



Python con Excel II Gráficos básicos, Caso histórico de RRHH

Graficos con Python en Excel, creando un dataframe vinculado a una conexión de Power Query en Excel. Convirtiendo graficos de Python en interactivos basados en segmentadores de tablas/graficos dinámicos.

Jose Ignacio González Gómez

Departamento de Economía, Contabilidad y Finanzas - Universidad de La Laguna

www.jggomez.eu

V. 2.5

Ejercicio basado : [NEW Python in Excel - PYTHON + EXCEL + ChatGPT = Easy!](#)

Contenido

Introducción a los gráficos de Python en Excel.....	2
Caso Histórico de RRHH Gráficos básicos. Mynda Treacy.....	2
Presentacion	2
Información disponible	2
Librerías graficas necesarias, seaborn	2
Análisis con Python para Excel.....	3
Fase I. Importar librerías necesarias, pandas y seaboron	3
Fase II. Cargar Datos. Dataframe.....	3
Fase III. Estadísticas básicas, general y seleccionada.	4
Fase IV. Algunos graficos de la biblioteca seaborn	5

Introducción a los gráficos de Python en Excel

Uno de los mejores usos de Python en Excel es crear los increíbles gráficos disponibles a través de bibliotecas de Python como [Seaborn](#) y [Matplotlib](#). Estas bibliotecas abren una gran variedad de gráficos que simplemente no podemos hacer con los gráficos integrados de Excel.

Sobre la librería Seaborn buscar documentación e ideas....

Caso Histórico de RRHH Gráficos básicos. Mynda Treacy.

Fuente: [NEW Python in Excel - PYTHON + EXCEL + ChatGPT = Easy!](#)

Presentacion

Con este caso vamos a llevar a cabo graficos básicos con Python y Excel haciendo uso de la librería seaborn

Información disponible

Vamos a trabajar con un fichero correspondiente a un informe del departamento de Recursos Humanos de una empresa que contiene los registros básicos del histórico de los trabajadores (1470 registros). El fichero llamado Graficos Interactivos “Mynda Treacy Original” que tiene formato csv, presenta la siguiente estructura con los campos o columnas disponibles.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Edad	Rotación	Viajes de neg	Tarifa diaria	Departamen	Distancia de	Educación	Campo de ec	Número de e	Satisfacción	Género	Tarifa por hor	Implicación	Nivel laboral	Rol laboral	Satisfacción	Estado civil
2	41	Yes	Poco Frecue	1102	Comercial	1	Estudios univ	Biología, eco	1	2	Femenino	94	3	2	Ejecutivo de	4	Soltero
3	49	No	Frecuente	279	I+D	8	Secundaria c	Biología, eco	2	3	Masculina	61	2	2	Investigador	2	Casado
4	37	Yes	Poco Frecue	1373	I+D	2	Estudios univ	Otras	4	4	Masculina	92	2	1	Técnico de la	3	Soltero

	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI
1	Ingresos mer	Tarifa mensu	Número de e	Mayores de 1	Horas extra	Porcentaje d	Calificación	Satisfacción	Horas estándar	Nivel de opci	Total de años	Tiempos de fi	Conciliación	Años en la e	Años en el pu	Años desde el i	Años con el gerente actual	
2	5993	19479	8	Y	Yes	11	3	1	80	0	8	0	1	6	4	0	5	
3	5130	24907	1	Y	No	23	4	4	80	1	10	3	3	10	7	1	7	
4	2090	2396	6	Y	Yes	15	3	2	80	0	7	3	3	0	0	0	0	

