Minería de datos G. 571

Conociendo tus datos

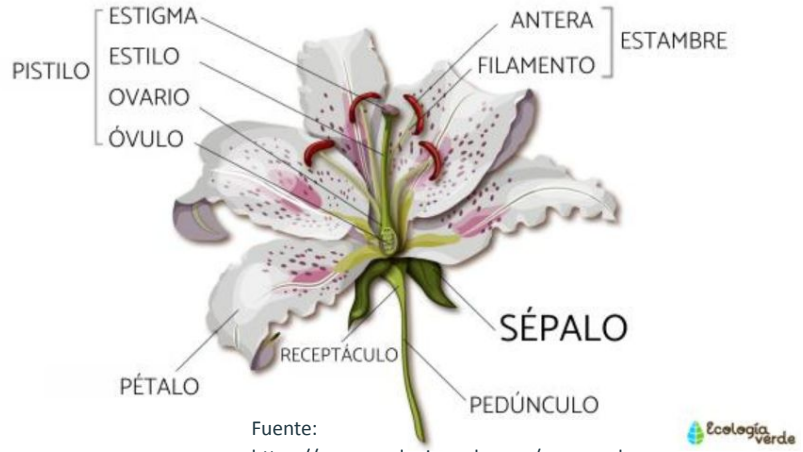
Zavala Román Irvin Eduardo 1270771

Dataset sobre flores Iris

Este dataset consiste en 3 tipos de plantas iris con 4 propiedades de cada flor relacionado a sépalos y pétalos.

```
iris_dataset = pd.read_csv("iris.csv")
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
..	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica



Fuente: https://www.ecologiaverde.com/que-son-los-sepalos-y-su-funcion-3231.html

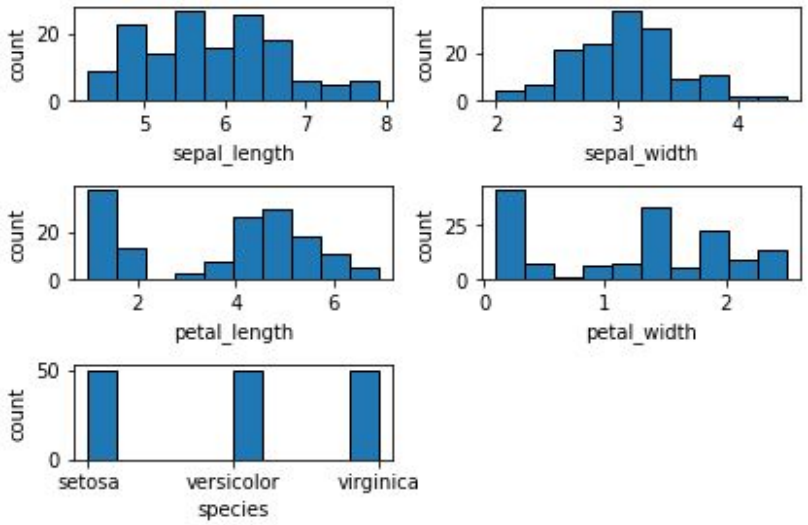
Dataset sobre flores Iris: Estadísticas

```
print(iris_dataset.describe())
```

	sepal length	sepal width	petal length	petal width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Dataset sobre flores Iris: Histogramas

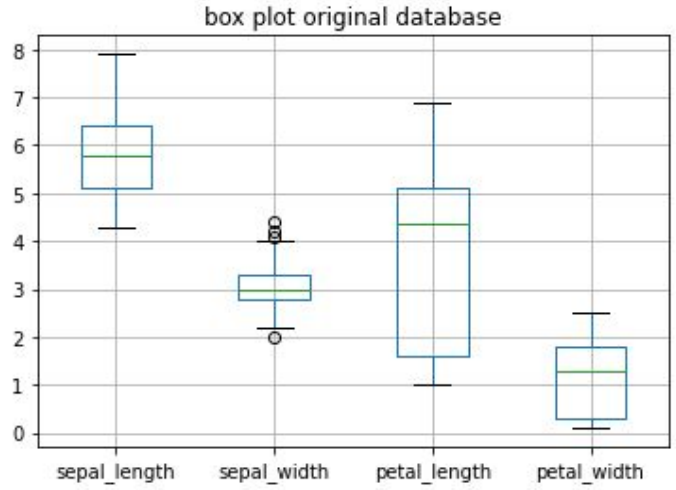
```
atributos = list(iris_dataset.columns.values )
for i in range(len(atributos)):
    plt.subplot(3, 2, i+1)
    plt.hist(iris_dataset[atributos[i]], bins=10, edgecolor='black')
    plt.xlabel(atributos[i])
    plt.ylabel('count')
fig.tight_layout(pad = 1.2)
plt.show()
```



En los atributos del sépalo se tiene una tendencia ligeramente a la derecha, siendo casi simétrica en en width. En los pétalos se puede ver una distribución bimodal.

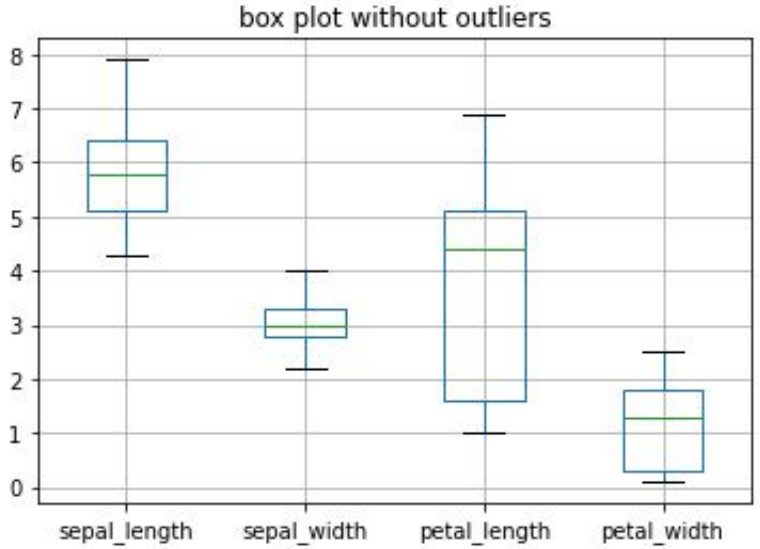
Dataset sobre flores Iris: Boxplot

