



Python con Excel II Gráficos interactivos, 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 [Interactive Python Charts in Excel • My Online Training Hub](#)

Contenido

Introducción a los gráficos interactivos de Python en Excel	2
Caso Histórico de RRHH. Mynda Treacy	2
Presentacion	2
Información disponible	2
Creando nuestro fichero de trabajo y la conexión a la fuente de datos.	3
Creacion de TD y segmentador	3
Análisis con Python para Excel.....	5
Fase I. Cargar Datos. Dataframe con conexión a una consulta	5
Fase II. Crear lista de valores seleccionados por el segmentador.	5
Fase III. Creamos el nuevo df_filtrado.....	7
Fase IV Grafico de burbujas.....	7

Introducción a los gráficos interactivos de Python en Excel

[Microsoft said it couldn't be done - Interactive Python Charts in Excel - YouTube](#)

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.

Tradicionalmente usamos los paneles o segmentadores para hacer nuestros gráficos dinámicos interactivos, vamos a presentar a continuación como aportar esta interactividad usando gráficos basados diseñados en Python en Excel ([Turn any idea into an app with Base44](#))

Caso Histórico de RRHH. Mynda Treacy

Fuente: [Microsoft said it couldn't be done - Interactive Python Charts in Excel - YouTube](#)

Presentacion

Con este caso vamos a llevar a cabo gráficos interactivos con Python y Excel haciendo uso de los filtros y segmentadores relacionados con las tablas dinámicas

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 Gráficos 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 l	Nivel laboral	Rol laboral	Satisfacción	Estado civil	
2	41	Yes	Poco Frecue	1102	Comercial	1	Estudios uni	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 uni	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 mens	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 en	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

Creando nuestro fichero de trabajo y la conexión a la fuente de datos.

Antes de comenzar a trabajar con los datos en Python en Excel vamos a crear nuestro fichero de trabajo en blanco y generar solo la conexión, comenzamos por tanto abriendo nuestro editor de consultas Power Query (PQ) y aplicamos las transformaciones básicas necesarias como proponer encabezado de columnas, tipo de datos, etc..

El resultado sera que contaremos solo con la conexión a la fuente no a la tabla y sobre esta conexión se basara la configuración del Dataframe

