

CURSO DE ESTADÍSTICA - PARTE

1

Trabajo de Análisis Descriptivo de un Conjunto de Datos

Utilizando los conocimientos adquiridos en nuestro entrenamiento realice un análisis descriptivo básico del conjunto de datos central que utilizamos durante el curso.

Vamos a construir histogramas, calcular y evaluar medidas de tendencia central, medidas de localización y de dispersión de los datos.

Siga el rutero propuesto y ve completando las células de código vacías. Intenta pensar en Más informaciones interesantes que pueden ser exploradas en nuestro dataset.

▼ 1.1 Dataset del proyecto

Muestra de domicilios Colombia - 2018

Las investigaciones por muestras de domicilios realizadas anualmente, busca encontrar características generales de la población, de educación, trabajo, rendimiento y otras, de acuerdo con las necesidades de información del país, tales como las características de migración, fertilidad, casamientos, salud, nutrición, entre otros temas. Estas muestras al pasar de los años consistuyen una herramienta importante para la formulación, validación y evaluación de políticas dirigidas al desarrollo socioeconómico y la mejora de las condiciones de vida en Colombia.

Datos

Los datos fueron creados de manera didáctica para este curso.

▼ Variables utilizadas

Ingreso

Ingresos mensuales (en miles de pesos) del trabajo principal para personas de 10 años o más.

Edad

Edad del entrevistado en la fecha de referencia en años.

Altura

Altura del entrevistado en metros.

Ciudad

Código de referência a 27 ciudades analizadas.

Sexo

Código	Descripción
0	Masculino
1	Femenino

Años de Estudio

Código	Descripción
1	Sin estudios y menos de 1 año
2	1 año
3	2 años
4	3 años
5	4 años
6	5 años
7	6 años
8	7 años
9	8 años
10	9 años
11	10 años
12	11 años
13	12 años
14	13 años
15	14 años
16	15 años o más
17	No se sabe

Código	Descripción
	No aplica

Color

Código	Descripción
0	Indio
2	Blanco
4	Negro
6	Amarillo
8	Moreno
9	Sin declarar

▼ Tratamiento a los datos

Algunos de los tratamientos de datos más frecuentes son:

1. Eliminar las observaciones (líneas) con entradas de datos inválidos;
2. Eliminar observaciones donde hay datos perdidos (missing data);
3. Filtros propios de la investigación, por ejemplo: considerar solo las encuestas realizadas a la cabeza de familia (responsable por el domicilio).

▼ Utilice la célula abajo para importar las bibliotecas que necesite para ejecutar las tareas

Sugerencias: pandas, numpy, seaborn

```
import pandas as pd
import numpy as np
import seaborn as sns
```

▼ Importe el dataset y almacene el contenido en un DataFrame

```
datos = pd.read_csv('datos.csv')
```

▼ Visualice el contenido del DataFrame

```
datos.head()
```

	Ciudad	Sexo	Edad	Color	Años de Estudio	Ingreso	Altura
0	11	0	23	8	12	800	1.603808
1	11	1	23	2	12	1150	1.739790
2	11	1	35	8	15	880	1.760444
3	11	0	46	2	6	3500	1.783158
4	11	1	47	8	9	150	1.690631



Para evaluar el comportamiento de la variable INGRESO vamos a construir una tabla de frecuencias considerando las siguientes clases según el salario mínimo (SM)

Describa los puntos más relevantes que usted observe en la tabla y en el gráfico.

Clases de ingreso:

A ► Más de 25 SM

B ► De 15 a 25 SM

C ► De 5 a 15 SM

D ► De 2 a 5 SM

E ► Hasta 2 SM

Para construir las clases de ingreso considere que el salario mínimo era de **\$ 788,00** miles de pesos colombianos.