```
iris_dataset.boxplot(column=['sepal_length', 'sepal_width', 'petal_length', 'petal_width'])
plt.title("box plot original database")
plt.show()
```



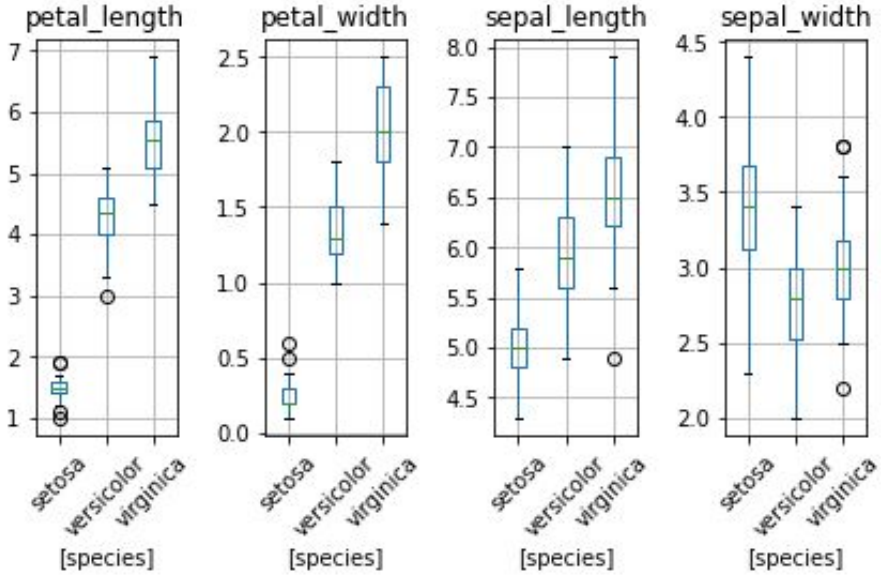
Dataset sobre flores Iris: Outliers

```
iris_dataset_sin_anomalias = iris_dataset
for i in atributos:
    if(i == "species"):
        break
    q1 = iris_dataset[i].quantile(0.25)
    q3  = iris_dataset[i].quantile(0.75)
    qr = q3-q1
    q3 = q3+1.5*qr
    q1 = q1-1.5*qr
    iris_dataset_sin_anomalias = iris_dataset_sin_anomalias[(iris_dataset_sin_anomalias[i] < q3) & (iris_dataset_sin_anomalias[i] > q1)]
```



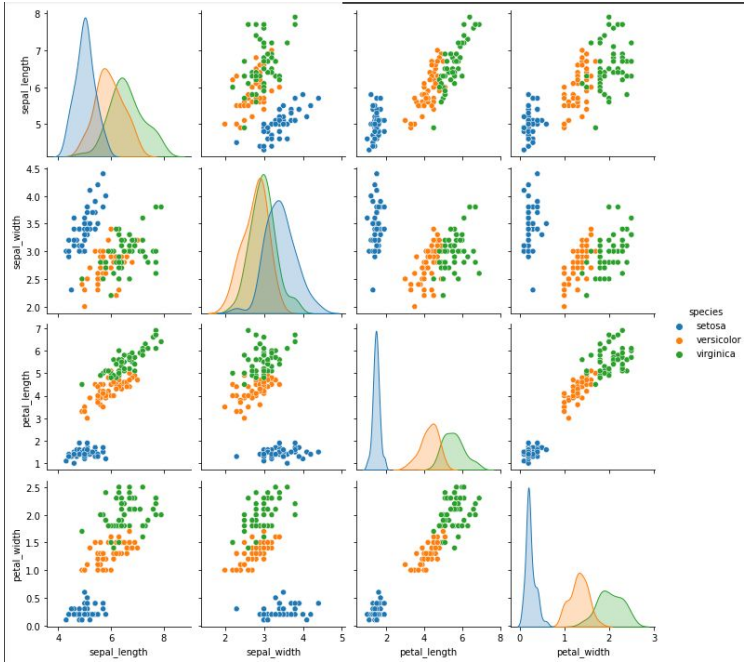
Dataset sobre flores Iris: Otro boxplot