Ilustración 1

1	Edad	18	Ingresos mensuales
2	Rotación	19	Tarifa mensual
3	Viajes de negocios	20	Número de empresas trabajadas
4	Tarifa diaria	21	Mayores de 18 años
5	Departamento	22	Horas extra
6	Distancia desde casa	23	Porcentaje de aumento salarial
7	Educación	24	Calificación del rendimiento
8	Campo de educación	25	Satisfacción con la relación
9	Número de empleados	26	Horas estándar
10	Satisfacción con el entorno laboral	27	Nivel de opciones sobre acciones
11	Género	28	Total de años trabajados
12	Tarifa por hora	29	Tiempos de formación del año pasado
13	Implicación laboral	30	Conciliación entre la vida laboral y personal
14	Nivel laboral	31	Años en la empresa
15	Rol laboral	32	Años en el puesto actual
16	Satisfacción laboral	33	Años desde el último ascenso
17	Estado civil	34	Años con el gerente actual

Tabla 1

Librerías graficas necesarias, seaborn

Debemos cargar las librerías de graficas sobre las cuales vamos a trabajar, pero podríamos confirmar previamente que estas por defecto se han cargado cuando inicializamos Python en Excel y eso se muestra en la Ilustración 2

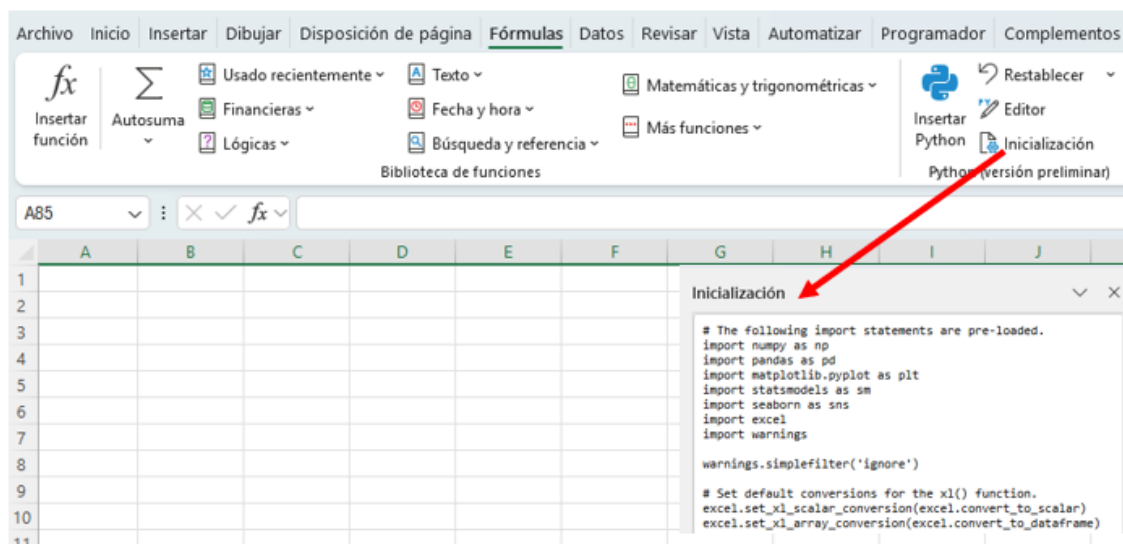


Ilustración 2

Confirmamos que la librería de gráficos con los que vamos a trabajar está cargada al inicializarse Python en Excel y es seaborn ([seaborn 0.13.2 documentation](https://seaborn.pydata.org/seaborn/doc/0.13.2.html))

Análisis con Python para Excel

Fase I. Importar librerías necesarias, pandas y seaborn

En caso de que la librerías que necesitamos para nuestro proyecto no este cargada, este sería nuestro primer paso, vamos a garantizar la importación de las librerías

