NIVEL 4

GRÁFICOS PARA EXPLORAR UN DATAFRAME

Universidad de los Andes

N4

GRÁFICOS PARA EXPLORAR LOS DATOS DE UN DATAFRAME

- ✓ Para generar gráficos a partir de los datos almacenados en un DataFrame, se utiliza la librería matplotlib
- ✓ Matplotlib provee una interfaz de alto nivel para dibujar atractivos gráficos estadísticos

PREPARAR LA FIGURA

Lo primero que hay que hacer antes de empezar a hacer una gráfica es preparar una figura:

Terminal 4/A

```
In [49]: plt.figure()
Out[49]: <Figure size 432x288 with 0 Axes><Figure size 432x288 with 0 Axes>
```

plt es matplotlib.pyplot

PREPARAR LOS DATOS QUE SE VAN A GRAFICAR

Luego se extraen o filtran los datos con los que se construirán los gráficos:

Terminal 4/A

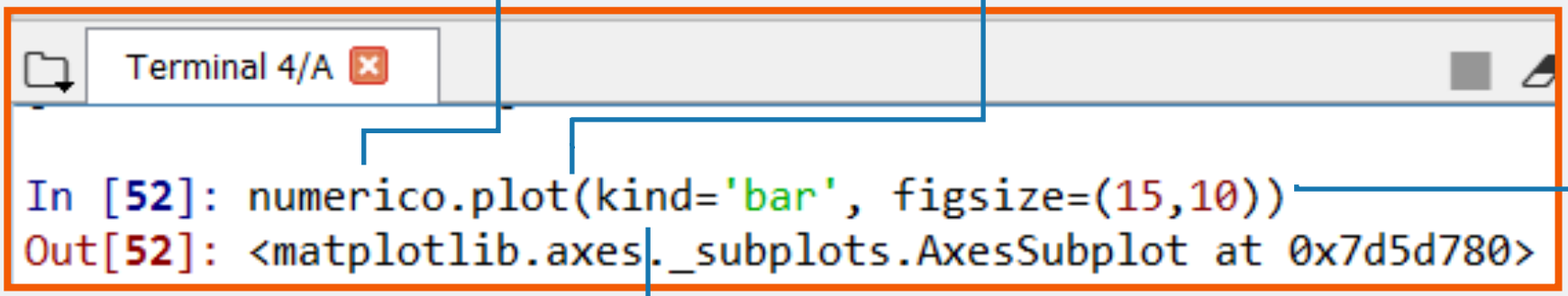
```
In [50]: numerico = peajes.iloc[0:5, 4:10]

In [51]: numerico
Out[51]:
   TAR_PLENA_I  TAR_PLENA_II      ...       TAR_PLENA_V  TAR_PLENA_VI
0        13800         17600      ...             42900         52600
1        10400         12600      ...             30700         38400
2        10400         12600      ...             30700         38400
3        10400         12600      ...             30700         38400
4        10400         12600      ...             30700         38400

[5 rows x 6 columns]
```

Aquí se extraen las columnas numéricas y las primeras 5 filas

A GRAFICAR!

Los datos que se van a utilizar son los que están en el DataFrame "numérico", que es sobre el que se está llamando el método plot

Se utiliza el método plot para generar la gráfica



El parámetro kind sirve para indicar el tipo de gráfica: en este caso, va a ser una gráfica de barras