```
fig, axes = plt.subplots(1,4,sharex=False,sharey=False)
iris_dataset.boxplot(by="species", ax=axes, rot=45)
fig.suptitle('')
fig.tight_layout(pad = 1.2)
plt.show()
```



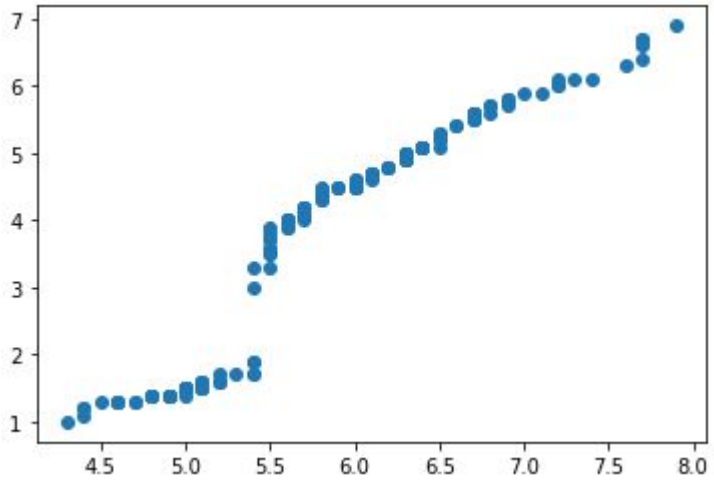
Dataset sobre flores Iris: Pairplot

```
#import seaborn as sn
sn.pairplot(iris_dataset,
                hue="species")
plt.show()
```



Dataset sobre flores Iris: Intento qq plot

```
sepal = iris_dataset[['sepal_length']].sort_values(by=['sepal_length'])
petal = iris_dataset[['petal_length']].sort_values(by=['petal_length'])
plt.plot(sepal,petal, "o")
plt.show()
```



Dataset sobre flores Iris
¿Faltan datos? ¿Cosas interesantes?

Hablando de atributos no agregaría más cosas ya que no tengo ninguna hipótesis que quiera comprobar respecto a tamaños de las partes de las flores, pero si agregaría más especies de Iris para tener un dataset más robusto y que no depende de solo 3 tipos de flores. Se me hace curioso como los datos relacionados al sépalo son más normales respecto al pétalo, llegando a crear distribuciones bimodales.

Dataset sobre zapatos

Consiste en una lista de información de 23,940 zapatos para hombres por la base de datos de Amazon. El dataset incluye nombre, marca, cantidad vendidos, precio, detalles y rating. Es importante limpiar los datos para su buen procesamiento.

```
men_shoes = pd.read_csv("MEN_SHOES.csv")
atributos = list(men_shoes.columns.values)
#Un poquito de limpieza
men_shoes = men_shoes.replace('[^A-Za-z0-9]+','', regex=True)
men_shoes.columns = men_shoes.columns.str.replace(' ', '')
#Se convierten a valores numericos para evitar cosas raras
men_shoes['How_Many_Sold'] = pd.to_numeric(men_shoes['How_Many_Sold'])
men_shoes['Current_Price'] = pd.to_numeric(men_shoes['Current_Price'])
#Hacemos algo con NaNs
men_shoes = men_shoes.dropna()
```

Dataset sobre zapatos: Estadísticas

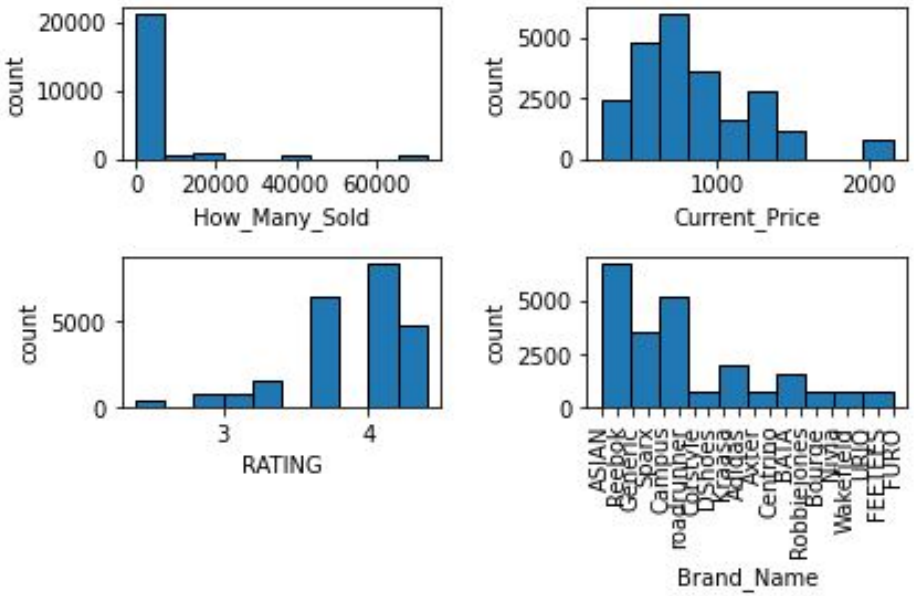
```
#Estadisticas del dataset
print(men_shoes.describe())
```

	How Many Sold	Current Price	RATING
count	23142.000000	23142.000000	23142.00000
mean	3607.896552	842.258621	3.82069
std	10896.836132	387.523381	0.40462
min	2.000000	231.000000	2.40000
25%	173.000000	588.000000	3.60000
50%	406.500000	776.500000	4.00000
75%	1795.000000	1080.000000	4.00000
max	72611.000000	2159.000000	4.40000

Dataset sobre zapatos: Histogramas

