Photo by Edgar Chaparro on Unsplash

Estadística con python, proyecto real: empresa de zapatillas

Benjamin Flores Copa · Follow

8 min read · Sep 11, 2020

Pensemos que este es tu primer trabajo como data scientist en una empresa de zapatillas que ha operado durante 30 años, hay una persona que te da un tour por la empresa hasta que llegas al área de inventarios y percibes que las cosas en ese lugar están algo alborotadas.

Llegas a tu estación de trabajo, se te acerca el líder del equipo de data science y te asigna la primera tarea. Te entrega un dataset que contiene las ventas de los años 2014, 2015 y 2016, el problema que tiene la empresa es gestionando los inventarios, ya que, en la venta de zapatillas existe muchas tallas y no siempre se puede llevar la organización de estos y analizar la demanda de cada talla. Para que podamos corresponder la demanda de los

clientes es importante contar con la cantidad de zapatillas necesarias; pero eso puede incluir otro problema en el que se puede comprar zapatillas por demás y estas no serán vendidas. Las tendencias en el mercado de la zapatillas cambian constantemente, además de que estas zapatillas que no pueden ser vendidas ocupan un espacio en el inventario, entonces es importante poder predecir la cantidad máxima y mínima de zapatillas para el año entrante.

En ese momento te das cuenta de que yo soy tu compañero de trabajo, entonces me comentas sobre el problema y juntos nos aventuraremos a encontrar la solución.

Si prefieres el video de Youtube: https://youtu.be/7peQjqGSvAg

Comencemos con la importación de las librerías, para poder realizar la manipulación de nuestro dataset utilizaremos **pandas**, para visualizar gráficos utilizaremos **seaborn** conjuntamente con **matplotlib**.

```
import math
import pandas as pd
```

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

Primero analizaremos las variables del dataset, utilizando la librería pandas

```
df = pd.read_csv('shoes_dataset.csv')
df
```

	InvoiceNo	Date	Country	ProductID	Shop	Gender	Size (US)	Size (Europe)	Size (UK)	UnitPrice	Discount	SalePrice
0	52389	1/1/2014	United Kingdom	2152	UK2	Male	11.0	44	10.5	$159.00	0%	$159.00
1	52390	1/1/2014	United States	2230	US15	Male	11.5	44-45	11.0	$199.00	20%	$159.20
2	52391	1/1/2014	Canada	2160	CAN7	Male	9.5	42-43	9.0	$149.00	20%	$119.20
3	52392	1/1/2014	United States	2234	US6	Female	9.5	40	7.5	$159.00	0%	$159.00
4	52393	1/1/2014	United Kingdom	2222	UK4	Female	9.0	39-40	7.0	$159.00	0%	$159.00
...	...	...	...	...	...	...	...	...	...	...	...	...
14962	65773	12/31/2016	United Kingdom	2154	UK2	Male	9.5	42-43	9.0	$139.00	0%	$139.00
14963	65774	12/31/2016	United States	2181	US12	Female	12.0	42-43	10.0	$149.00	0%	$149.00
14964	65775	12/31/2016	Canada	2203	CAN6	Male	10.5	43-44	10.0	$179.00	30%	$125.30
14965	65776	12/31/2016	Germany	2231	GER1	Female	9.5	40	7.5	$199.00	0%	$199.00
14966	65777	12/31/2016	Germany	2156	GER1	Female	6.5	37	4.5	$139.00	10%	$125.10

14967 rows × 12 columns

Data Cleaning

En nuestro conjunto de datos tenemos la columna Date, esta columna representa la fecha en la cuál se registro la venta. Si queremos agrupar por año, por mes, deberíamos de separar en otras columnas que, es lo que haremos.

```
df['Date'] = pd.to_datetime(df['Date'])
df['Year'] = df['Date'].dt.year
df['Day'] = df['Date'].dt.day
df['Month'] = df['Date'].dt.month
```

Tener el precio con el signo dolar no nos sirve ya que deberíamos de tenerlo en un punto flotante el cuál nos permitirá realizar cálculos. Seleccionando el valor después del signo dolár, nos permite obtener la parte numérica.

```
df['SalePrice'] = df['SalePrice'].apply(lambda x: float(x[2:]))
df['UnitPrice'] = df['UnitPrice'].apply(lambda x: float(x[2:]))
```

Data Analysis

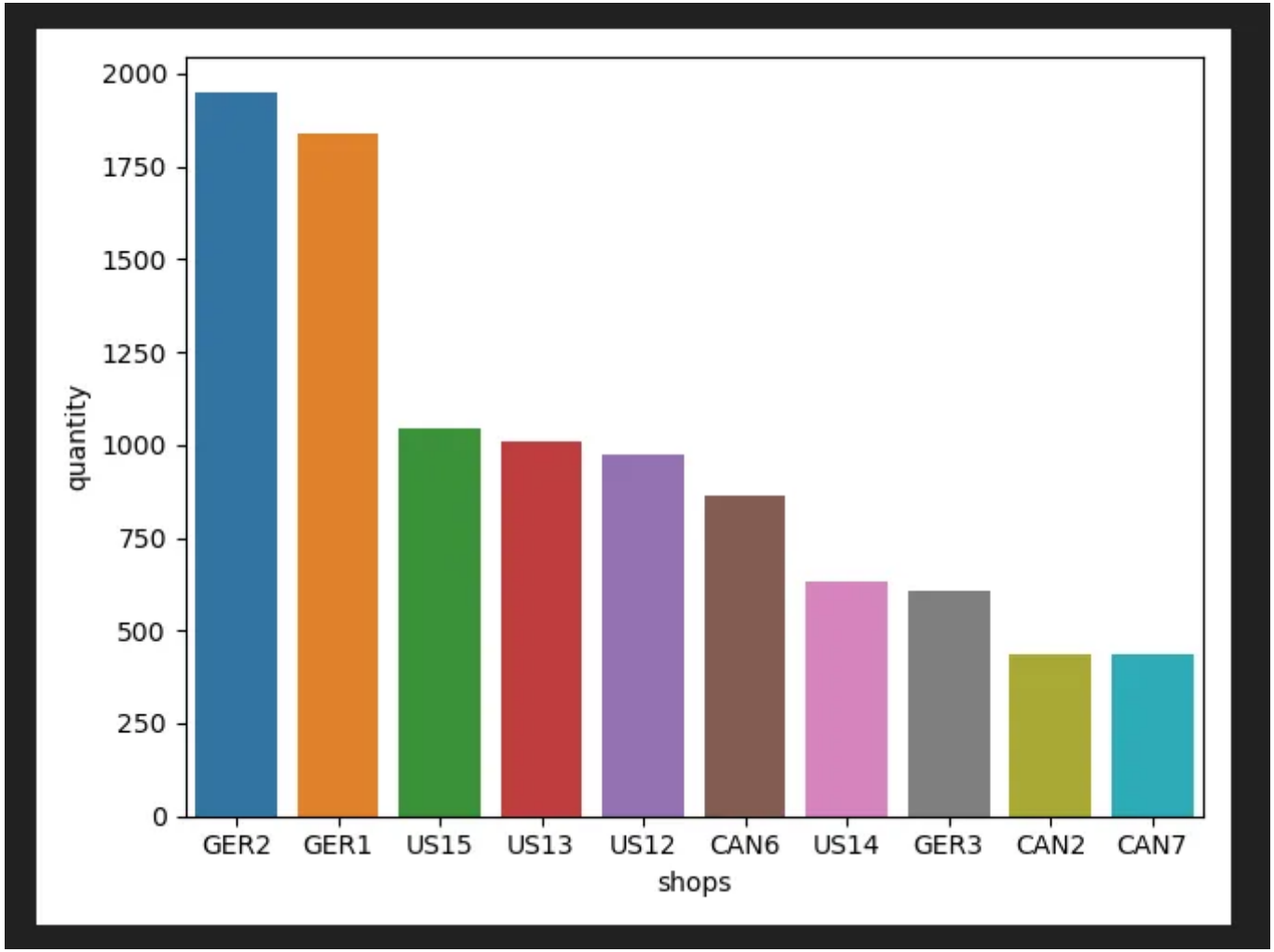
Bien, ahora realizaremos análisis de datos en nuestro dataset, seleccionaremos las variables categóricas y numéricas.

```
categorical_variables = ['Country', 'ProductID', 'Shop', 'Gender',
'Size (US)', 'Discount', 'Year', 'Month']

numerical_variables = ['UnitPrice', 'SalePrice']
```