Siga los pasos abajo:

▼ 1º Definir los intervalos de las clases

```
clases = [
    datos.Ingreso.min(),
    2 * 788,
    5 * 788,
    15 * 788,
    25 * 788,
    datos.Ingreso.max()
```

```
]
clases

[0, 1576, 3940, 11820, 19700, 200000]
```

▼ 2º Definir los labels de las clases

```
labels = ['E', 'D', 'C', 'B', 'A']
```

▼ 3º Construir la columna de frecuencias

```
frecuencia = pd.value_counts(
    pd.cut(x = datos.Ingreso,
           bins = clases,
           labels = labels,
           include_lowest = True)
)
frecuencia
```

E	49755
D	18602
C	7241
B	822
A	420

Name: Ingreso, dtype: int64

▼ 4º Construir la columna de porcentajes


```
porcentaje = pd.value_counts(
    pd.cut(x = datos.Ingreso,
           bins = clases,
           labels = labels,
           include_lowest = True),
    normalize = True
) * 100
porcentaje
```

E	64.751432
D	24.208745
C	9.423477
B	1.069755
A	0.546590

Name: Ingreso, dtype: float64

5º Juntar las columnas de frecuencias y porcentajes y ordenar las líneas según los labels de las clases

```
dist_frec_ingreso = pd.DataFrame(  
    {'Frecuencia': frecuencia, 'Porcentaje (%)': porcentaje}  
)  
dist_frec_ingreso.sort_index(ascending = False)
```

	Frecuencia	Porcentaje (%)	
A	420	0.546590	
B	822	1.069755	
C	7241	9.423477	
D	18602	24.208745	
E	49755	64.751432	

Construya um gráfico de barras para visualizar las informaciones de la tabla de frecuencias de arriba

```
dist_frec_ingreso['Frecuencia'].plot.bar(width = 1, color = 'blue', alpha = 0.2, figsize=(14,
```

<Axes: >



Conclusiones

El orden de frecuencia es Descendente empezando por E, hay mas porcentaje de personas que ganan hasta el doble del salario mínimo que los que ganan más de 25 el salario mínimo

- ▼ Cree un histograma para las variables QUANTITATIVAS de nuestro dataset

Describa los puntos más relevantes que usted observa en los gráficos (assimetrías y sus tipos, posibles causas para determinados comportamientos etc.)

```
ax = sns.distplot(datos['Edad'])
ax.figure.set_size_inches(14, 6)
ax.set_title('Distribución de Frecuencias - EDAD', fontsize=18)
ax.set_xlabel('Años', fontsize=14)
ax
```

```
<ipython-input-10-85d802e9093e>:1: UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see

<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
ax = sns.distplot(datos['Edad'])  
<Axes: title={'center': 'Distribución de Frecuencias - EDAD'}, xlabel='Años',  
ylabel='Density'>
```



```
ax = sns.distplot(datos['Altura'])  
ax.figure.set_size_inches(14, 6)  
ax.set_title('Distribución de Frecuencias - ALTURA', fontsize=18)  
ax.set_xlabel('Metros', fontsize=14)  
ax
```



```
<ipython-input-11-f61b1386f811>:1: UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see

<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
ax = sns.distplot(datos['Altura'])  
<Axes: title={'center': 'Distribución de Frecuencias - ALTURA'}, xlabel='Metros',  
ylabel='Density'>
```

Distribución de Frecuencias - ALTURA

```
ax = sns.distplot(datos['Ingreso'])  
ax.figure.set_size_inches(14, 6)  
ax.set_title('Distribución de Frecuencias - INGRESO', fontsize=18)  
ax.set_xlabel('Miles de pesos colombianos', fontsize=14)  
ax
```

```
<ipython-input-12-4f06dec29c9c>:1: UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see

<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

Conclusiones

Escriba sus conclusiones aquí...

Para la variable INGRESO, construya un histograma solamente con las

- ▼ informaciones de las personas con rendimiento hasta \$ 20.000,00 (miles de pesos).

```
ax = sns.distplot(datos.query('Ingreso < 20000')['Ingreso'])
ax.figure.set_size_inches(14, 6)
ax.set_title('Distribución de Frecuencias - INGRESO - Personas con ingreso hasta $ 20.000,00')
ax.set_xlabel('Miles de pesos colombianos', fontsize=14)
ax
```