```
#Histogramas
fig = plt.figure()
men_shoes_sin_anomalias = men_shoes.select_dtypes$['number'])
atributos_sa = list(men_shoes_sin_anomalias.columns.values$
for i in range(len(atributos_sa)):
    plt.subplot(2, 2, i+1)
    plt.hist(men_shoes_sin_anomalias[atributos_sa[i]], bins=10, edgecolor='black')
    plt.xlabel(atributos_sa[i])
    plt.ylabel('count')
plt.subplot(2, 2, 4)
plt.hist(
      men_shoes['Brand_Name'], bins=10, edgecolor='black')
plt.xlabel('Brand_Name')
plt.xticks(rotation ='vertical')
plt.ylabel('count')
fig.tight_layout(pad = 1.2)
plt.show()
```

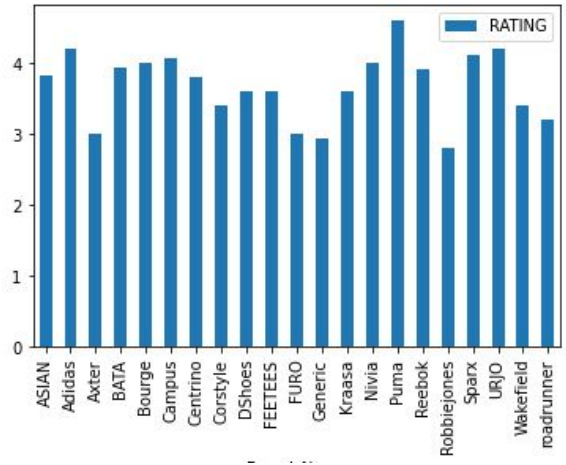
Dataset sobre zapatos: Histogramas

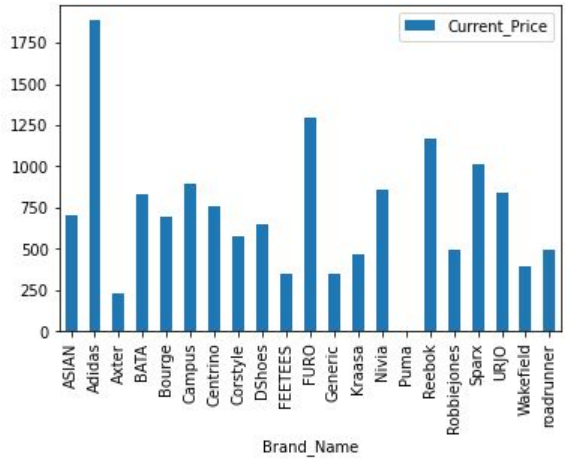


Se puede ver que la mayoría de tenis no pasan de 10,000 ventas. El precio es asimétrico a la derecha, el rating asimétrico a la izquierda y que la marca Asian es una de las marcas con más calzado en el dataset.

Dataset sobre zapatos: Barplots

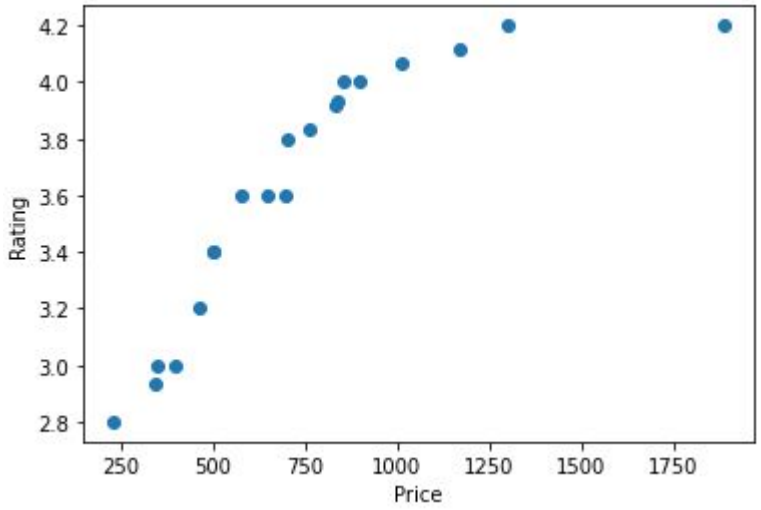
```
#Barplots
dataset = dataset.groupby('Brand_Name').mean().reset_index()
dataset.plot('Brand_Name', 'RATING', kind='bar')
plt.show()
dataset.plot('Brand_Name', 'Current_Price', kind='bar')
plt.show()
```





Dataset sobre zapatos: Calidad/Precio

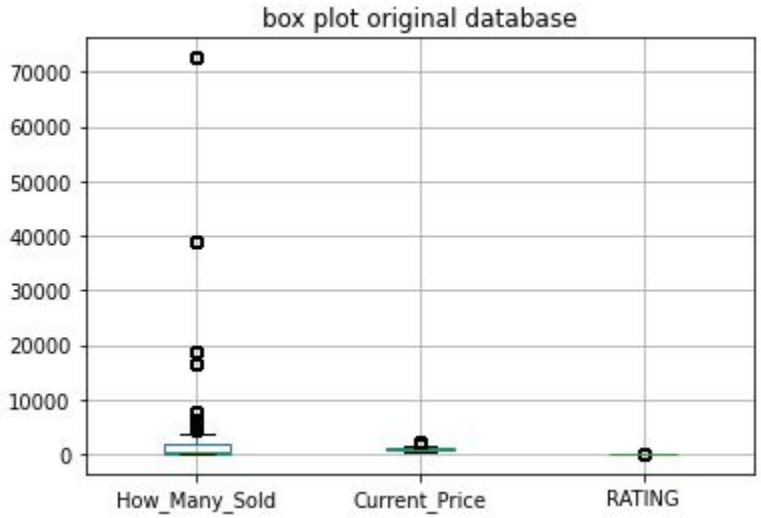
```
rating = dataset[['RATING']].sort_values(by=['RATING'])
price = dataset[['Current_Price']].sort_values(by=['Current_Price'])
plt.plot(price,rating, "o")
plt.xlabel('Price')
plt.ylabel('Rating')
plt.show()
```

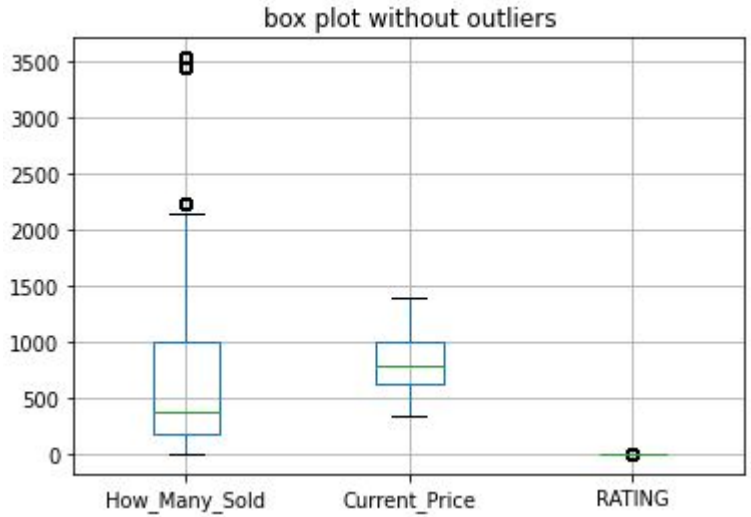


Dataset sobre zapatos: Boxplots