En nuestra empresa de zapatillas tenemos diferentes sedes los cuáles están representadas por el código de la tienda ejm: UK2, US15, etc. Veamos las 10 primeras tiendas que más venden zapatillas indistintamente del género, país u otra variable.

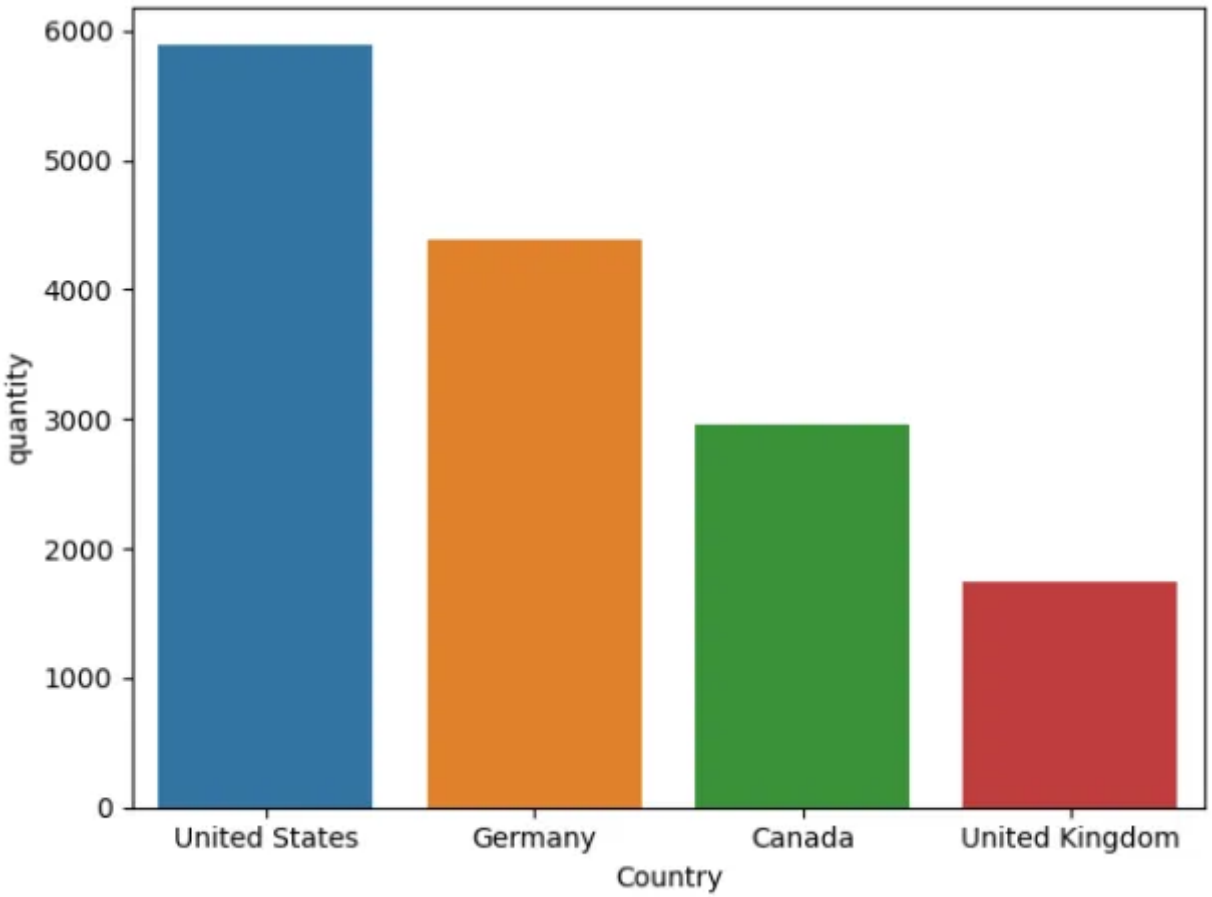
En el gráfico podemos observas que las tiendas GER2 y GER1 son las tiendas que más ventas realizan, en base al código de la tienda podemos inferir que son alemanas.