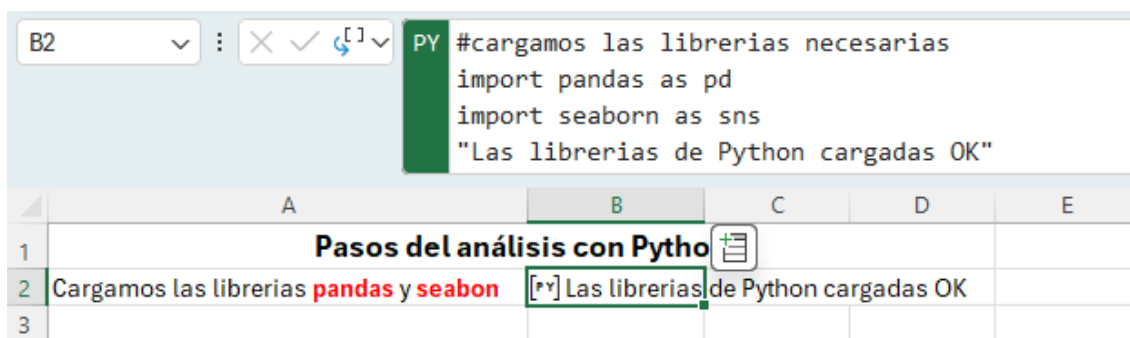


Ilustración 3

En este caso el código es fácil de interpretar, importamos dos librerías esenciales para análisis y visualización de datos. El mensaje final es solo informativo y no afecta la ejecución del código.

Fase II. Cargar Datos. Dataframe

Pasamos a definir el Dataframe, para lo cual seleccionamos la tabla que contiene en la hoja "Tabla" llamada TablaDatos y seleccionamos todo incluidos los encabezados

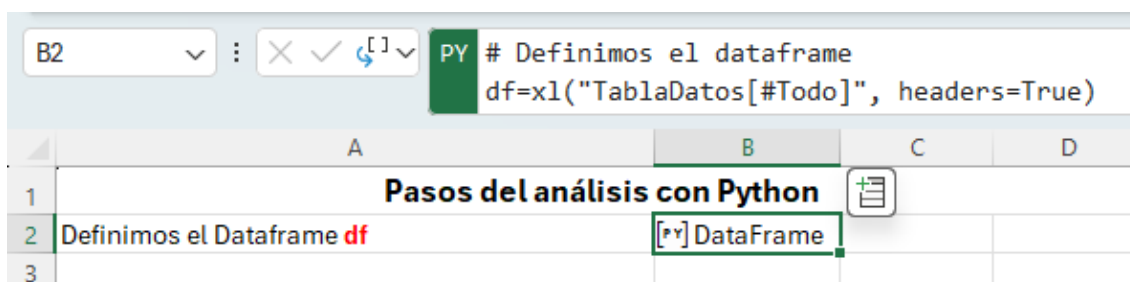


Ilustración 4

El script Python:

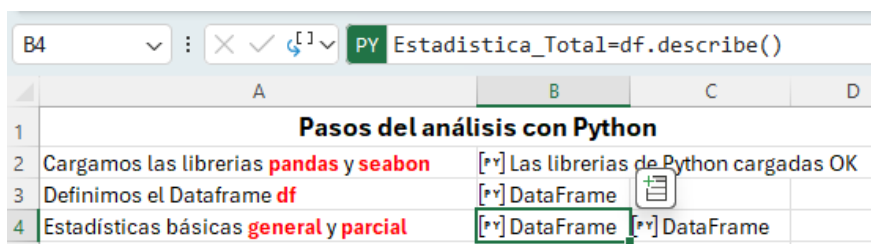
```
# Definimos el dataframe
df=xl("TablaDatos[#Todo]", headers=True)
```

Código comentado

Estamos creando (definiendo) un **DataFrame** que contiene **todos los datos** de la tabla llamada "TablaDatos" en tu archivo de Excel, con los encabezados correctamente asignados.

Fase III. Estadísticas básicas, general y seleccionada.