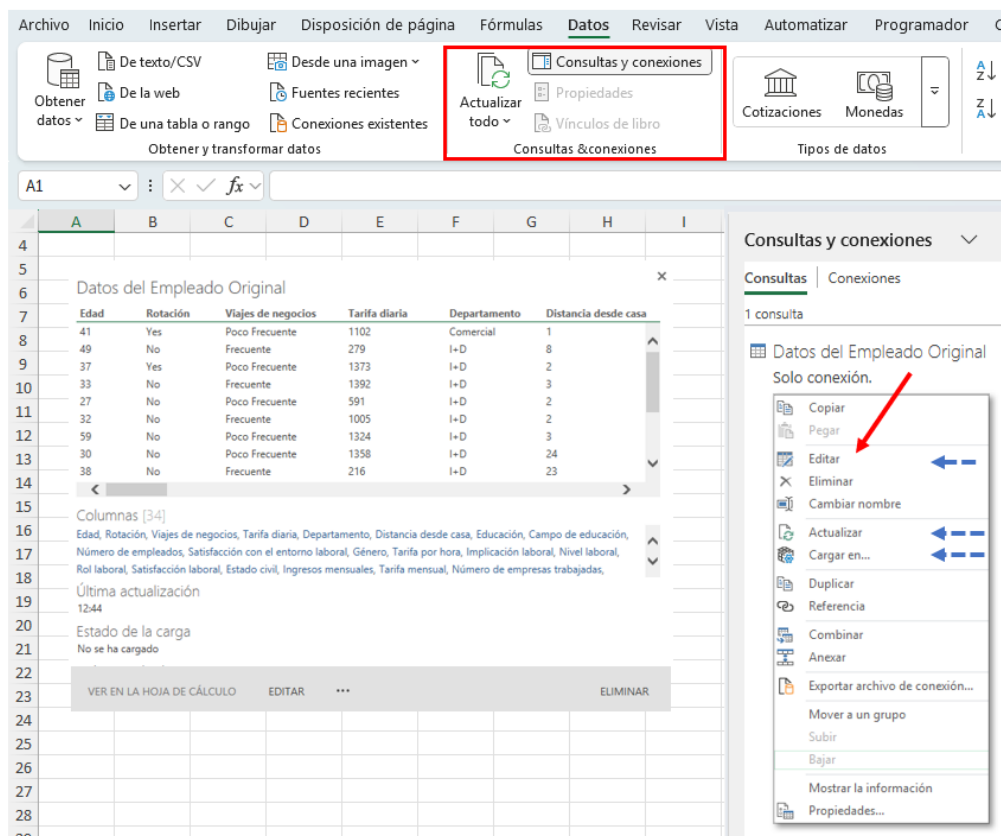


Ilustración 2

De esta forma tenemos creada la conexión a la fuente

Creacion de TD y segmentador

Como vamos a dar dinamismo a los graficos de Python en Excel, vamos a apoyarnos en los segmentadores disponibles cuando trabajamos con tablas dinámicas, por procedemos a crear una simple TD con un segmentador que nos va a permitir generar el dinamismo.