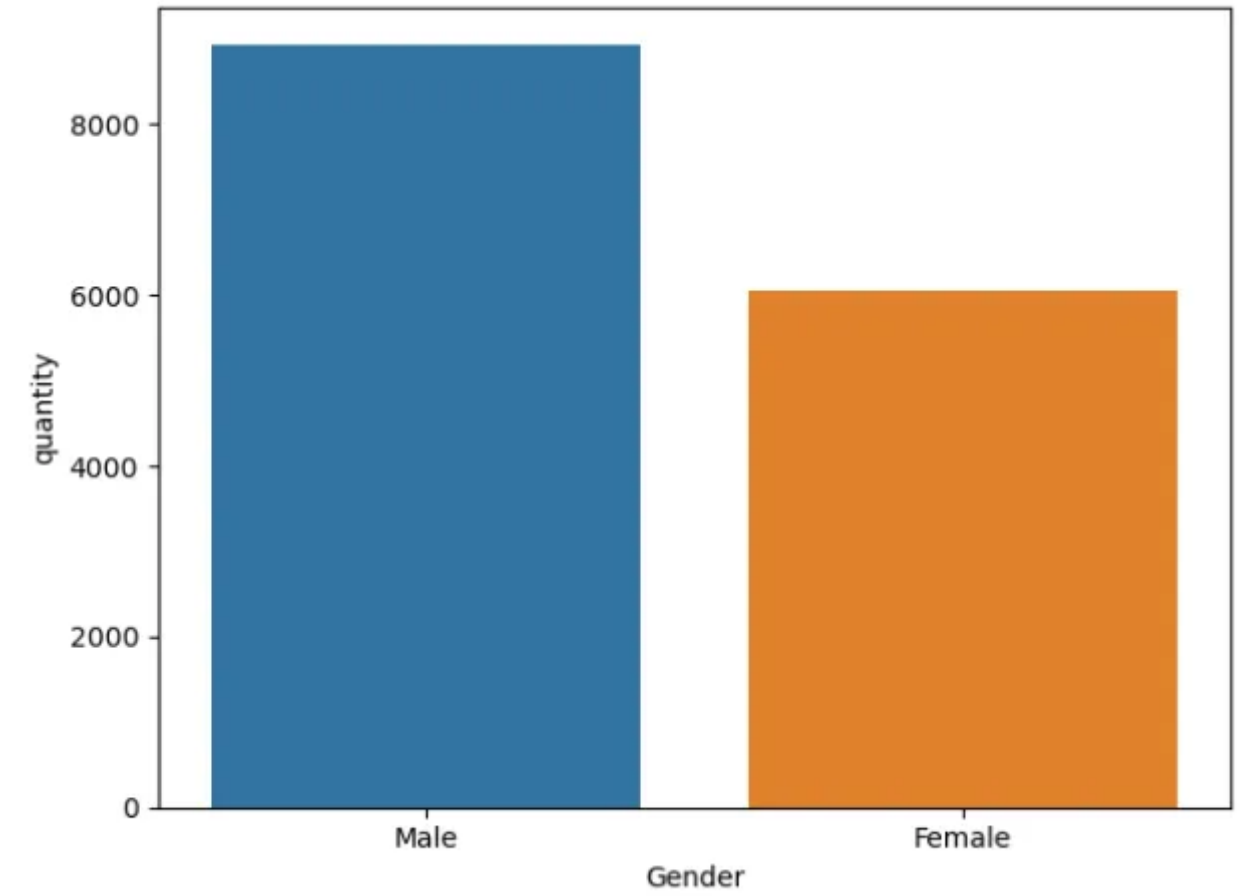
Luego realizamos un recorrido por todas nuestras variables numéricas para poder encontrar percepciones acerca de nuestro dataset. Realizaremos un recorrido por nuestras variables categóricas, realizaremos el conteo de las frecuencias utilizando la función value_counts(), crearemos un dataframe para poder utilizar la función barplot() de seaborn.

```
for cat_variable in categorical_variables:
    frequency = df[cat_variable].value_counts()
    df_frequency = pd.DataFrame({'index': frequency.index.tolist(),
'values': frequency.tolist()})
    sns.barplot(x='index', y='values', data=df_frequency)
    plt.show()
```

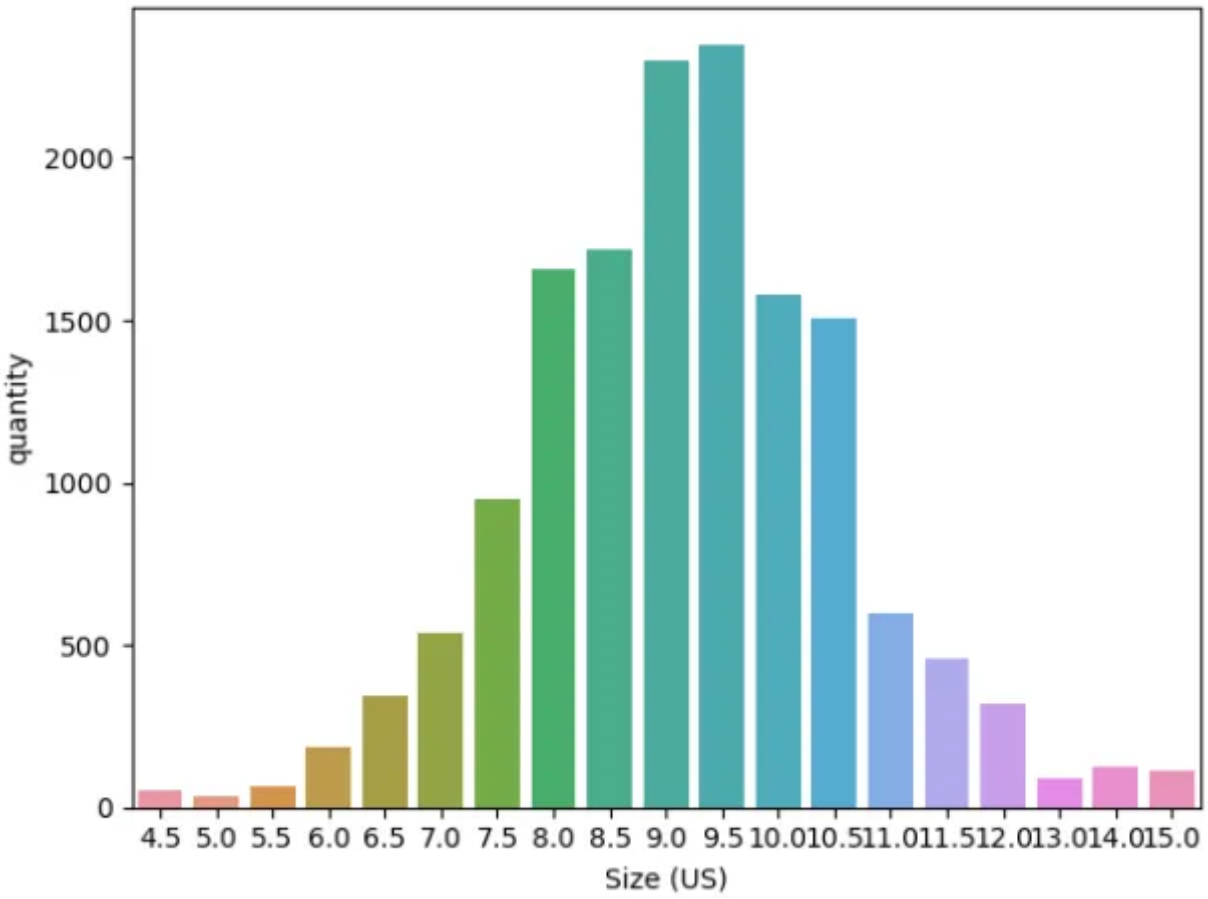
Iremos viendo las gráficas que son útiles, en el gráfico de abajo tenemos las ventas por país, viendo así que, el país que más ventas registró durante estos años es Estados Unidos.