```
#Se van a usar las columnas numericas para quitar outliers
men_shoes_sin_anomalias = men_shoes.select_dtypes (['number'])
atributos_sa = list(men_shoes_sin_anomalias.columns.values )
#Boxplot original
men_shoes.boxplot (column=['How_Many_Sold', 'Current_Price', 'RATING'])
plt.title("box plot original database" )
plt.show()
#Se quitan outliers con metodo cuantiles de la presentacion
for i in atributos_sa:
    q1 = men_shoes_sin_anomalias [i].quantile(0.25)
    q3  = men_shoes_sin_anomalias [i].quantile(0.75)
    qr = q3-q1
    q3 = q3+1.5*qr
    q1 = q1-1.5*qr
    men_shoes_sin_anomalias = men_shoes_sin_anomalias [(men_shoes_sin_anomalias [i] < q3) &
(men_shoes_sin_anomalias [i] > q1)]
#Boxplot sin outliers
men_shoes_sin_anomalias.boxplot (column=['How_Many_Sold', 'Current_Price', 'RATING'])
plt.title("box plot without outliers" )
plt.show()
```

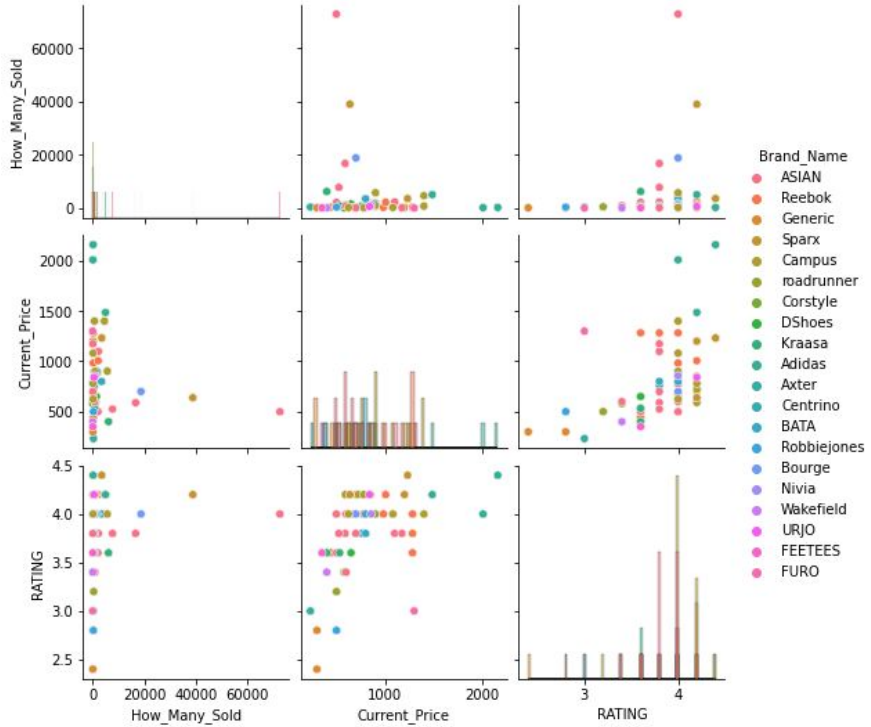
Dataset sobre zapatos: Boxplots





Dataset sobre zapatos: Pairplot