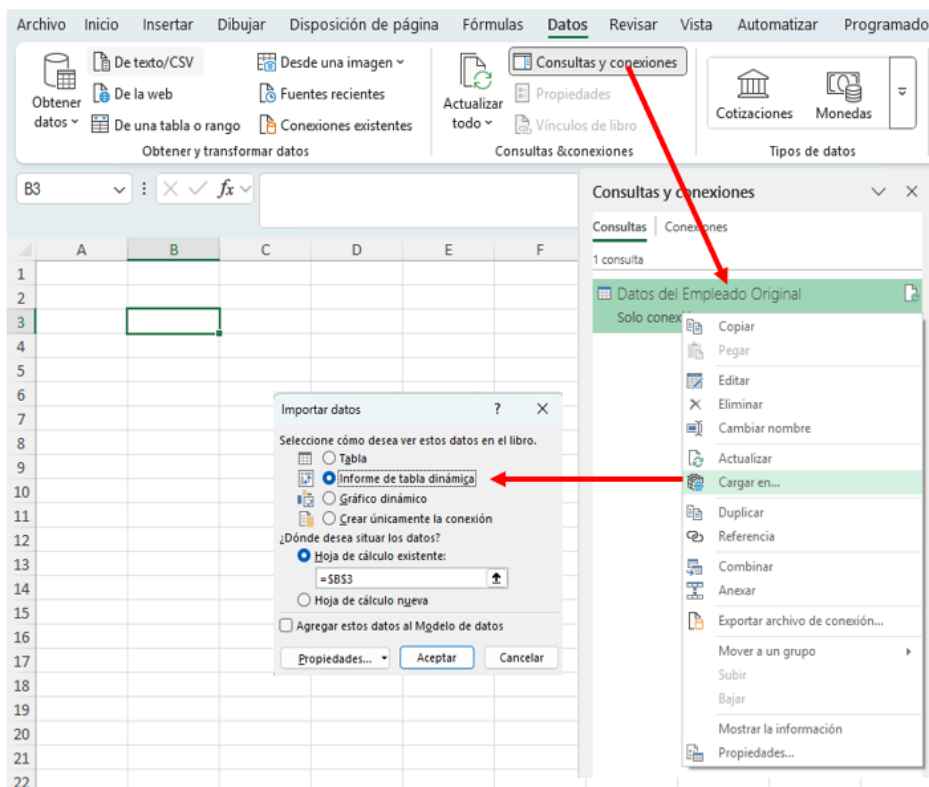


Ilustración 3

Así el resultado obtenido será el siguiente

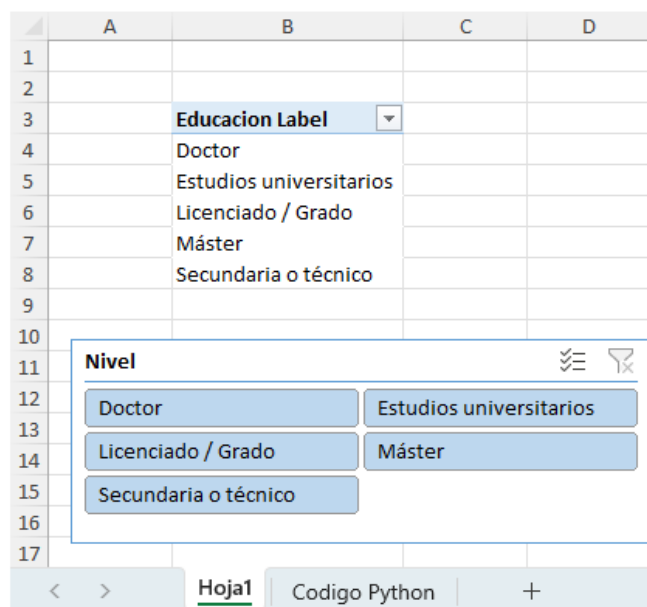


Ilustración 4

Una tabla dinámica que la utilizaremos como lista de valores y un segmentador que nos permitirá filtrar.

Análisis con Python para Excel

Fase I. Cargar Datos. Dataframe con conexión a una consulta

Comenzamos por tanto definiendo el Dataframe, para lo cual y en este caso tendremos que hacer referencia a la conexión contenida en el fichero.

Esto implica que nuestro Dataframe estará relacionado no con una tabla sino con una conexión que hemos creado previamente a la fuente de datos (Ilustración 2), para

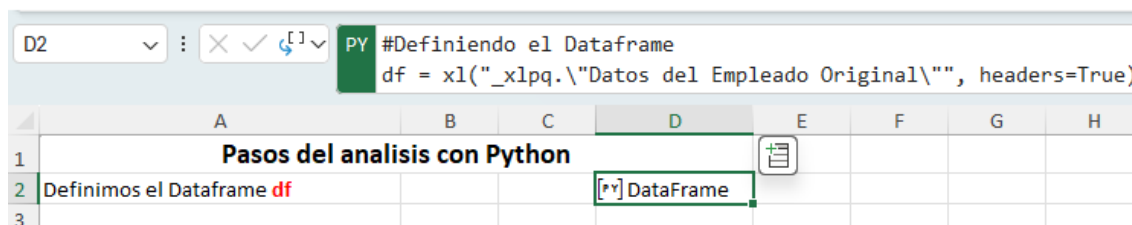


Ilustración 5

El script Python:

```
#Definiendo el Dataframe
df = xl("_xlpq.\"Datos del Empleado Original\"", headers=True)
```

Código comentado

Estamos creando (definiendo) un **DataFrame** (una estructura de datos tabular de pandas) a partir de una fuente de datos en Excel.

"_xlpq.\"Datos del Empleado Original\"" significa que accedemos a una **consulta de Power Query** que está cargada en la hoja de cálculo con el nombre . "Datos del Empleado Original".

Detalle

- df: Es el nombre que daremos al DataFrame que estamos creando y es donde se almacenarán los datos extraídos de Excel.
- xl(...): Ese xl no es una función estándar de Python, sino que viene del entorno de Python en Excel. Es una función especial que se usa en Python dentro de Excel para acceder a rangos de celdas. Esta función permite leer datos directamente desde una hoja de Excel y convertirlos en un DataFrame de pandas, lo cual es muy útil para análisis de dato. Es como si hicieras un `pd.DataFrame(...)`, pero el objeto ya se construye directamente con lo que tienes en la hoja de Excel
- "_xlpq.\"Datos del Empleado Original\"": Este es el nombre interno que Excel usa para referirse a una consulta de Power Query. El prefijo _xlpq. indica que se está accediendo a un **objeto de Excel** (como una tabla, un rango con nombre o una consulta)
- "Datos del Empleado Original" es probablemente el **nombre de una tabla** definida en la hoja de Excel.
- headers=True: esto le dice a Python que la primera fila del rango contiene los nombres de las columnas. Es útil para que el DataFrame tenga encabezados adecuados.