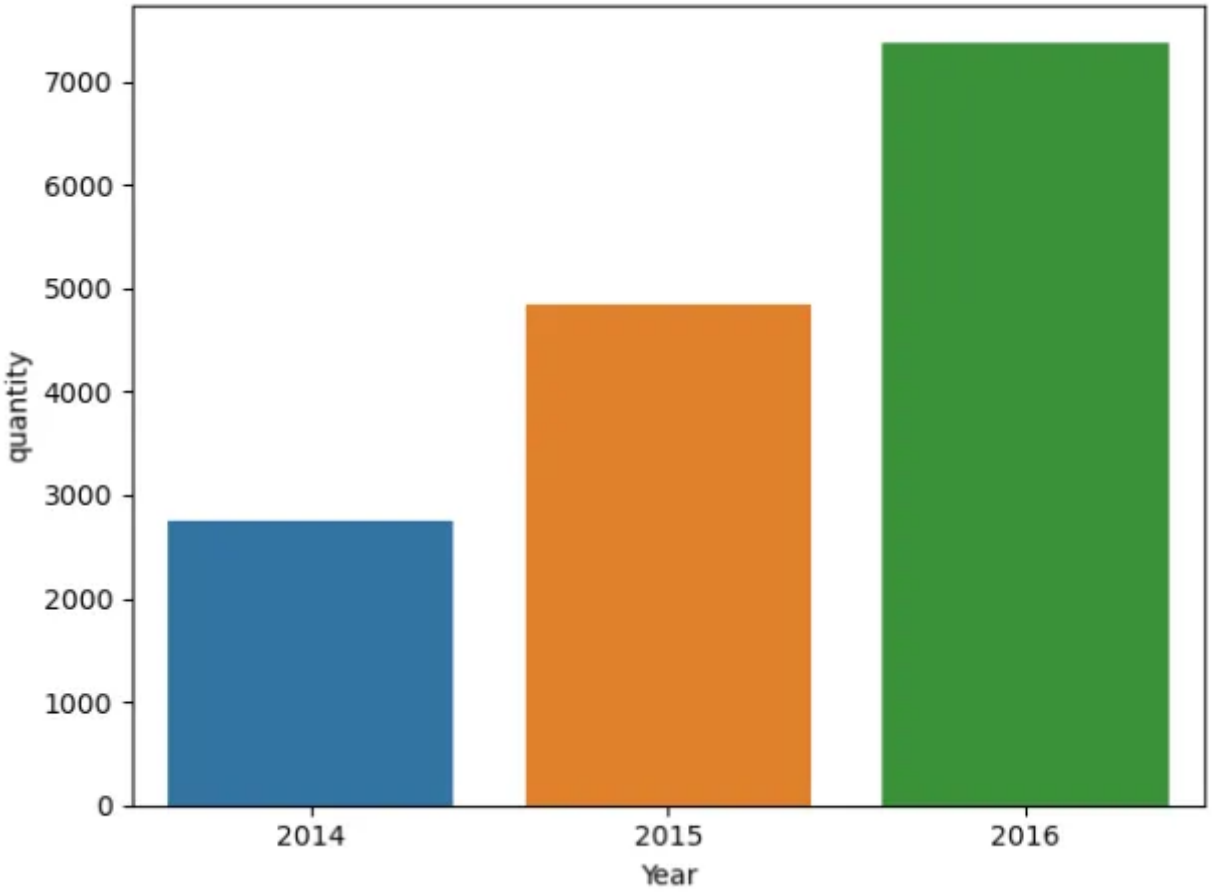
En la gráfica abajo veamos que los hombres son los que más compraron respecto a las mujeres, con una diferencia de más de 2000 unidades.



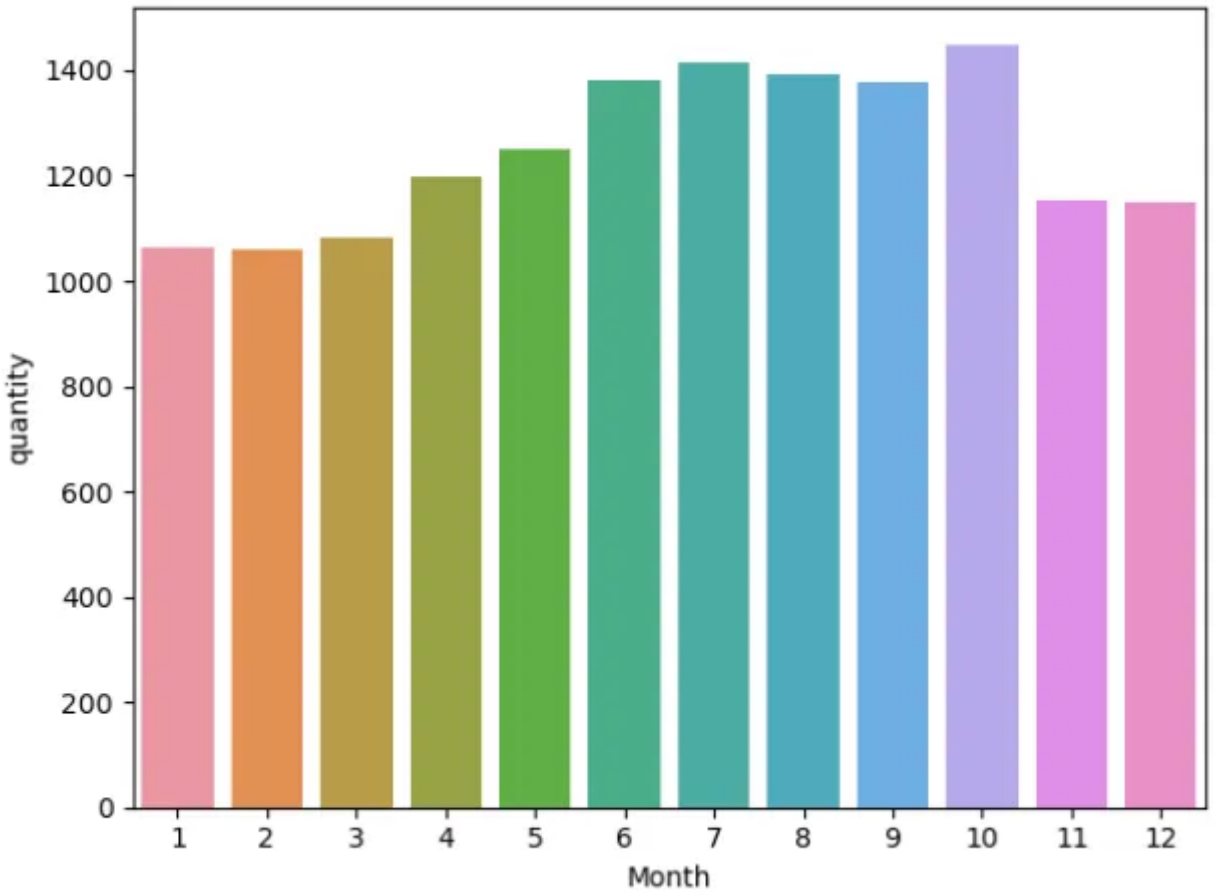
En la gráfica de abajo vemos la venta de zapatillas por talla indistintamente de género, país, otra variable. Notamos que es una distribución normal y las tallas más vendidas son la 9.0 y 9.5