```
<ipython-input-13-93edb7b317a4>:1: UserWarning:
```

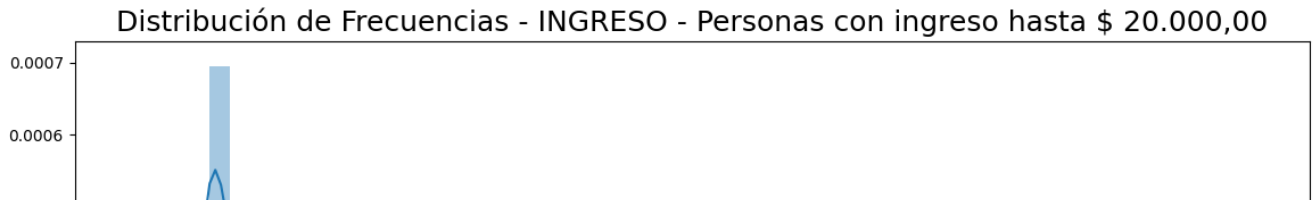
```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see

<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
ax = sns.distplot(datos.query('Ingreso < 20000')['Ingreso'])
<Axes: title={'center': 'Distribución de Frecuencias - INGRESO - Personas con ingreso hasta $ 20.000,00'}, xlabel='Miles de pesos colombianos', ylabel='Density'>
```



Construya una tabla de frecuencias y una con los porcentajes cruzando las variables SEXO y COLOR

Evalue el resultado de la tabla y escriba sus conclusiones principales

Utilize los diccionarios abajo para renombrar las líneas y columnas de las tablas de frecuencias y de los gráficos en nuestro proyecto

```
sexo = {
    0: 'Masculino',
    1: 'Femenino'
}
color = {0: 'Indio',
        2: 'Blanco',
        4: 'Negro',
        6: 'Amarillo',
        8: 'Pardo',
        9: 'Sin declarar'}
anos_de_estudio = {
    1: 'Sin estudios y menos de 1 año',
    2: '1 año',
    3: '2 años',
    4: '3 años',
    5: '4 años',
    6: '5 años',
    7: '6 años',
    8: '7 años',
    9: '8 años',
    10: '9 años',
    11: '10 años',
```

```

12: '11 años',
13: '12 años',
14: '13 años',
15: '14 años',
16: '15 años ou más',
17: 'No se sabe'
}

```

```

frecuencia = pd.crosstab(datos.Sexo,
                        datos.Color
                        )
frecuencia.rename(index = sexo, inplace = True)
frecuencia.rename(columns = color, inplace = True)
frecuencia

```

	Color	Indio	Blanco	Negro	Amarillo	Pardo
Sexo						
Masculino		256	22194	5502	235	25063
Femenino		101	9621	2889	117	10862

```

porcentaje = pd.crosstab(datos.Sexo,
                        datos.Color,
                        normalize = True
                        ) * 100
porcentaje.rename(index = sexo, inplace = True)
porcentaje.rename(columns = color, inplace = True)
porcentaje

```

	Color	Indio	Blanco	Negro	Amarillo	Pardo
Sexo						
Masculino		0.333160	28.883394	7.160333	0.305830	32.617126
Femenino		0.131442	12.520822	3.759761	0.152264	14.135867

▼ Realize, para la variable INGRESO, un análisis descriptivo con las herramientas que aprendimos en nuestro entrenamiento.

▼ Obtenga la media aritmética

```
datos.Ingreso.mean()
```

```
2000.3831988547631
```

▼ Obtenga la mediana

```
datos.Ingreso.median()
```

```
1200.0
```

▼ Obtenga la moda

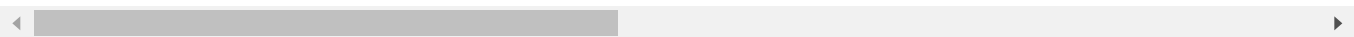
```
datos.Ingreso.mode()[0]
```

```
788
```

▼ Obtenga la desviación media absoluta

```
datos.Ingreso.mad()
```

```
<ipython-input-20-1ddc9cda72c3>:1: FutureWarning: The 'mad' method is deprecated and will be removed in a future version of pandas.  
datos.Ingreso.mad()  
1526.4951371638058
```



▼ Obtenga la varianza

```
datos.Ingreso.var()
```

```
11044906.006217021
```

▼ Obtenga la desviación estandar

```
datos.Ingreso.std()
```

```
3323.3877303464037
```

▼ Obtenga la media, mediana y valor máximo de la variable INGRESO según el SEXO y el COLOR

Destaque los puntos más importantes que usted observa en las tabulaciones

El parámetro *aggfunc* de la función *crosstab()* puede recibir una lista de funciones.

Ejemplo: *aggfunc* = {'mean', 'median', 'max'}