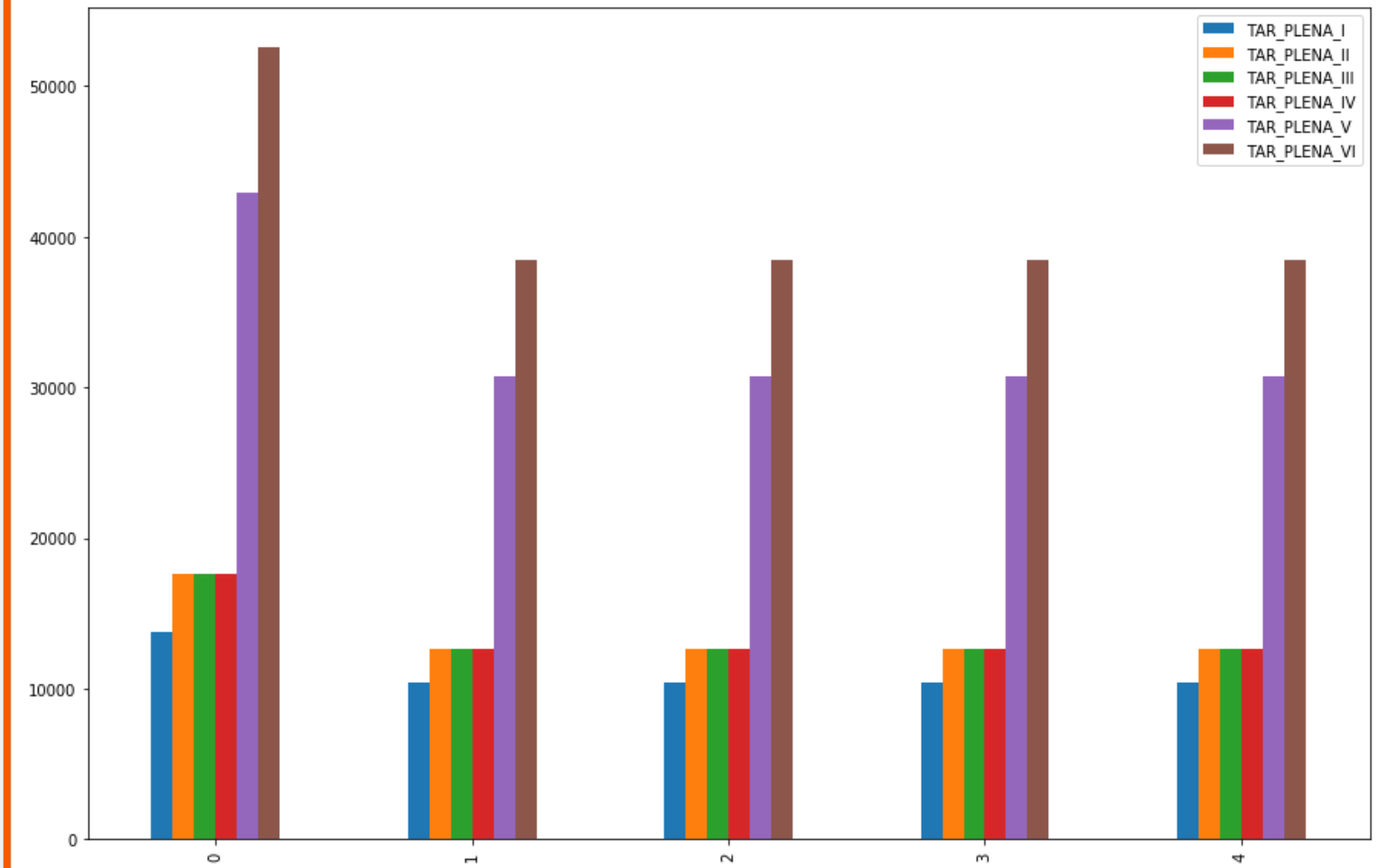
El parámetro figsize es una tupla e indica el tamaño de la figura (ancho, alto)

N4

GRÁFICA DE BARRAS

Terminal 4/A

```
In [52]: numerico.plot(kind='bar', figsize=(15,10))
Out[52]: <matplotlib.axes._subplots.AxesSubplot at 0x7d5d780>
```