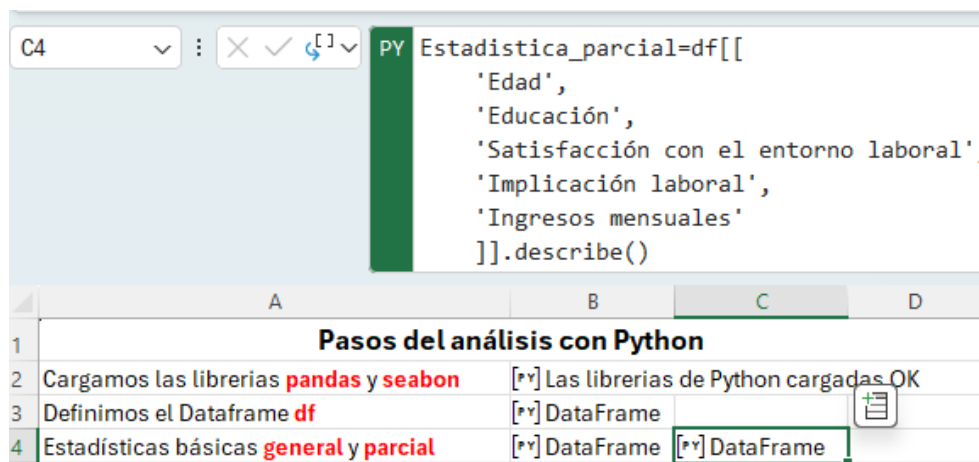
Vamos a presentar las estadísticas descriptivas básicas, tanto general como para un conjunto de columnas seleccionadas. Para ello vamos a utilizar la clase describe



	A	B	C	D
1	Pasos del análisis con Python			
2	Cargamos las librerías pandas y seaborn	[✓] Las librerías de Python cargadas OK		
3	Definimos el Dataframe df	[✓] DataFrame		
4	Estadísticas básicas general y parcial	[✓] DataFrame	[✓] DataFrame	

Ilustración 5

En este caso nos va a dar la estadísticas descriptiva de todos los elementos numéricos de nuestro dataframe. Si quisiéramos disponer de esta estadística solo para un conjunto de columnas, campos o variables tendríamos que seleccionarlos tal y como se muestra en el siguiente script, donde como podemos observar en la Ilustración 6 separamos los campos deseados por comas.



	A	B	C	D
1	Pasos del análisis con Python			
2	Cargamos las librerías pandas y seaborn	[✓] Las librerías de Python cargadas OK		
3	Definimos el Dataframe df	[✓] DataFrame		
4	Estadísticas básicas general y parcial	[✓] DataFrame	[✓] DataFrame	

Ilustración 6

Como el resultado que genera es una tabla de filas y columnas hemos decidido desplegarla en otra hoja nueva que hemos llamado Estadística que nos va permitir una mejor visualización.

B2											
x1('Codigo Python'!B4")											
A	B	D	E	F	G	H	I	J	K	L	
Estadística descriptiva general											
	Edad	Tarifa diaria	Distancia desde	Educación	Número de e	Satisfacción	Tarifa por hc	Implicación	Nivel laboral	Satisfacción	
count	1470	1470	1470	1470	1470	1470	1470	1470	1470	1470	
mean	36,9238095	802,485714	9,192517007	2,91292517	1024,86531	2,72176871	65,8911565	2,72993197	2,06394558	2,72857143	
std	9,13537349	403,5091	8,106864436	1,02416494	602,024335	1,09308221	20,3294276	0,71156114	1,1069399	1,10284612	
min	18	102	1	1	1	1	30	1	1	1	
25%	30	465	2	2	491,25	2	48	2	1	2	
50%	36	802	7	3	1020,5	3	66	3	2	3	
75%	43	1157	14	4	1555,75	4	83,75	3	3	4	
max	60	1499	29	5	2068	4	100	4	5	4	
B13											
x1('Codigo Python'!C4")											
Estadística descriptiva seleccionada											
	Edad	Educación	Satisfacción cor	Implicación	Ingresos mensuales						
count	1470	1470	1470	1470	1470						
mean	36,9238095	2,91292517	2,721768707	2,72993197	6502,93129						
std	9,13537349	1,02416494	1,093082215	0,71156114	4707,95678						
min	18	1	1	1	1009						
25%	30	2	2	2	2911						
50%	36	3	3	3	4919						
75%	43	4	4	3	8379						
max	60	5	4	4	19999						