```
#Pairplot respecto a marcas,
IMPORTANTENO CAMBIAR KIND Y DIAG_KIND
sn.pairplot(
    men_shoes,
    hue="Brand_Name",
    kind = "scatter",
    diag_kind= "hist")
plt.show()
```



Dataset sobre flores Zapatos
¿Faltan datos? ¿Cosas interesantes?

Yo no le agregaría nada al dataset, pero si le quitaria la columna de ‘detalles del producto’ ya que estorba el realizar el análisis y preferiría que fuera un atributo numérico. Se me hizo curiosa la relación que hay entre el precio/rating ya que no existe un calzado que se salga de las normas siendo barato y con un rating alto.

Dataset sobre felicidad (2015)

Consiste en una evaluación de varios factores que influyen en la felicidad en una escala de 0 a 10.

```
happiness = pd.read_csv("2015.csv")
atributos = list(happiness.columns.values)
```

Dataset sobre felicidad (2015): Estadísticas

```
print(happiness.describe())
```

	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)
count	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000
mean	79.493671	5.375734	0.047885	0.846137	0.991046	0.630259
std	45.754363	1.145010	0.017146	0.403121	0.272369	0.247078
min	1.000000	2.839000	0.018480	0.000000	0.000000	0.000000
25%	40.250000	4.526000	0.037268	0.545808	0.856823	0.439185
50%	79.500000	5.232500	0.043940	0.910245	1.029510	0.696705
75%	118.750000	6.243750	0.052300	1.158448	1.214405	0.811013
max	158.000000	7.587000	0.136930	1.690420	1.402230	1.025250

	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
count	158.000000	158.000000	158.000000	158.000000
mean	0.428615	0.143422	0.237296	2.098977
std	0.150693	0.120034	0.126685	0.553550
min	0.000000	0.000000	0.000000	0.328580
25%	0.328330	0.061675	0.150553	1.759410
50%	0.435515	0.107220	0.216130	2.095415
75%	0.549092	0.180255	0.309883	2.462415
max	0.669730	0.551910	0.795880	3.602140

Dataset sobre felicidad (2015): Histogramas