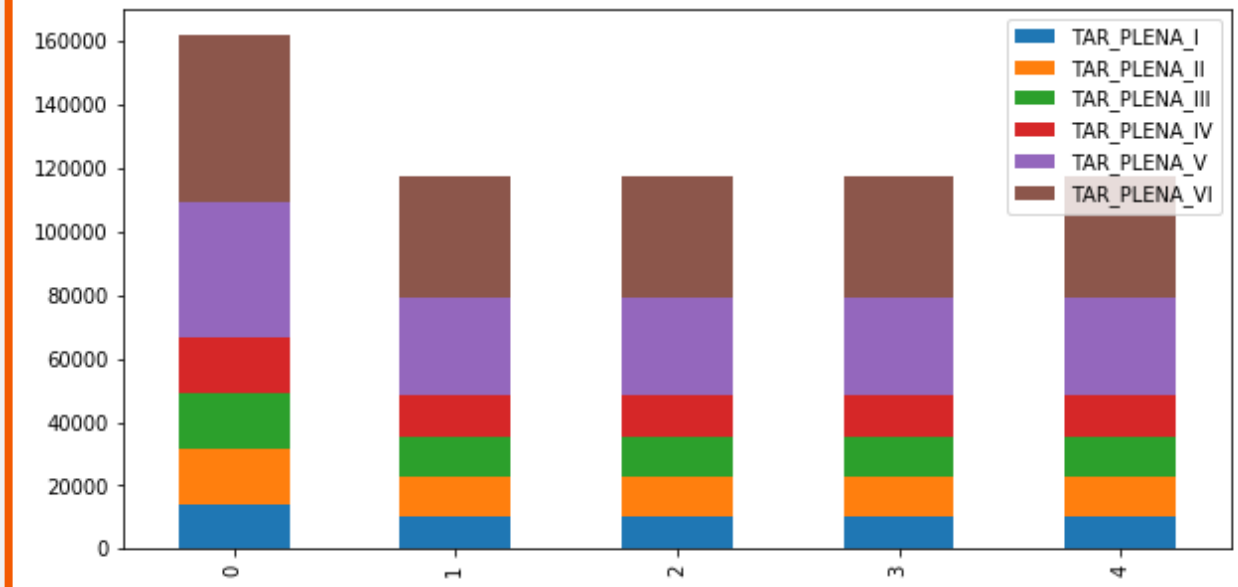
N4

OTRA GRÁFICA DE BARRAS

Esta segunda gráfica apila las barras usando el parámetro stacked que por defecto tiene valor False

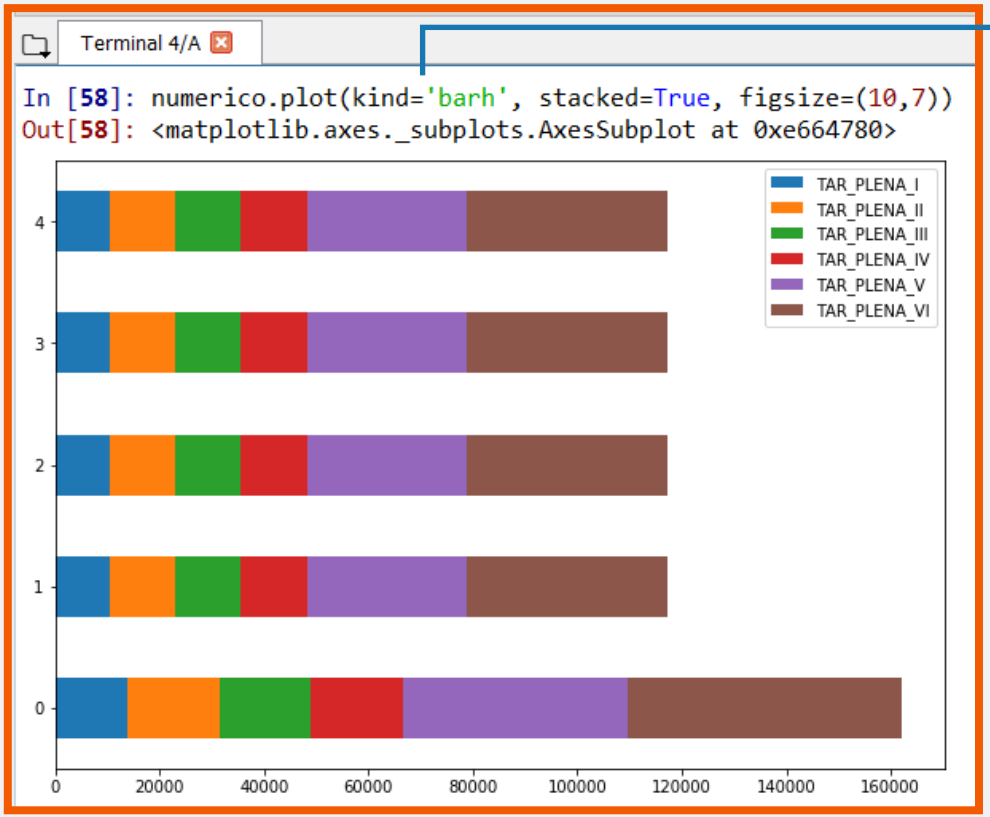
Terminal 4/A

```
In [54]: numerico.plot(kind='bar', stacked=True, figsize=(10,5))
Out[54]: <matplotlib.axes._subplots.AxesSubplot at 0xea516a0>
```