```
ingreso_estadisticas_por_sexo_y_color = pd.crosstab(datos.Color,
                                                    datos.Sexo,
                                                    values = datos.Ingreso,
                                                    aggfunc = {'mean', 'median', 'max'})
ingreso_estadisticas_por_sexo_y_color.rename(index = color, inplace = True)
ingreso_estadisticas_por_sexo_y_color.rename(columns = sexo, inplace = True)
ingreso_estadisticas_por_sexo_y_color
```

	max		mean		median	
Sexo	Masculino	Femenino	Masculino	Femenino	Masculino	Femenino
Color						
Indio	10000	120000	1081.710938	2464.386139	797.5	788.0
Blanco	200000	100000	2925.744435	2109.866750	1700.0	1200.0
Negro	50000	23000	1603.861687	1134.596400	1200.0	800.0
Amarillo	50000	20000	4758.251064	3027.341880	2800.0	1500.0
Pardo	100000	30000	1659.577425	1176.758516	1200.0	800.0

Conclusiones

Escriba sus conclusiones aquí...

Obtenga las medidas de dispersión de la variable INGRESO según el SEXO y el COLOR

Destaque los puntos más importantes que usted observa en las tabulaciones

O parámetro *aggfunc* de la función *crosstab()* puede recibir una lista de funciones.

Ejemplo: *aggfunc* = {'mad', 'var', 'std'}

```
ingreso_dispersion_por_sexo_y_color = pd.crosstab(datos.Color,
                                                    datos.Sexo,
                                                    aggfunc = {'mad', 'var', 'std'},
                                                    values = datos.Ingreso).round(2)
ingreso_dispersion_por_sexo_y_color.rename(index = color, inplace = True)
```

```
ingreso_dispersion_por_sexo_y_color.rename(columns = sexo, inplace = True)
ingreso_dispersion_por_sexo_y_color
```

```
<ipython-input-24-921e2e5dbfd9>:1: FutureWarning: The 'mad' method is deprecated and will
ingreso_dispersion_por_sexo_y_color = pd.crosstab(datos.Color,
<ipython-input-24-921e2e5dbfd9>:1: FutureWarning: The 'mad' method is deprecated and will
ingreso_dispersion_por_sexo_y_color = pd.crosstab(datos.Color,
<ipython-input-24-921e2e5dbfd9>:1: FutureWarning: The 'mad' method is deprecated and will
ingreso_dispersion_por_sexo_y_color = pd.crosstab(datos.Color,
<ipython-input-24-921e2e5dbfd9>:1: FutureWarning: The 'mad' method is deprecated and will
ingreso_dispersion_por_sexo_y_color = pd.crosstab(datos.Color,
<ipython-input-24-921e2e5dbfd9>:1: FutureWarning: The 'mad' method is deprecated and will
ingreso_dispersion_por_sexo_y_color = pd.crosstab(datos.Color,
<ipython-input-24-921e2e5dbfd9>:1: FutureWarning: The 'mad' method is deprecated and will
ingreso_dispersion_por_sexo_y_color = pd.crosstab(datos.Color,
<ipython-input-24-921e2e5dbfd9>:1: FutureWarning: The 'mad' method is deprecated and will
ingreso_dispersion_por_sexo_y_color = pd.crosstab(datos.Color,
<ipython-input-24-921e2e5dbfd9>:1: FutureWarning: The 'mad' method is deprecated and will
ingreso_dispersion_por_sexo_y_color = pd.crosstab(datos.Color,
<ipython-input-24-921e2e5dbfd9>:1: FutureWarning: The 'mad' method is deprecated and will
ingreso_dispersion_por_sexo_y_color = pd.crosstab(datos.Color,
```

	mad		std		var	
Sexo	Masculino	Femenino	Masculino	Femenino	Masculino	Femenino
Color						
Indio	798.91	3007.89	1204.09	11957.50	1449841.13	1.429818e+08
Blanco	2261.01	1670.97	4750.79	3251.01	22570023.41	1.056909e+07
Negro	975.60	705.45	1936.31	1349.80	3749293.59	1.821960e+06
Amarillo	3709.60	2549.15	5740.82	3731.17	32957069.62	1.392166e+07
Pardo	1125.83	811.58	2312.09	1596.23	5345747.15	2.547960e+06

Conclusiones

Escriba sus conclusiones aquí...

► Construya um box plot de la variable INGRESO según SEXO y COLOR

¿Es posible verificar algún comportamiento diferenciado en el rendimiento entre los grupos de personas analizados? Evalúe el gráfico y destaque los puntos más importantes.

1º - Utilice solamente las informaciones de personas con ingreso abajo de \$ 10.000

2º - Para incluir una tercera variable en la construcción de un boxplot utilice el parámetro *hue* e indique la variable que quiere incluir en la subdivisión.

Más informaciones: <https://seaborn.pydata.org/generated/seaborn.boxplot.html>