Fase II. Crear lista de valores seleccionados por el segmentador.

Lo que vamos a construir es una lista de valores seleccionados por el segmentador, un nuevo dataframe (df2) condicionado a los valores seleccionados según el segmentador y para ello creamos el siguiente script.

Pasos del análisis con Python

- Definimos el Dataframe **df**
- Definimos Lista **Selección** según segmentador

list 2x1

Doctor
Máster

Nivel

Doctor	Estudios universitarios
Licenciado / Grado	Máster
Secundaria o técnico	

Ilustración 6

```
df2=xl("Hoja1!B3:B8",headers=True)
# cuando se filtre y evitar los vacios creamos
#lista basada en los valores filtrados
Selección=df2['Educacion Label'].dropna().tolist()
```

Explicacion del codigo

Primero seleccionamos un nuevo dataframe correspondientes a los valores de la tabla dinámica, que están en la hoja 1 y que llamamos df2

Educacion Label

Doctor
Estudios universitarios
Licenciado / Grado
Máster
Secundaria o técnico

Nivel

Doctor	Estudios universitarios
Licenciado / Grado	Máster
Secundaria o técnico	

Ilustración 7

Como estos valores pueden cambiar según afecte el segmentador los convertimos en lista con el nombre de Selección.

```
Selección= df2['Educacion Label'].dropna().tolist()
```

Fase III. Creamos el nuevo df_filtrado

Una vez disponemos de la lista de vamos aplicarlo para obtener un nuevo dataframe (df_filtrado) con los registros seleccionados aplicando la lista anterior (selección).

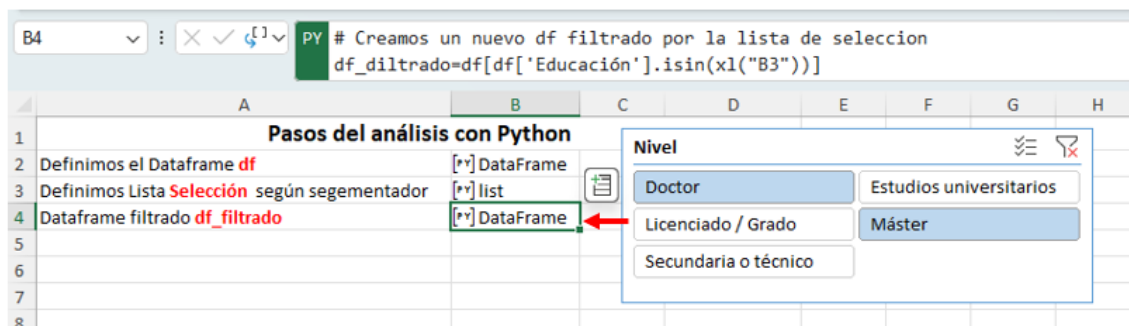


Ilustración 8

Creamos un nuevo df filtrado por la lista de seleccion
df_filtrado=df[df['Educación'].isin(xl('B3'))]

Explicacion del codigo.

- **df['Educación']** Aquí estás accediendo a la columna llamada "Educación" dentro del DataFrame original (df). Esta columna contiene los valores que vas a comparar con una lista en nuestro caso era selección.
- **xl("B3")** Lee el contenido de la celda de Excel, que contiene una lista de valores que hemos llamado Selección y que se corresponde con los elementos seleccionados del segmentador.
- **isin(...)** Esta función de pandas verifica si cada valor en la columna "Educación" está presente en la lista que devuelve xl("B3") Selección.
- **df_filtrado[...]** Es el resultado final done estás **filtrando el DataFrame** para quedarte solo con las filas donde el valor de la columna "Educación" **coincide con alguno de los valores** en la lista de la celda B3.

