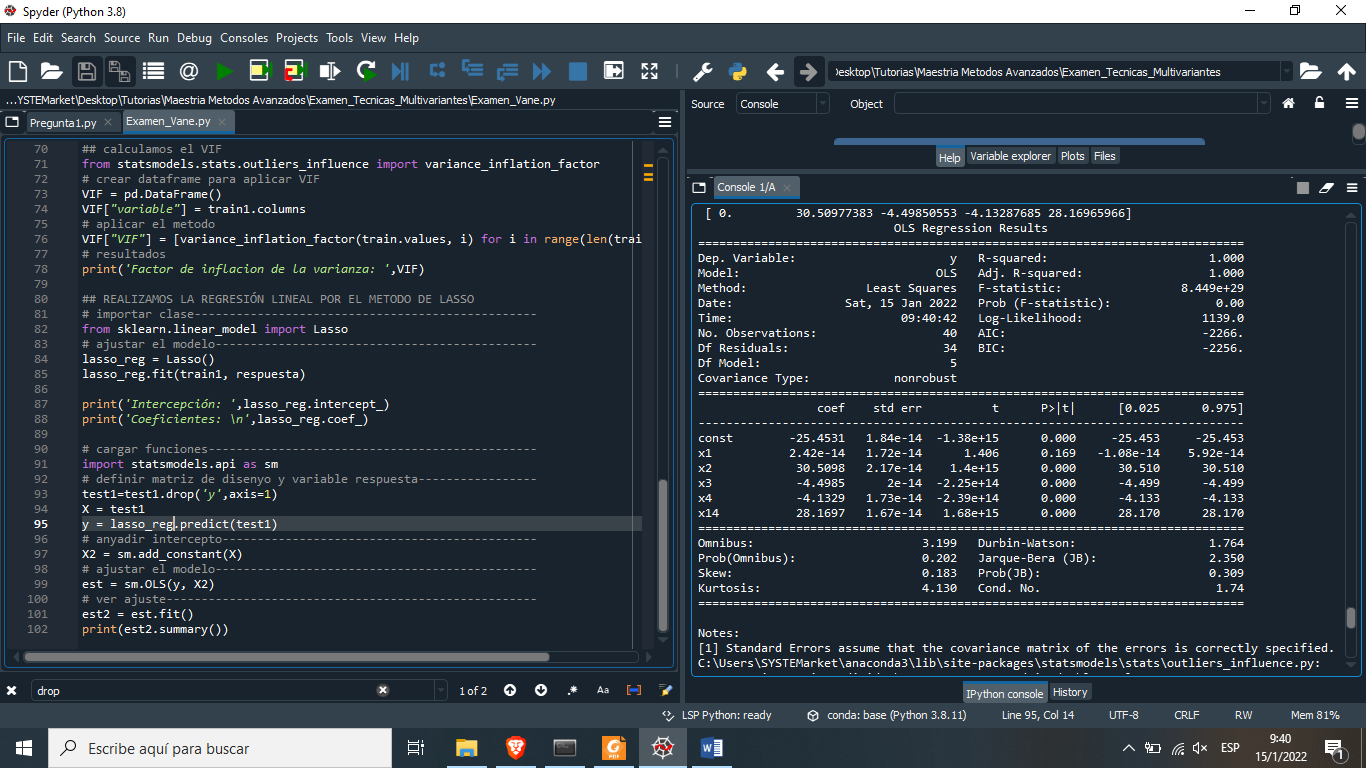
Por lo que el resultado de la acuación lienal que ajusta el modelo es:

**PREGUNTA 5**

Con el modelo de regresión lineal



Con el modelo de LASSO



**PREGUNTA 2**

Los datos anómalos consisten en uno o mas puntos que se encuentra por fuera de los rangos normales de la regresión, lo cuales provocan que los resultados de los cálculos estadísticos provoquen varias fallas, para poder detectar estos valores es necesario realizar un análisis gráfico de todos los puntos de la muestra y confirmar si todos estan agrupados en una forma como constante, lineal, curva, etc., y poder visualizar los terminos excluidos, para poder trabajar con este tipo de datos es necesario aplicar diferentes modelos de regresión como la lineal, el cual ayuda a detectar este y eliminarlo, de no poder hacerlo se deberá cambiar de modelo, por lo general cuando se trabaja con datos multivariantes estos datos anómalos son mas dificil de detectar y eliminar, aunque es posible intentar realizar la regresión penalizada, como las del tipo Ridge y Lasso, y la red elástica, sin embargo de no poder eliminarse es preferible anularlo de forma manual.