```
ax = sns.boxplot(x = 'Ingreso', y = 'Color', hue = 'Sexo', data=datos.query('Ingreso < 10000')

ax.figure.set_size_inches(14, 8)    # Personalizando el tamaño de la figura

ax.set_title('Box-plot del INGRESO por SEXO y COLOR', fontsize=18)    # Configurando el título

ax.set_xlabel('Miles de pesos colombianos', fontsize=14)    # Configurando el label del eje X

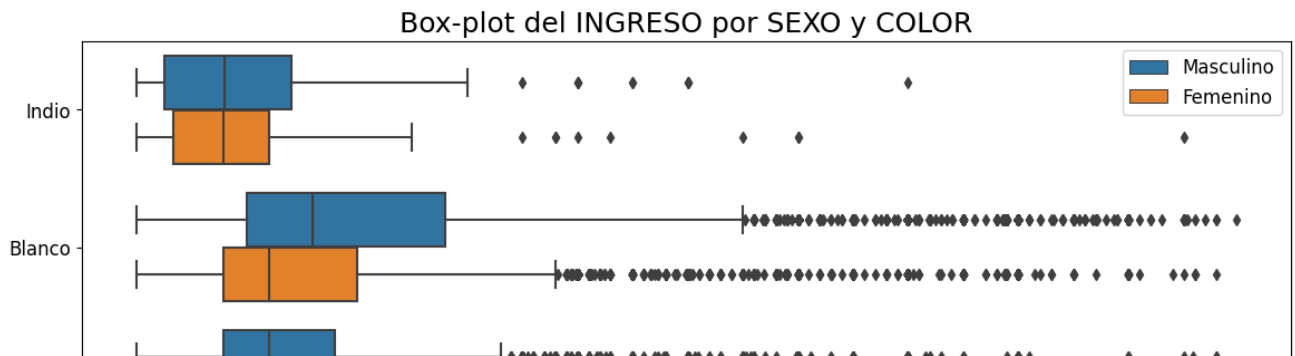
ax.set_ylabel('Color', fontsize=14)    # Configurando el label del eje Y
ax.set_yticklabels(['Indio', 'Blanco', 'Negro', 'Amarillo', 'Pardo'], fontsize=12)
    # Configurando el label de cada categoría del eje Y

# Configuraciones de la leyenda del gráfico (Sexo)
handles, _ = ax.get_legend_handles_labels()
ax.legend(handles, ['Masculino', 'Femenino'], fontsize=12)

ax
```



```
<Axes: title={'center': 'Box-plot del INGRESO por SEXO y COLOR'}, xlabel='Miles de pesos colombianos', ylabel='Color'>
```



Conclusiones

La mayoría de los resultados en los boxplot, es notorio que los hombres ganan más que los hombres, en el caso de los indios es menos notorio pero si existe ese comportamiento.

¿Cuál es el porcentaje de personas de nuestro *dataset* que ganan un salario mínimo (\$ 788,00) o menos?

Utilize la función `percentileofscore()` do *scipy* para realizar estos análisis.

Más informaciones:

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.percentileofscore.htm>

```
from scipy import stats
```

```
porcentaje = stats.percentileofscore(datos.Ingreso, 788, kind = 'weak')
print("{0:.2f}%".format(porcentaje))
```

28.87%

¿Cuál es el valor máximo ganado por 99% de las personas de nuestro *dataset*?

Utilice el método `quantile()` de *pandas* para realizar estos análisis.