Veamos las ventas en los distintos años de nuestro dataset para así poder determinar si las ventas crecieron o no. Podemos ver que notablemente las ventas crecieron año a año.



Ahora es importante también analizar las ventas mes a mes para poder ver si en algún mes en específico las ventas fueron mayores a las demás. Podemos ver que no hay un incremento significativo pero podríamos resaltar el mes 10 que es Octubre.



Analizaremos nuestras variables numéricas, que son UnitPrice y SalePrice. UnitPrice es el precio de la zapatilla mientras que SalePrice es el precio de venta de la zapatilla después de aplicarle o no un descuento.

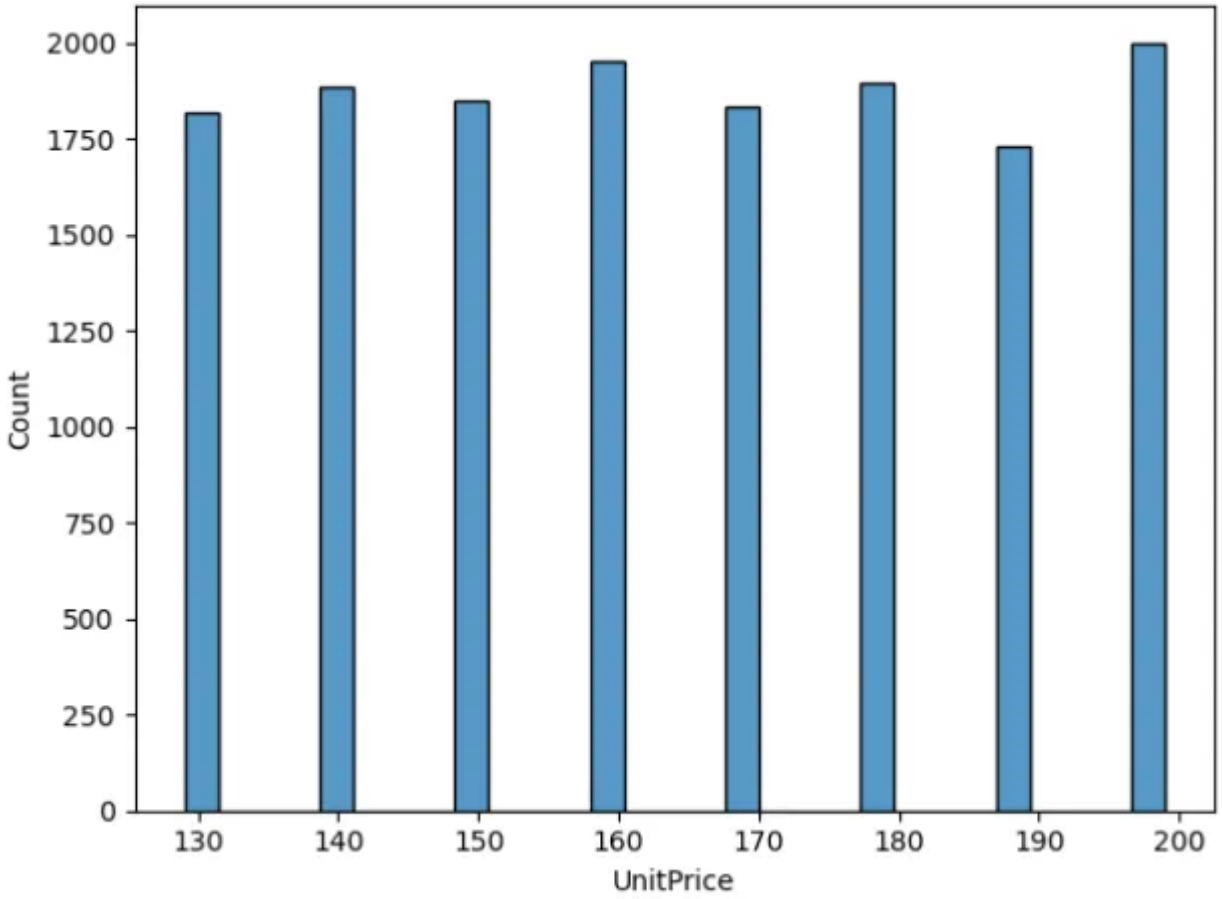
Realizaremos un recorrido de nuestras variables numéricas, con la función histplot() de seaborn realizaremos un histograma que nos sirve para dibujar

nuestras variables numéricas representadas en intervalos.