Fase IV Grafico de burbujas

Vamos a crear un gráfico de burbujas con las siguientes propiedades (ver Tabla 1):

- Eje X (Ingresos mensuales) representa el nivel de ingresos mensuales
- Eje Y será la edad
- Color de las burbujas se corresponde nivel laboral que va de 1 a 5
- Tamaño de la burbuja es el nivel de educación (Educación) filtrado o selecciona según segmentador.

El script de Python es el siguiente:

```
import seaborn as sns
sns.set_theme(style='white')
sns.relplot(
    x='Edad',
    y='Ingresos mensuales',
    hue='Nivel laboral',
    size='Educación',
    sizes=(40, 400),
    alpha=0.5,
    palette='muted',
    height=6,
    data=df_filtrado
)
```

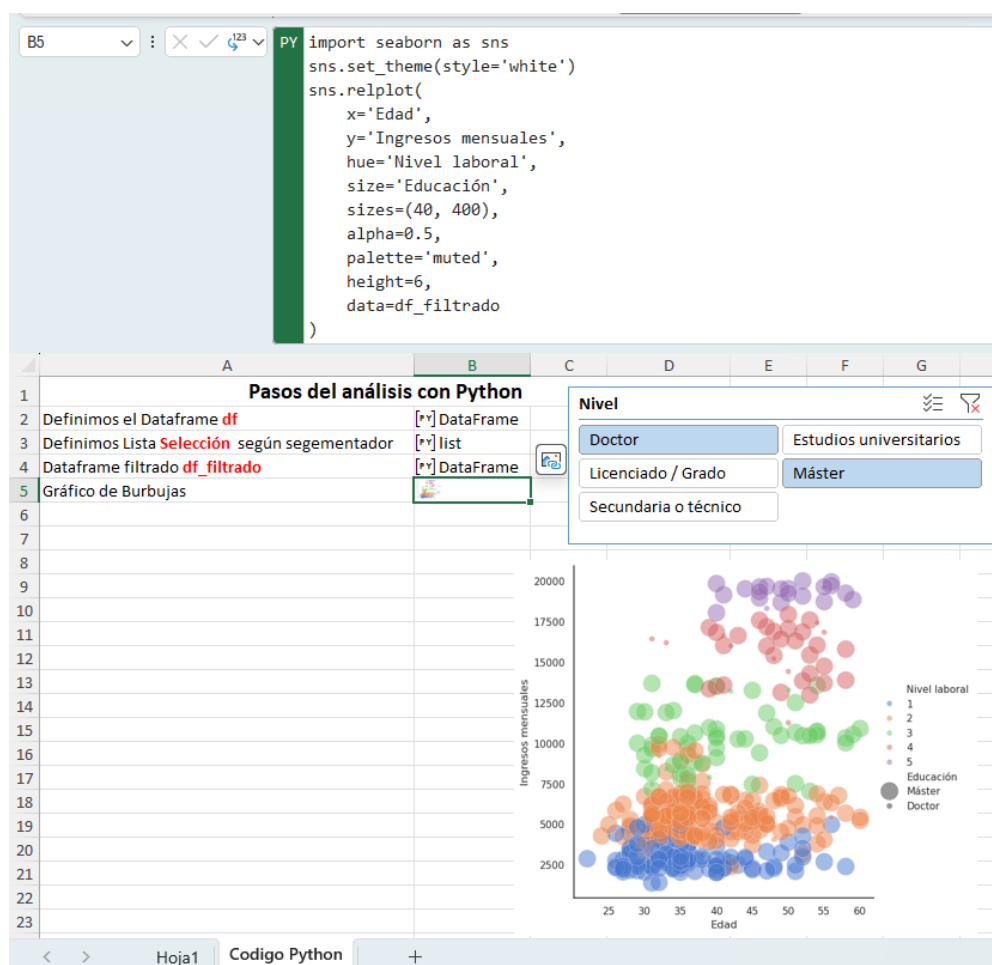


Ilustración 9

Código comentado

Este script utiliza Seaborn, una biblioteca de visualización basada en Matplotlib, para crear un gráfico relacional (scatter plot) con varias dimensiones visuales.