```
valor = datos.Ingreso.quantile(.99)
print("$ {0:.2f}".format(valor))
```

\$ 15000.00

Obtenga la media, mediana, valor máximo y desviación estandar de la variable INGRESO según AÑOS DE ESTUDIO y SEXO

Destaque los puntos más importantes que usted observa en las Tabulaciones

O parámetro *aggfunc* de la función *crosstab()* puede recibir una lista de funciones.

Ejemplo: *aggfunc = ['mean', 'median', 'max', 'std']*

```
ingreso_estadisticas_porsexo_y_estudo = pd.crosstab(datos['Años de Estudio'],
                                                    datos.Sexo,
                                                    aggfunc = {'mean', 'median', 'max', 'std'},
                                                    values = datos.Ingreso).round(2)
ingreso_estadisticas_porsexo_y_estudo.rename(index = anos_de_estudio, inplace = True)
ingreso_estadisticas_porsexo_y_estudo.rename(columns = sexo, inplace = True)
ingreso_estadisticas_porsexo_y_estudo
```

max

mean

median

std

▼ Construya un box plot de la variable INGRESO según AÑOS DE ESTUDIO y SEXO

¿Es posible verificar algún comportamiento diferenciado en el rendimiento entre los grupos de personas analizadas? Evalúe el gráfico y destaque los puntos más importantes.

1º - Utilice solamente las informaciones de personas con ingreso abajo de \$ 10.000

2º - Utilice la variable EDAD para identificar si la desigualdad se verifica para personas de la misma edad. Ejemplo: `data=datos.query('Ingreso < 10000 and Edad == 40')` ou `data=datos.query('Ingreso < 10000 and Edad == 50')`

3º - Para incluir una tercera variable en la construcción de un boxplot utilice el parámetro `hue` e indique la variable que quiere incluir en la subdivisión.

Más informaciones: <https://seaborn.pydata.org/generated/seaborn.boxplot.html>

```

# ---
20000      40000      40000 40      022.60      40000      0000      4545.50      00
ax = sns.boxplot(x = 'Ingreso', y = 'Años de Estudio', hue = 'Sexo', data=datos.query('Ingres
ax.figure.set_size_inches(14, 8)      # Personalizando el tamaño de la figura

ax.set_title('Box-plot del INGRESO por SEXO y AÑOS DE ESTUDIO', fontsize=18)      # Configurand
ax.set_xlabel('Miles de pesos colombianos', fontsize=14)      # Configurando el label del eje X

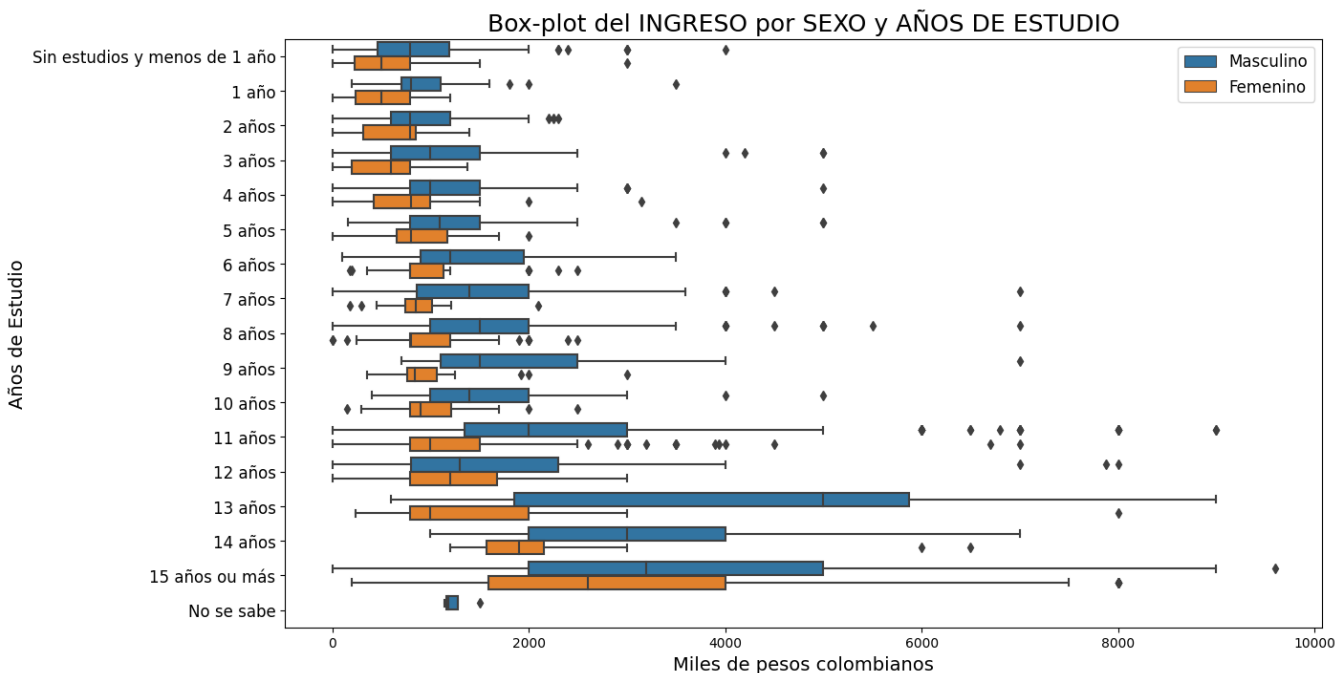
ax.set_ylabel('Años de Estudio', fontsize=14)      # Configurando el label del eje Y
ax.set_yticklabels([key for key in anos_de_estudio.values()], fontsize=12)      # Configurando