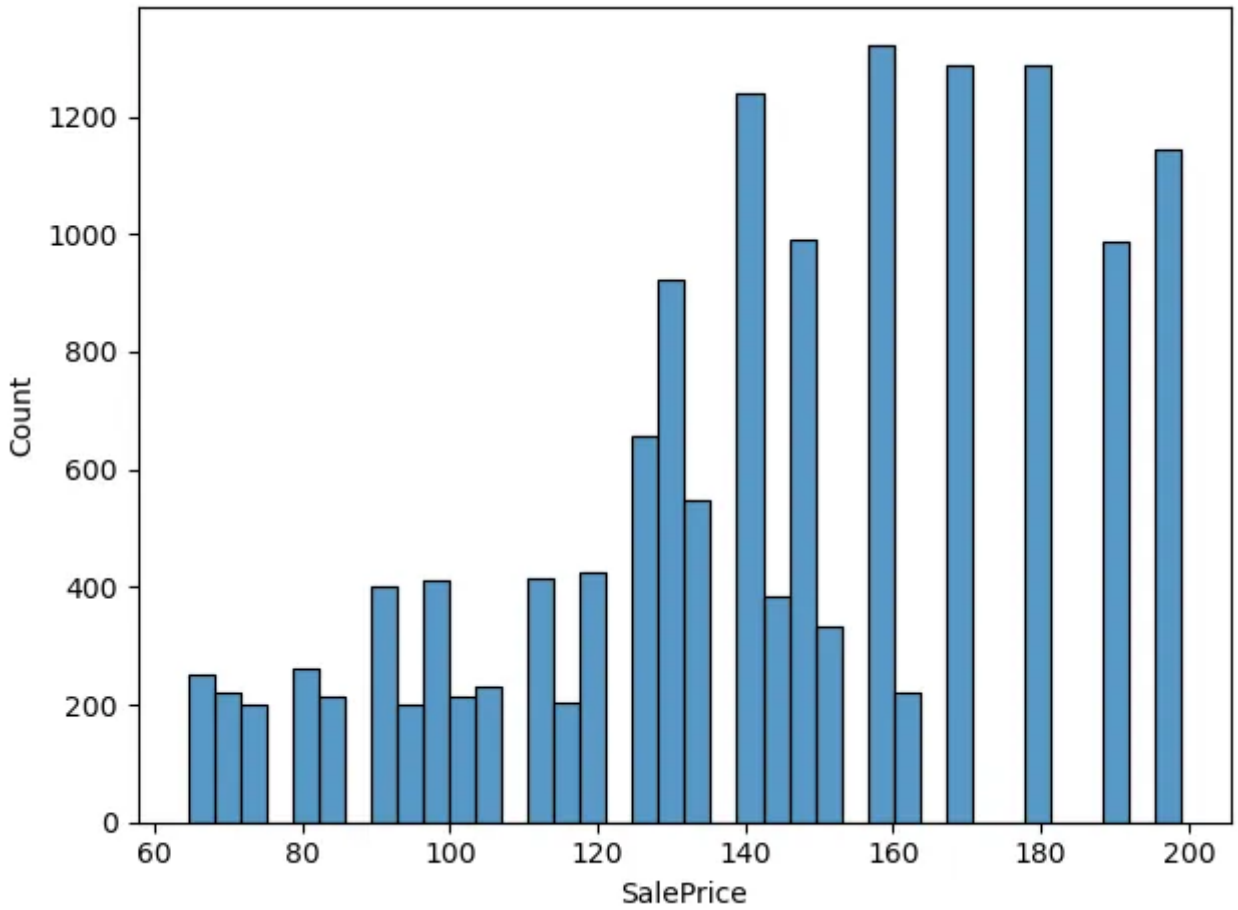
Nota: La función histplot() de seaborn esta disponible desde la versión 0.11

```
for num_variable in numerical_variables:
    sns.histplot(df[num_variable], bins='auto')
    plt.show()
```

Al ver el histograma de nuestra variable UnitPrice podemos notar que es una distribución uniforme. Existen tantas zapatillas cerca de un mismo precio.



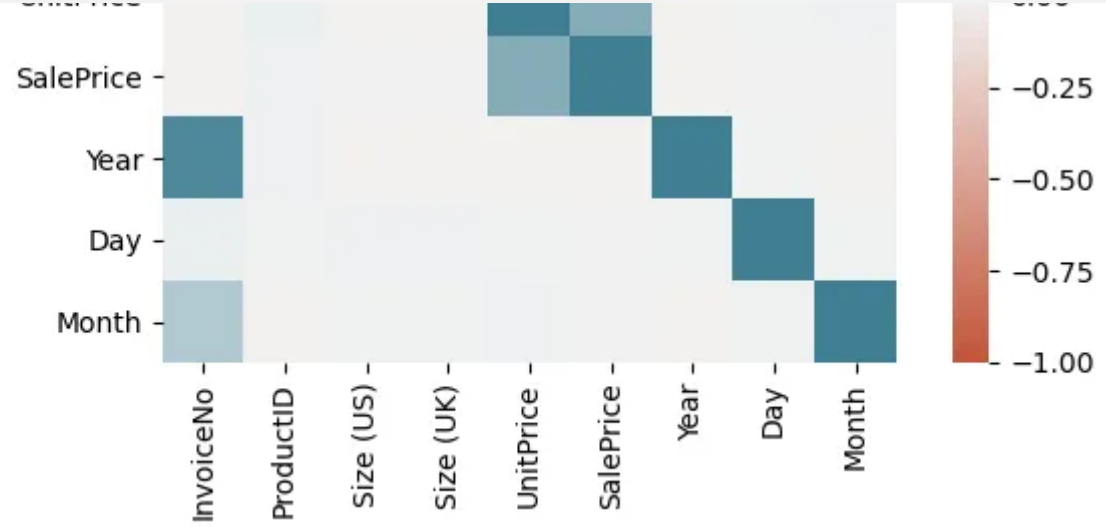
Observando nuestra variables SalePrice podemos notar claramente los descuentos aplicados.



Ahora encontremos si existe alguna correlación entre nuestras variables.

```
corr = df.corr()
sns.heatmap(corr, vmin=-1, vmax=1,
center=0,cmap=sns.diverging_palette(20, 220, n=200),square=True)
plt.show()
```

Podemos ver que no existe una correlación que nos importe, la correlación que se encontró es la de InvoiceNo y Year pero estas variables ambas son categóricas, si tendríamos que darle una explicación sería de que a mayor cantidad de zapatillas el InvoiceNo incrementa junto con el año.



Solución Propuesta

Analizados nuestros datos hemos percibido que Estados Unidos es el principal mercado, los hombres son un mercado mayor que el de las mujeres y que los dos últimos años conforman la mayoría de las ventas.

Para poder resolver nuestro problema, que es, poder calcular la cantidad máxima de zapatillas que se venderán en base a nuestras ventas que se han realizado los años anteriores.

Utilizaremos los datos del año 2015 y 2016, las ventas realizadas a los hombres y al país de Estados Unidos. Agruparemos basándonos en la talla que es Size (US), en base al año y al mes. Con la función size() realizaremos el conteo de nuestros datos para obtener la cantidad de tallas vendidas, con la función unstack(level=0) podremos utilizar Size (US) como nuestras columnas y con fillna(value=0) el cual, llenara los valores Nan con el valor 0

```
grouped = df[(df['Year'] != 2014) & (df['Gender'] == 'Male')
            & (df['Country'] == 'United States')].groupby(
            ['Size (US)', 'Year', 'Month']).size()\
            .unstack(level=0).fillna(value=0)
```

Tendremos los datos de esta forma. En las columnas estará la talla de las zapatillas y en las filas como índices tenemos el año y mes. En los datos están las cantidades.