N4

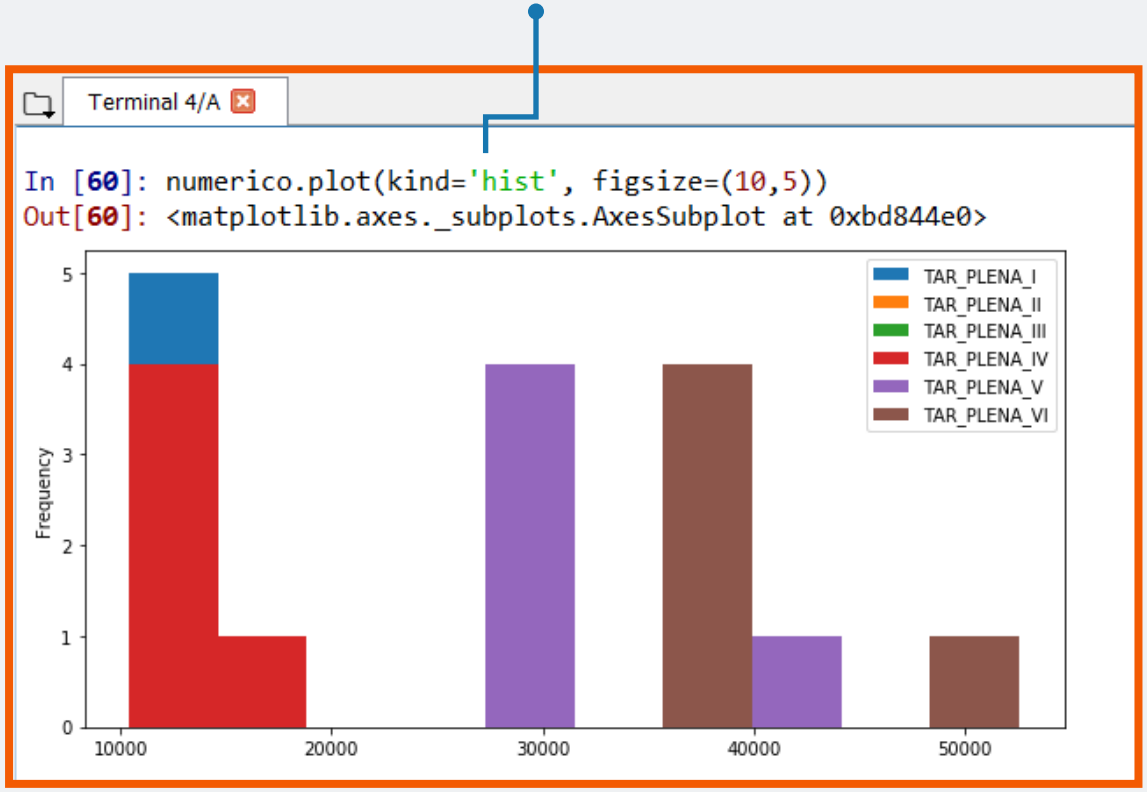
OTRA GRÁFICA DE BARRAS



En este tercer diagrama de barras, las barras son horizontales (kind es barh)

HISTOGRAMA

Este nuevo diagrama es un histograma, en el que se cuenta cuántas veces aparece cada valor dentro del conjunto de datos. En este caso estamos dejando a la librería que tome todas las decisiones sobre cómo organizar el histograma. En particular, no le dijimos cómo queríamos agrupar los valores y Pandas decidió agruparlos en 10 grupos