# Configurações da legenda do gráfico (Sexo)
handles, _ = ax.get_legend_handles_labels()
ax.legend(handles, ['Masculino', 'Femenino'], fontsize=12)

ax

```

```
<Axes: title={'center': 'Box-plot del INGRESO por SEXO y AÑOS DE ESTUDIO'},
xlabel='Miles de pesos colombianos', ylabel='Años de Estudio'>
```



Conclusiones

Sigue la tendencia de que los hombres tienen más años de estudio que las mujeres, es más notorio en el grupo de 13 años, ciertamente esto es un problema social que debe ser tomado en cuenta.

Obtenga la media, mediana, valor máximo y desviación estandar de la variable INGRESO según las CIUDADES

Destaque los puntos más importantes que usted observa en las tabulaciones

Utilice el método `groupby()` de `pandas` conjuntamente con el método `agg()` para contruir la tabulación. El método `agg()` puede recibir un diccionario especificando cual columna del DataFrame deve ser utilizada y cual lista de funciones estadísticas queremos obtener, por ejemplo: `datos.groupby(['Ciudad']).agg({'Ingreso': ['mean', 'median', 'max', 'std']})`

```
ingreso_estadisticas_por_ciudad = datos.groupby(['Ciudad']).agg({'Ingreso': ['mean', 'median', 'max', 'std']})
ingreso_estadisticas_por_ciudad
```



Ciudad	Ingreso			
	mean	median	max	std
11	1789.761223	1200.0	50000	2406.161161
12	1506.091782	900.0	30000	2276.233415
13	1445.130100	900.0	22000	1757.935591
14	1783.588889	1000.0	20000	2079.659238
15	1399.076871	850.0	50000	2053.779555
16	1861.353516	1200.0	15580	2020.688632
17	1771.094946	1000.0	60000	2934.590741
21	1019.432009	700.0	30000	1887.816905
22	1074.550784	750.0	40000	2373.355726
23	1255.403692	789.0	25000	1821.963536
24	1344.721480	800.0	15500	1651.805500
25	1293.370487	788.0	30000	1950.272431
26	1527.079319	900.0	50000	2389.622497
27	1144.552602	788.0	11000	1237.856197
28	1109.111111	788.0	16000	1478.997878
29	1429.645094	800.0	200000	3507.917248
31	2056.432084	1200.0	100000	3584.721547
32	2026.383852	1274.0	100000	3513.846868
33	2496.403168	1400.0	200000	5214.583518
35	2638.104986	1600.0	80000	3503.777366
41	2493.870753	1500.0	200000	4302.937995
42	2470.854945	1800.0	80000	3137.651112
43	2315.158336	1500.0	35000	2913.335783
50	2262.604167	1500.0	42000	3031.419122
51	2130.652778	1500.0	35000	2512.630178

▼ Construya un box plot de la variable INGRESO según las CIUDADES

¿Es posible verificar algún comportamiento diferenciado en el rendimiento entre los grupos analizados? Evalúe el gráfico y destaque los puntos más importantes.

1º - Utilice solamente las informaciones de personas con ingreso abajo de \$ 10.000