	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.0	10.5	11.0	11.5	
2015/1	0.00000	4.00000	0.00000	0.00000	5.00000	4.00000	10.00000	18.00000	8.00000	7.00000	5.00000	3.00000	1
2015/2	0.00000	1.00000	0.00000	1.00000	3.00000	5.00000	14.00000	16.00000	13.00000	13.00000	5.00000	3.00000	0
2015/3	0.00000	0.00000	0.00000	2.00000	1.00000	3.00000	7.00000	23.00000	13.00000	8.00000	4.00000	2.00000	0
2015/4	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000	9.00000	15.00000	15.00000	8.00000	7.00000	3.00000	3
2015/5	3.00000	0.00000	1.00000	0.00000	6.00000	2.00000	17.00000	16.00000	7.00000	15.00000	5.00000	6.00000	1
2015/6	1.00000	0.00000	2.00000	0.00000	6.00000	11.00000	16.00000	16.00000	20.00000	10.00000	5.00000	9.00000	4
2015/7	1.00000	2.00000	1.00000	3.00000	4.00000	6.00000	20.00000	19.00000	22.00000	16.00000	4.00000	6.00000	1
2015/8	3.00000	3.00000	0.00000	2.00000	0.00000	6.00000	21.00000	26.00000	18.00000	8.00000	6.00000	6.00000	2
2015/9	5.00000	0.00000	1.00000	3.00000	5.00000	4.00000	13.00000	25.00000	22.00000	16.00000	3.00000	3.00000	4
2015/10	4.00000	1.00000	2.00000	2.00000	6.00000	12.00000	17.00000	17.00000	15.00000	16.00000	5.00000	8.00000	3
2015/11	0.00000	3.00000	2.00000	2.00000	3.00000	4.00000	10.00000	35.00000	11.00000	17.00000	5.00000	3.00000	6
2015/12	0.00000	3.00000	3.00000	2.00000	3.00000	2.00000	12.00000	24.00000	14.00000	14.00000	10.00000	7.00000	3
2016/1	4.00000	3.00000	0.00000	3.00000	7.00000	12.00000	17.00000	19.00000	17.00000	13.00000	5.00000	4.00000	3
2016/2	1.00000	2.00000	0.00000	2.00000	9.00000	12.00000	13.00000	25.00000	26.00000	16.00000	16.00000	3.00000	0
2016/3	3.00000	0.00000	1.00000	3.00000	7.00000	8.00000	13.00000	27.00000	26.00000	22.00000	13.00000	6.00000	0

Bien una vez que tenemos estos datos, lo que podríamos hacer es calcular el promedio por columna y de esa forma sabríamos cuánto vamos a vender de zapatillas basándonos nada más que en el promedio o media. Esto no nos

lleva a resolver nuestro problema ya que la media tiende a verse afectada por los valores atípicos además de que no nos da el valor máximo que venderemos basados en nuestros datos.

Primero observemos que los datos que tenemos es una muestra porque estamos analizando datos de 3 años con respecto a los 30 años que viene operando la tienda de zapatillas.

Para poder resolver este problema haremos uso de intervalos de confianza, usar estos intervalos nos permitirá calcular los valores mínimos y máximos basados en nuestros datos. Utilizar un nivel de confianza del 95% es recomendado, empecemos.

Con la siguiente fórmula podemos calcular el intervalo de confianza.

$$\text{C.I.}_{\text{mean}}: \quad \bar{x} \pm \left(t_{\frac{\alpha}{2},n-1} \times \frac{s}{\sqrt{n}}\right) \tag{1}$$

where

$\bar{x}$: sample mean

α : significance level

n : number of samples

s : sample standard deviation

t : t-score. depends on α and degrees of freedom $n-1$

Primero haremos el cálculo de la media o promedio y el error estándar, el error estándar es igual a la desviación estándar de la muestra dividida por la raíz cuadrada de la cantidad de nuestros datos.

Haremos el cálculo por talla de zapatilla que tenemos.

```
means = []
standard_errors = []
for column in grouped.columns:
    means.append(grouped[column].mean())
    standard_errors.append(grouped[column].sem())

d = {'means': means, 'std_error': standard_errors}
df_calculations = pd.DataFrame(data=d, index=grouped.columns)
```

Tenemos en la primera columna la media por talla de zapatilla y el error estándar.

	means	std_error
6.0	2.16667	0.39318
6.5	1.58333	0.34006
7.0	1.33333	0.33872
7.5	2.33333	0.41120
8.0	4.79167	0.59885
8.5	7.87500	0.94469
9.0	16.33333	1.26214
9.5	25.58333	1.76614
10.0	18.79167	1.32558
10.5	14.95833	1.02058
11.0	7.54167	0.71975
11.5	5.33333	0.58256
12.0	3.08333	0.58022
13.0	1.20833	0.25523
14.0	1.95833	0.29781
15.0	0.54167	0.19015

Ahora realizaremos el cálculo del margen de error que es multiplicar t-score por el error estándar. Para calcular t-score de alpha medios, utilizamos un nivel de confianza de 95% y la cantidad de datos de n-1, n vendría a ser 24 debido a que estamos calculando por dos años. El valor resaltado podríamos tomarlo como 2.07

d.f. / α	0.1	0.05	0.025	0.01	0.005
1	3.078	6.314	12.706	31.821	63.657
2	1.886	2.920	4.303	6.965	9.925
3	1.638	2.353	3.182	4.541	5.841
4	1.533	2.132	2.776	3.747	4.604
5	1.476	2.015	2.571	3.365	4.032
6	1.440	1.943	2.447	3.143	3.707
7	1.415	1.895	2.365	2.998	3.499
8	1.397	1.860	2.306	2.896	3.355
9	1.383	1.833	2.262	2.821	3.250
10	1.372	1.812	2.228	2.764	3.169
11	1.363	1.796	2.201	2.718	3.106
12	1.356	1.782	2.179	2.681	3.055
13	1.350	1.771	2.160	2.650	3.012
14	1.345	1.761	2.145	2.624	2.977
15	1.341	1.753	2.131	2.602	2.947
16	1.337	1.746	2.120	2.583	2.921
17	1.333	1.740	2.110	2.567	2.898
18	1.330	1.734	2.101	2.552	2.878
19	1.328	1.729	2.093	2.539	2.861
20	1.325	1.725	2.086	2.528	2.845
21	1.323	1.721	2.080	2.518	2.831
22	1.321	1.717	2.074	2.508	2.819
23	1.319	1.714	2.069	2.500	2.807
24	1.318	1.711	2.064	2.492	2.797
25	1.316	1.708	2.060	2.485	2.787
26	1.315	1.706	2.056	2.479	2.779
27	1.314	1.703	2.052	2.473	2.771
28	1.313	1.701	2.048	2.467	2.763
29	1.311	1.699	2.045	2.462	2.756
30	1.310	1.697	2.042	2.457	2.750
35	1.306	1.690	2.030	2.438	2.724
40	1.303	1.684	2.021	2.423	2.704
50	1.299	1.676	2.009	2.403	2.678
60	1.296	1.671	2.000	2.390	2.660
120	1.289	1.658	1.980	2.358	2.617
inf.	1.282	1.645	1.960	2.326	2.576
CI*	80%	90%	95%	98%	99%

Calculemos el margen de error.