Ilustración 7

Así destaca en primer lugar que tenemos en la celda B2 tenemos una fórmula en Python que hace referencia al contenido que ya hemos descrito anteriormente y que se corresponde con la Ilustración 6 y lo desplegamos aquí (punto 2) para desplegar su contenido.

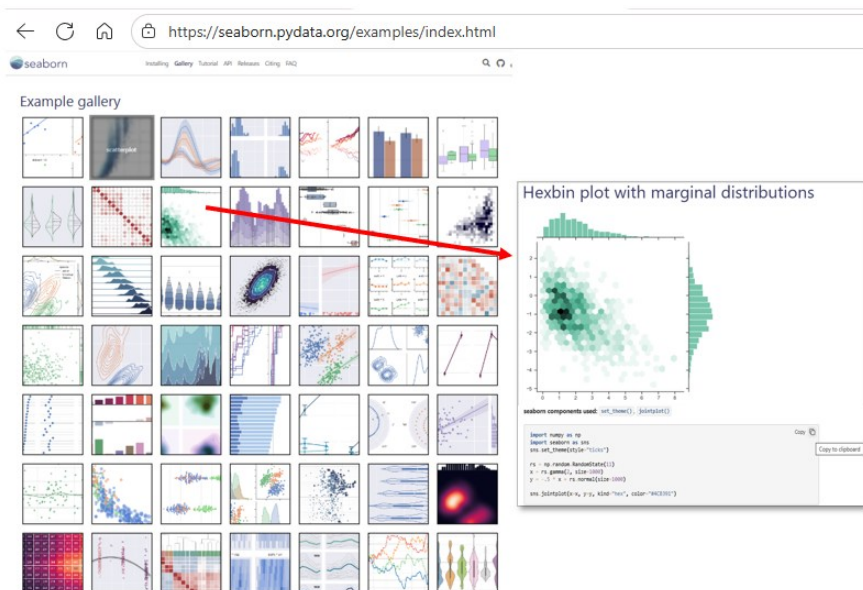
Igualmente en la celda B13 de la Ilustración 7 la tendremos asociada al dataframe definido en la celda c4 de la hoja Código Python y se encuentra en este sitio desplegada para una mejor visualización.

Fase IV. Algunos gráficos de la biblioteca seaborn

Seaborn Histogram

Gráfico de hexágonos (Hexbin plot)

Ya hemos cargado la biblioteca seaborn y procedemos a crear un gráfico tipo Hexbin plot que es un tipo de gráfico de hexágonos,



Un **Hexbin plot** (gráfico de hexágonos) es una forma de visualizar la **densidad de puntos** en un gráfico de dispersión bidimensional, especialmente útil cuando tienes **muchos puntos superpuestos**. En lugar de mostrar cada punto individual, el espacio se divide en **hexágonos** y se cuenta cuántos puntos caen dentro de cada uno. Luego, se colorean los hexágonos según esa densidad.

Esto ayuda a evitar el problema de **overplotting** (cuando hay tantos puntos que se superponen y no se puede ver la densidad real de los datos). Vamos a aplicarlo a nuestro caso