```
hapiness = happiness.select_dtypes (['number'])
happiness_sin_anomalias = happiness.select_dtypes (['number'])
atributos_sin_anomalias =  list(happiness_sin_anomalias.columns.values )
fig = plt.figure(figsize=(10,10))
for i in range(len(atributos_sin_anomalias)):
    plt.subplot(5, 2, i+1)
    plt.hist(happiness_sin_anomalias[atributos_sin_anomalias[i]], bins=10,
edgecolor='black')
    plt.xlabel(atributos_sin_anomalias[i])
    plt.ylabel('count')
fig.tight_layout(pad = 1.2)
plt.show()
```

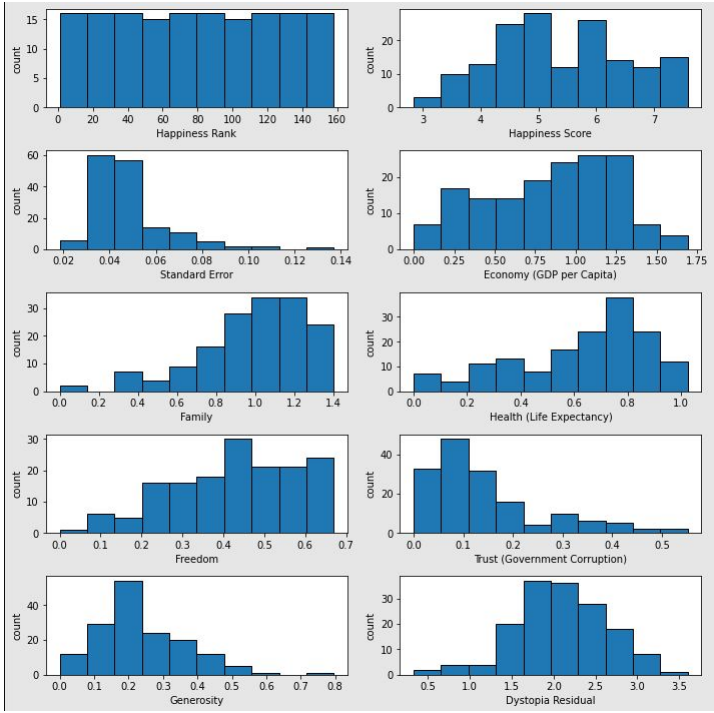
Dataset sobre felicidad (2015): Histogramas

Se puede observar una asimetría a la izquierda en los atributos:

Familia, libertad, esperanza de vida, economía y distopía.

Se puede observar una asimetría a la derecha en los atributos:

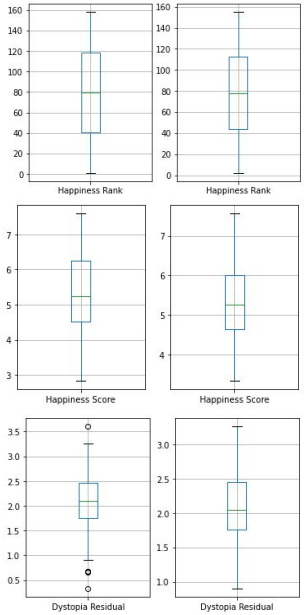
Error estándar, generosidad y confianza en el gobierno.

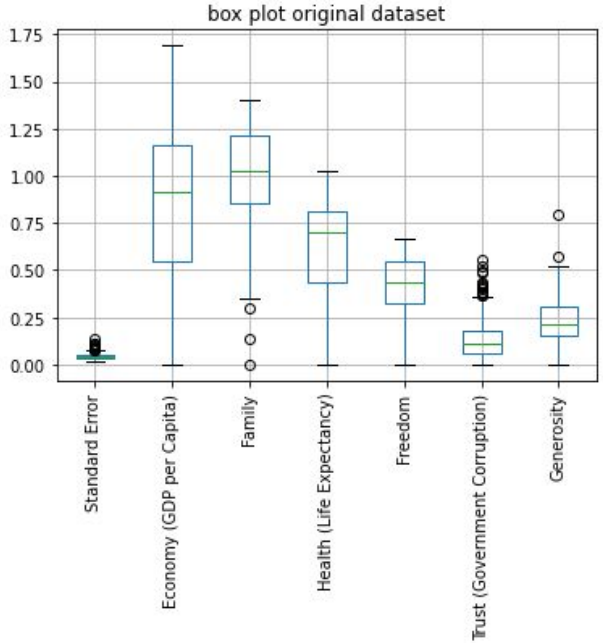


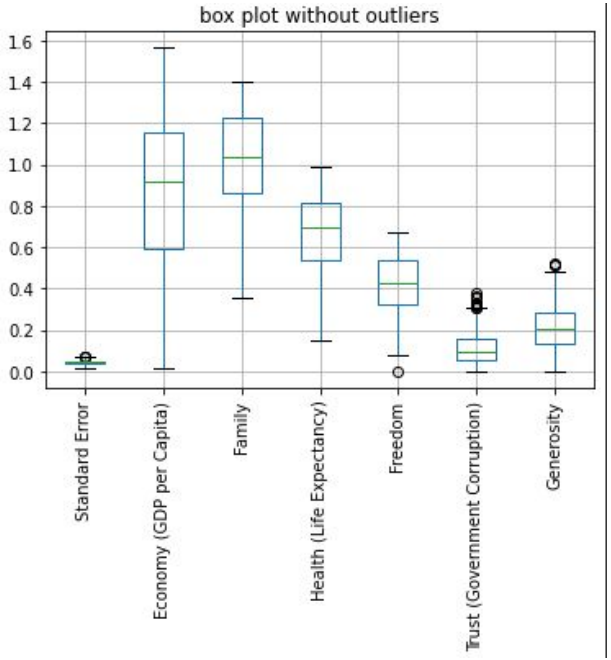
Dataset sobre felicidad (2015): Boxplot y outliers