```
ax = sns.boxplot(x = 'Ingreso', y = 'Ciudad', data=datos.query('Ingreso < 10000'), orient='h')

ax.figure.set_size_inches(14, 8)    # Personalizando o tamaño de la figura

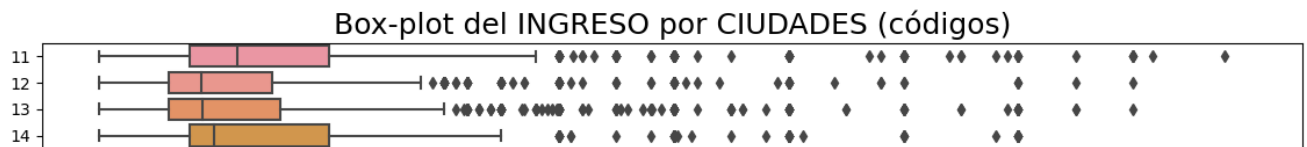
ax.set_title('Box-plot del INGRESO por CIUDADES (códigos)', fontsize=18)    # Configurando el

ax.set_xlabel('Miles de pesos colombianos', fontsize=14)    # Configurando el label del eje X

ax.set_ylabel('Códigos de las ciudades', fontsize=14)    # Configurando el label del eje Y
#ax.set_yticklabels([key for key in ciudad.values()], fontsize=12)    # Configurando el label

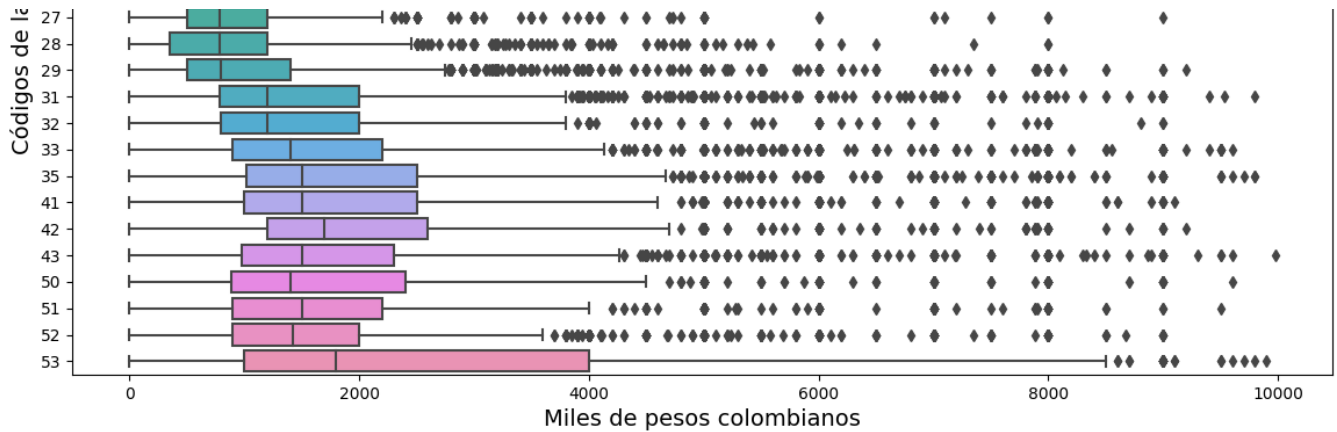
ax
```

```
<Axes: title={'center': 'Box-plot del INGRESO por CIUDADES (códigos)'}, xlabel='Miles de pesos colombianos', ylabel='Códigos de las ciudades'>
```



Conclusiones

No se tienen datos o nombres de las ciudades, tienen un comportamiento similar en los cuantiles, existen muchos outliers que son representativos en la muestra de datos.



✓ 1 s se ejecutó 17:40