**ANEXO: Codigo en Python del problema**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.datasets import make\_regression

def particiones(dataset, test\_part):

test\_part\_size = int(len(dataset) \* test\_part)

mezclar\_indices = np.random.permutation(len(dataset))

test\_indices = mezclar\_indices[:test\_part\_size]

train\_indices = mezclar\_indices[test\_part\_size:]

return dataset.iloc[train\_indices], dataset.iloc[test\_indices]

np.random.seed(25)

date="07/10/1986"

d=14

m=4

X,y=make\_regression(n\_features=d,n\_samples=200)

labels=[]

for i in range(d):

labels.append('x'+str(i+1))

df\_Data=pd.DataFrame(X,columns=labels)

df\_Data['y']=y

print(df\_Data)

fig, axes = plt.subplots(nrows = 4, ncols = 4, figsize = (8,15))

df\_Data.hist(bins = 6, ax = axes);

fig2, axes = plt.subplots(nrows = 4, ncols = 4, figsize = (8,15))

labels.append('y')

k=0

for i in range(4):

for j in range(4):

axes[i][j].boxplot(df\_Data[df\_Data.columns[k]])

axes[i][j].set\_title(labels[k])

k=k+1

print(df\_Data.describe())

print(df\_Data.info())

train, test = particiones(df\_Data, 0.2)

# variables x1,x2,x3,x5,x15

train1=pd.DataFrame()

train1['x1']=train['x1']

train1['x2']=train['x2']

train1['x3']=train['x3']

train1['x4']=train['x4']

train1['x14']=train['x14']

train1['y']=train['y']

test1=pd.DataFrame()

test1['x1']=test['x1']

test1['x2']=test['x2']

test1['x3']=test['x3']

test1['x4']=test['x4']

test1['x14']=test['x14']

test1['y']=test['y']

respuesta=train1['y']

train1=train1.drop('y',axis=1)

from sklearn.linear\_model import LinearRegression

# ajustar el modelo----------------------------------------------

lm1 = LinearRegression()

lm1.fit(train1, respuesta)

# obtener coeficientes del modelo--------------------------------

# intercepto

print('Intersección: ',lm1.intercept\_)

# coeficientes de regresion

print('Coeficientes: \n',lm1.coef\_)

## calculamos el VIF

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

# crear dataframe para aplicar VIF

VIF = pd.DataFrame()

VIF["variable"] = train1.columns

# aplicar el metodo

VIF["VIF"] = [variance\_inflation\_factor(train.values, i) for i in range(len(train1.columns))]

# resultados

print('Factor de inflacion de la varianza: ',VIF)

## REALIZAMOS LA REGRESIÓN LINEAL POR EL METODO DE LASSO

# importar clase-------------------------------------------------

from sklearn.linear\_model import Lasso

# ajustar el modelo----------------------------------------------

lasso\_reg = Lasso()

lasso\_reg.fit(train1, respuesta)

print('Intercepción: ',lasso\_reg.intercept\_)

print('Coeficientes: \n',lasso\_reg.coef\_)

# cargar funciones-----------------------------------------------

import statsmodels.api as sm

# definir matriz de disenyo y variable respuesta-----------------

test1=lm1.drop('y',axis=1)

X = test1

y = lasso\_reg.predict(test1)

# anyadir intercepto---------------------------------------------

X2 = sm.add\_constant(X)

# ajustar el modelo----------------------------------------------

est = sm.OLS(y, X2)

# ver ajuste-----------------------------------------------------

est2 = est.fit()

print(est2.summary())

# cargar funciones-----------------------------------------------

import statsmodels.api as sm

# definir matriz de disenyo y variable respuesta-----------------

test1=test1.drop('y',axis=1)

X = test1

y = lasso\_reg.predict(test1)

# anyadir intercepto---------------------------------------------

X2 = sm.add\_constant(X)

# ajustar el modelo----------------------------------------------

est = sm.OLS(y, X2)

# ver ajuste-----------------------------------------------------

est2 = est.fit()

print(est2.summary())