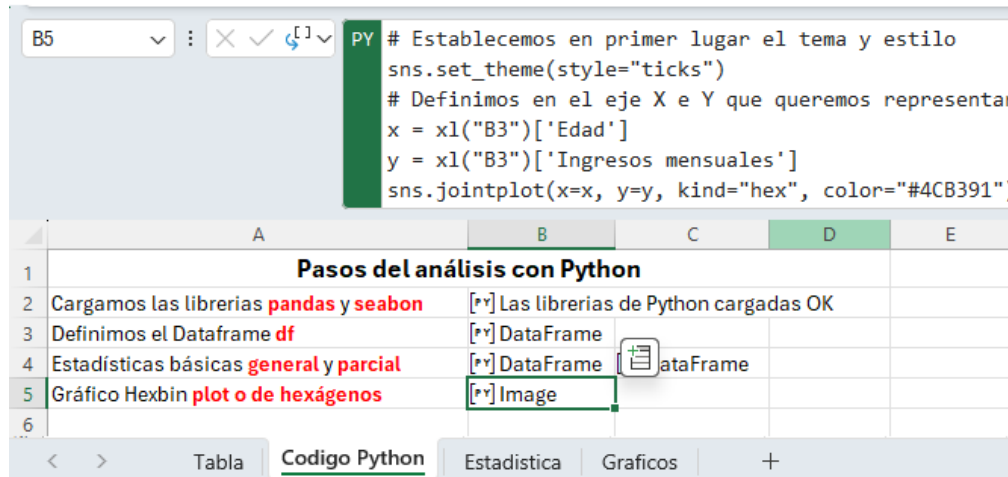


Ilustración 9

El script de Python es el siguiente:

```
# Establecemos en primer lugar el tema y estilo
sns.set_theme(style="ticks")
# Definimos en el eje X e Y que queremos representar
x = xl("B3")['Edad']
y = xl("B3")['Ingresos mensuales']
sns.jointplot(x=x, y=y, kind="hex", color="#4CB391")
```

Parámetros importantes:

- x, y: Las variables que quieres graficar.
- kind="hex": Especifica que quieres un gráfico de tipo hexbin.
- gridsize: Controla el tamaño de los hexágonos (por defecto es 50).
- cmap: Cambia la paleta de colores.

El resultado es el grafico propuesto y para una mejor visualización hemos creado una replica o referencia en una nueva hoja llamada Gráficos

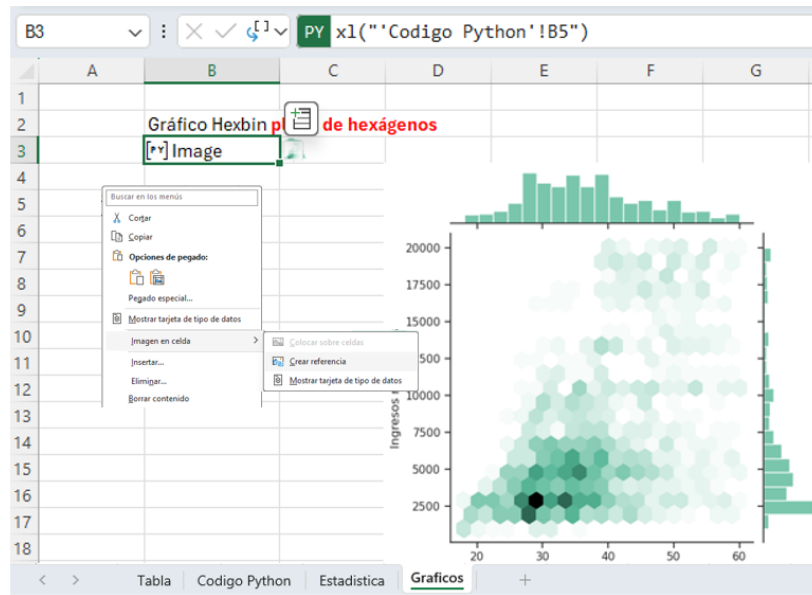


Ilustración 10

¿Cuándo usar un Hexbin Plot?

Usa un Hexbin plot cuando:

- Tienes **muchos datos** y un scatter plot tradicional se ve saturado.
- Quieres ver **zonas de alta densidad** de puntos.
- Estás trabajando con **datos continuos** en dos dimensiones.