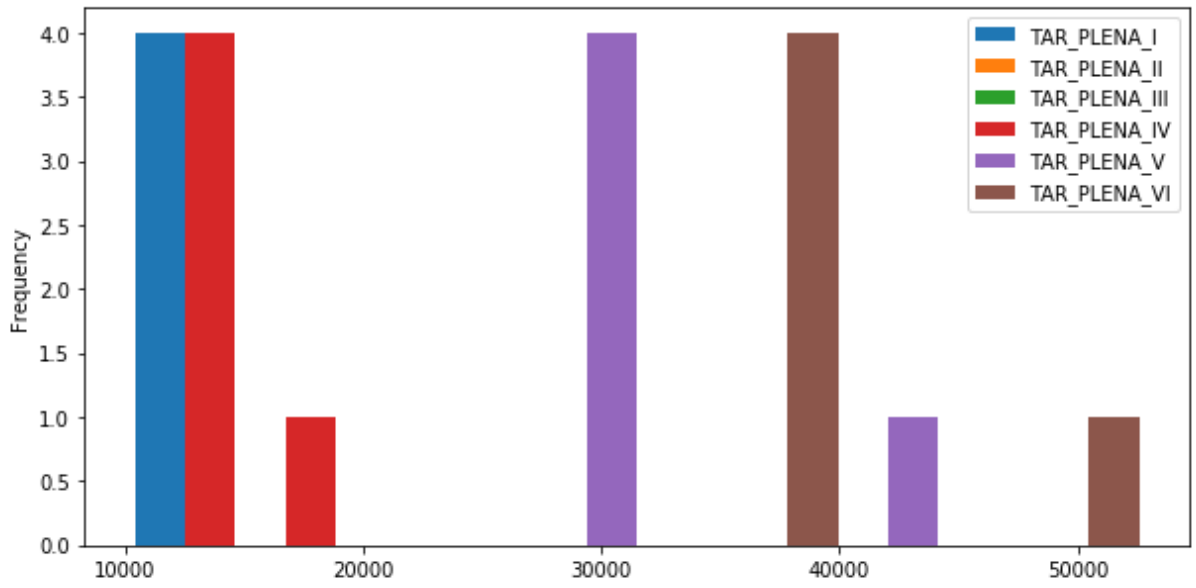


OTRO HISTOGRAMA

En este caso estamos indicando que queremos organizar los datos en 20 grupos, para tener más detalle

Terminal 4/A

```
In [61]: numerico.plot(kind='hist', bins=20, figsize=(10,5))
Out[61]: <matplotlib.axes._subplots.AxesSubplot at 0xbd792e8>
```