`sns.set_theme(style='white')`: Configura el estilo visual del gráfico. El estilo 'white' elimina los fondos grises y hace que el gráfico se vea más limpio.

A continuación, analizamos el bloque gráfico que crea un **gráfico de dispersión (scatter plot)** con múltiples variables visuales:

```
sns.relplot(
    x='Edad',
    y='Ingresos mensuales',
    hue='Nivel laboral',
    size='Educación',
    sizes=(40, 400),
    alpha=0.5,
    palette='muted',
    height=6,
    data=df_filtrado
)
```

Parámetros explicados:

- **x='Edad'**: El eje X representa la edad de las personas.
- **y='Ingresos mensuales'**: El eje Y representa los ingresos mensuales.
- **hue='Nivel laboral'**: Los puntos se colorean según el nivel laboral (por ejemplo: junior, senior, gerente...).

- **size='Educación'**: El tamaño de cada punto depende del nivel educativo (por ejemplo: secundaria, licenciatura, maestría...).
- **sizes=(40, 400)**: Define el rango de tamaños de los puntos (mínimo 40, máximo 400).
- **alpha=0.5**: Define la transparencia de los puntos (0 = totalmente transparente, 1 = opaco). Esto ayuda a ver mejor cuando hay muchos puntos superpuestos.
- **palette='muted'**: Usa una paleta de colores suaves para los distintos niveles laborales.
- **height=6**: Define la altura del gráfico en pulgadas.
- **data=df_filtrado**: Usa el DataFrame df_filtrado como fuente de datos.

Resultado. ¿Qué representa este gráfico?

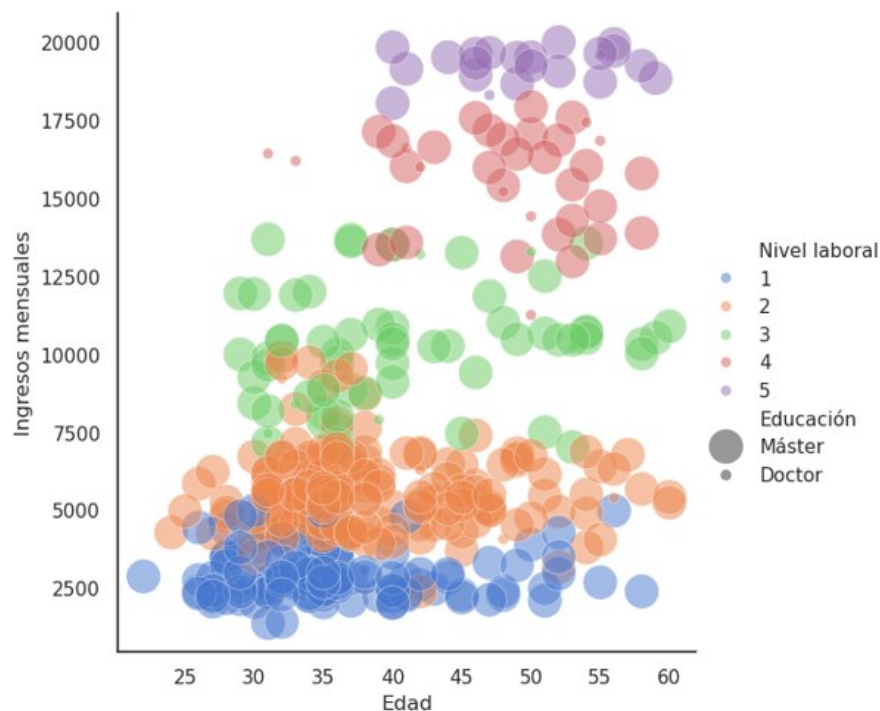


Ilustración 10

Es un gráfico de dispersión **multidimensional** que te permite ver cómo se relacionan la edad y los ingresos mensuales, diferenciando por nivel laboral (color) y nivel educativo (tamaño del punto). Es útil para detectar patrones, como si los ingresos aumentan con la edad o si ciertos niveles educativos están asociados a mayores ingresos.