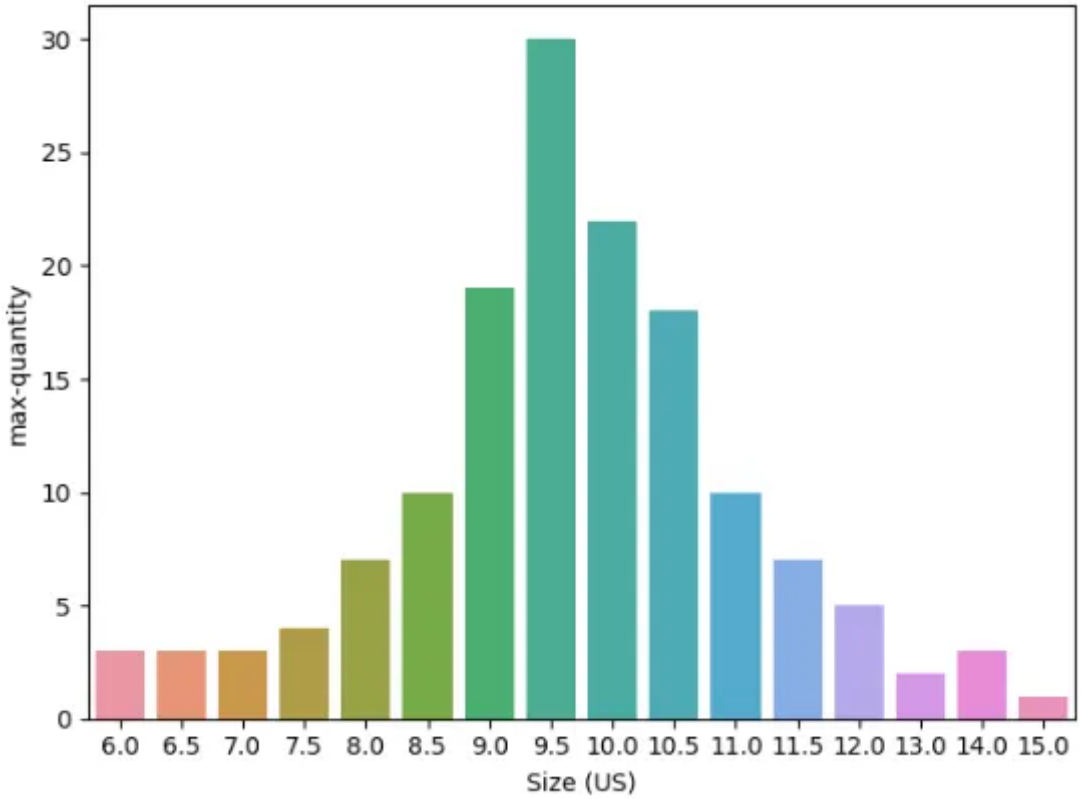
```
df_calculations['error_margin'] =
df_calculations['std_error'].apply(lambda x: x * 2.07)
```

	means	std_error	error_margin
6.0	2.16667	0.39318	0.81388
6.5	1.58333	0.34006	0.70392
7.0	1.33333	0.33872	0.70116
7.5	2.33333	0.41120	0.85118
8.0	4.79167	0.59885	1.23962
8.5	7.87500	0.94469	1.95551
9.0	16.33333	1.26214	2.61263
9.5	25.58333	1.76614	3.65592
10.0	18.79167	1.32558	2.74396
10.5	14.95833	1.02058	2.11261
11.0	7.54167	0.71975	1.48988
11.5	5.33333	0.58256	1.20589
12.0	3.08333	0.58022	1.20105
13.0	1.20833	0.25523	0.52832
14.0	1.95833	0.29781	0.61647
15.0	0.54167	0.19015	0.39361

Bien ahora sumemos y restemos a la media para poder calcular nuestro intervalo de confianza y además haremos un redondeo al alza para determinar la cantidad máxima de zapatillas con un nivel de confianza al 95% basados en los años 2015 y 2016. La columna math_round_up representa este valor.

	means	std_error	error_margin	low_margin	up_margin	math_round_up
6.0	2.16667	0.39318	0.81388	1.35279	2.98055	3
6.5	1.58333	0.34006	0.70392	0.87941	2.28726	3
7.0	1.33333	0.33872	0.70116	0.63217	2.03449	3
7.5	2.33333	0.41120	0.85118	1.48216	3.18451	4
8.0	4.79167	0.59885	1.23962	3.55205	6.03128	7
8.5	7.87500	0.94469	1.95551	5.91949	9.83051	10
9.0	16.33333	1.26214	2.61263	13.72071	18.94596	19
9.5	25.58333	1.76614	3.65592	21.92742	29.23925	30
10.0	18.79167	1.32558	2.74396	16.04771	21.53562	22
10.5	14.95833	1.02058	2.11261	12.84572	17.07094	18
11.0	7.54167	0.71975	1.48988	6.05178	9.03155	10
11.5	5.33333	0.58256	1.20589	4.12744	6.53923	7
12.0	3.08333	0.58022	1.20105	1.88228	4.28439	5
13.0	1.20833	0.25523	0.52832	0.68001	1.73666	2
14.0	1.95833	0.29781	0.61647	1.34186	2.57481	3
15.0	0.54167	0.19015	0.39361	0.14806	0.93527	1

Ahora grafiquemos para hacernos una idea de las cantidades.



Como vemos tiene la forma de una distribución normal, las cantidades máximas representan poder manejar el inventario de nuestra tienda de zapatillas, hemos realizado el cálculo para las ventas a los hombres que están en Estados Unidos. Podemos realizar cálculos para mujeres también y así de esa forma hacer un resumen basado incluso por tienda, todo esto

sigue la misma idea y el mismo procedimiento para calcular intervalos de confianza.

. . .

Después de lograr realizar todo este análisis de datos y poder predecir la cantidad máxima de zapatillas que se necesitarán mensualmente para hombres residentes en Estados Unidos. Realizas un resumen de esto que conseguimos juntos y lo presentas al líder del equipo de data science. En base al resumen que presentaste la empresa empieza a gestionar de manera efectiva sus inventarios y pueden pensar en resolver otro tipo de problemas, que de nuevo tu puedes ser la persona que puede resolverlos! .

Sígueme en Twitter donde estaré compartiendo historias del trayecto que voy siguiendo para convertirme en data scientist!

Código en Github

Python Data Science Statistics Spanish