```
hapiness = happiness.select_dtypes(['number'])
happiness_sin_anomalias = happiness.select_dtypes(['number'])
atributos_sin_anomalias = list(happiness_sin_anomalias.columns.values)
for i in atributos_sin_anomalias:
    q1 = happiness_sin_anomalias[i].quantile(0.25)
    q3  = happiness_sin_anomalias[i].quantile(0.75)
    qr = q3-q1
    q3 = q3+1.5*qr
    q1 = q1-1.5*qr
    happiness_sin_anomalias = happiness_sin_anomalias[(happiness_sin_anomalias[i] < q3) & (happiness_sin_anomalias[i] > q1)]
fig, axes = plt.subplots(nrows=1, ncols=2)
happiness.boxplot(column=['Happiness Rank'], ax = axes[0])
happiness_sin_anomalias.boxplot(column=['Happiness Rank'], ax = axes[1])
plt.show()
fig, axes = plt.subplots(nrows=1, ncols=2)
happiness.boxplot(column=['Happiness Score'], ax = axes[0])
happiness_sin_anomalias.boxplot(column=['Happiness Score'], ax = axes[1])
plt.show()
fig, axes = plt.subplots(nrows=1, ncols=2)
happiness.boxplot(column=['Dystopia Residual'], ax = axes[0])
happiness_sin_anomalias.boxplot(column=['Dystopia Residual'], ax = axes[1])
plt.show()
```

Dataset sobre felicidad (2015): Boxplot y outliers

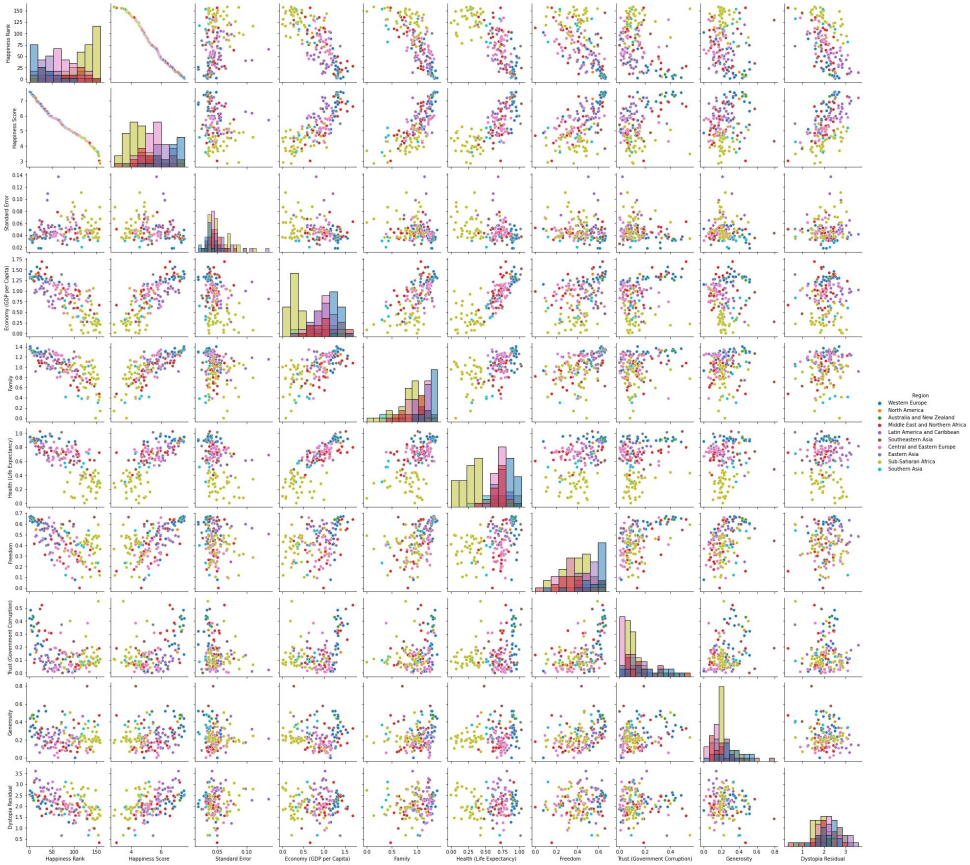






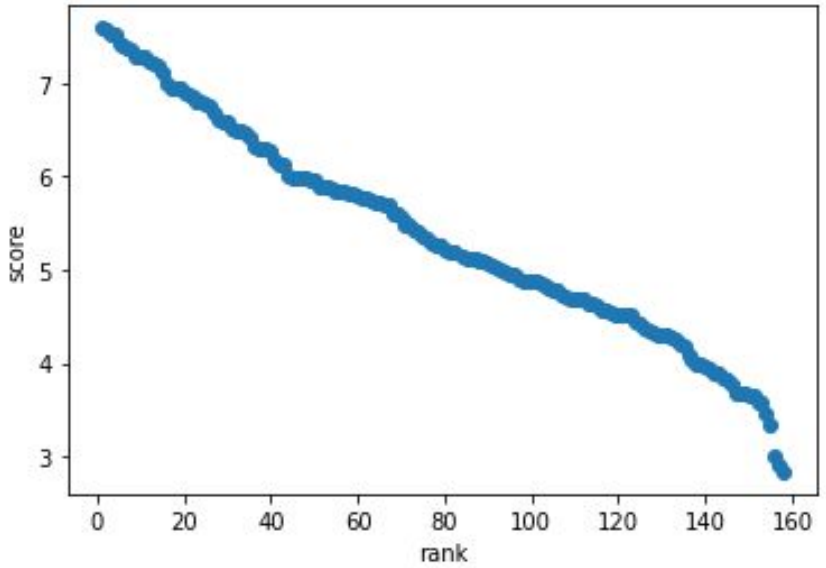
Dataset sobre felicidad (2015): Pairplot

```
#Pairplot respecto a regiones,
IMPORTANTENO CAMBIAR KIND Y
DIAG_KIND
sn.pairplot(
    happiness,
    hue="Region",
    kind = "scatter",
    diag_kind= "hist")
plt.show()
```



Dataset sobre felicidad (2015): Score vs Ranking

```
score = hapiness[['Happiness Score']]
rank = hapiness[['Happiness Rank']].sort_values(by=['Happiness Rank'])
plt.plot(rank,score, "o")
plt.xlabel('rank')
plt.ylabel('score')
plt.show()
```



Dataset sobre flores felicidad
¿Faltan datos? ¿Cosas interesantes?

La verdad no sabría qué agregarle al modelo ya que muchas cosas que se me ocurren que influyen en la felicidad son cualitativas, pero el modelo parece robusto y como esta tiene buenas puntuaciones. Se me hace interesante que la gente refleja con gran precisión la felicidad que su país/región puede darle con las condiciones existentes, habiendo una tendencia entre la calificación de los encuestados y el ranking del país.

Código

https://colab.research.google.com/drive/1rmSWgtGoe9Jab2fuT3Xavj07vpBPfr5b?usp=sharing