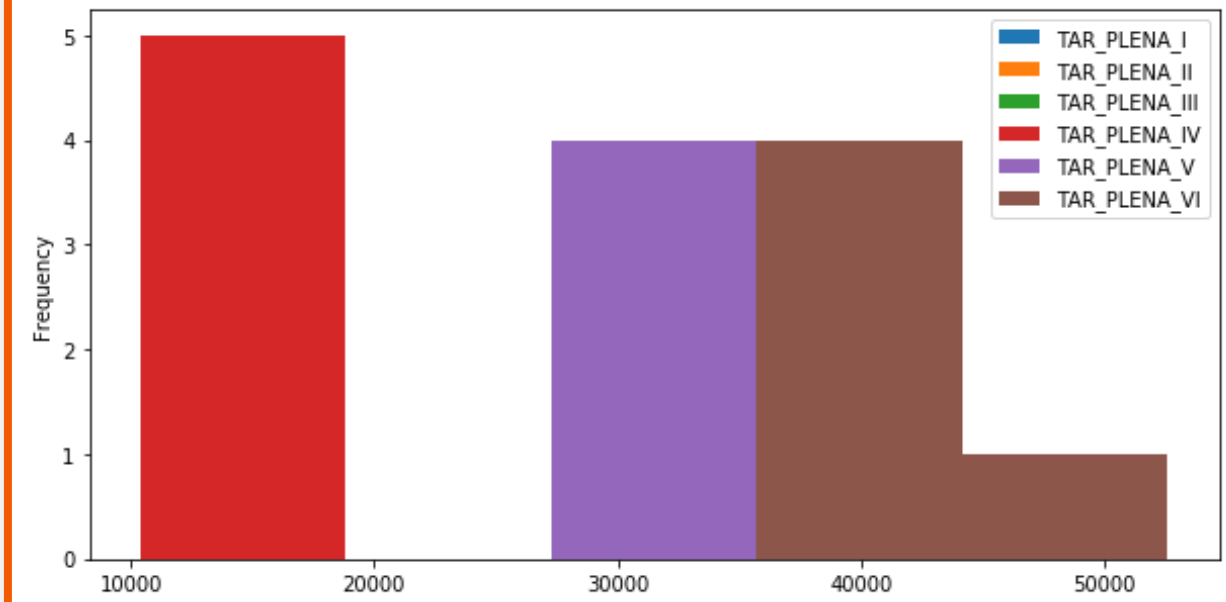


OTRO HISTOGRAMA

En este tercer caso indicamos que queremos organizar los datos en 5 grupos, así que tenemos mucho menos detalle

Terminal 4/A

```
In [63]: numerico.plot(kind='hist', bins=5, figsize=(10,5))
Out[63]: <matplotlib.axes._subplots.AxesSubplot at 0xd6226a0>
```



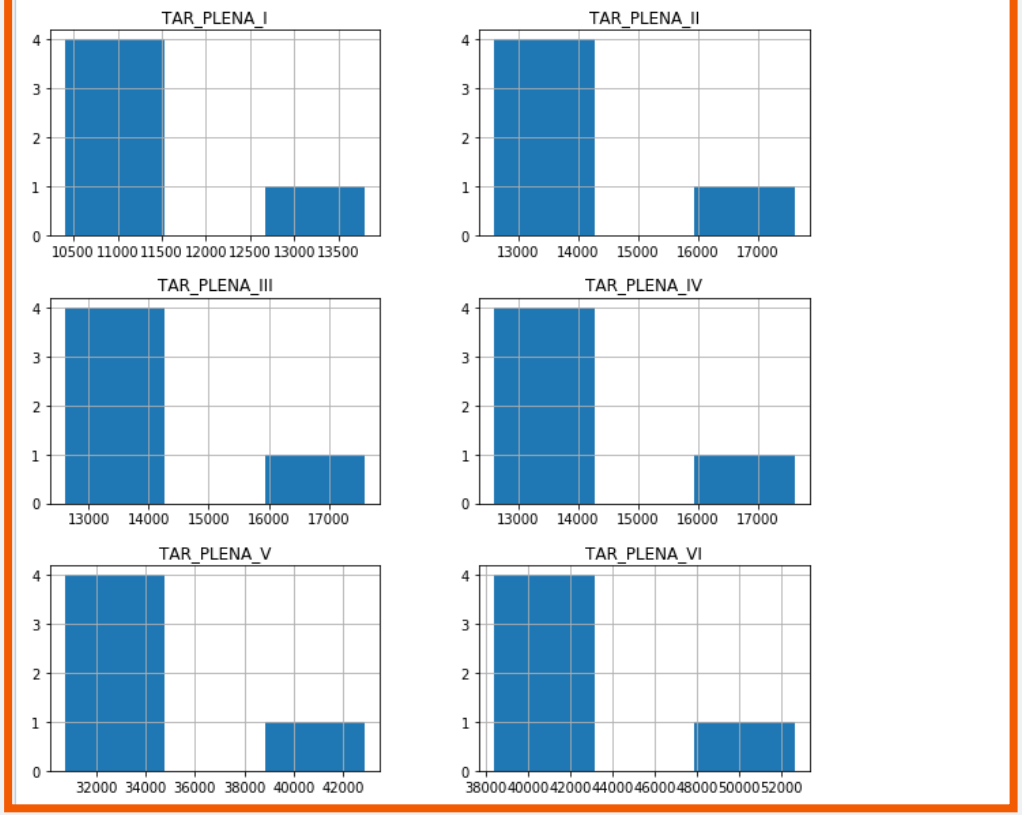
N4

OTRA FORMA DE CREAR HISTOGRAMAS

Esta es otra forma de crear histogramas: en este caso se crea un histograma para cada columna y además le indicamos que los valores se deben agrupar en 3 grupos para cada columna

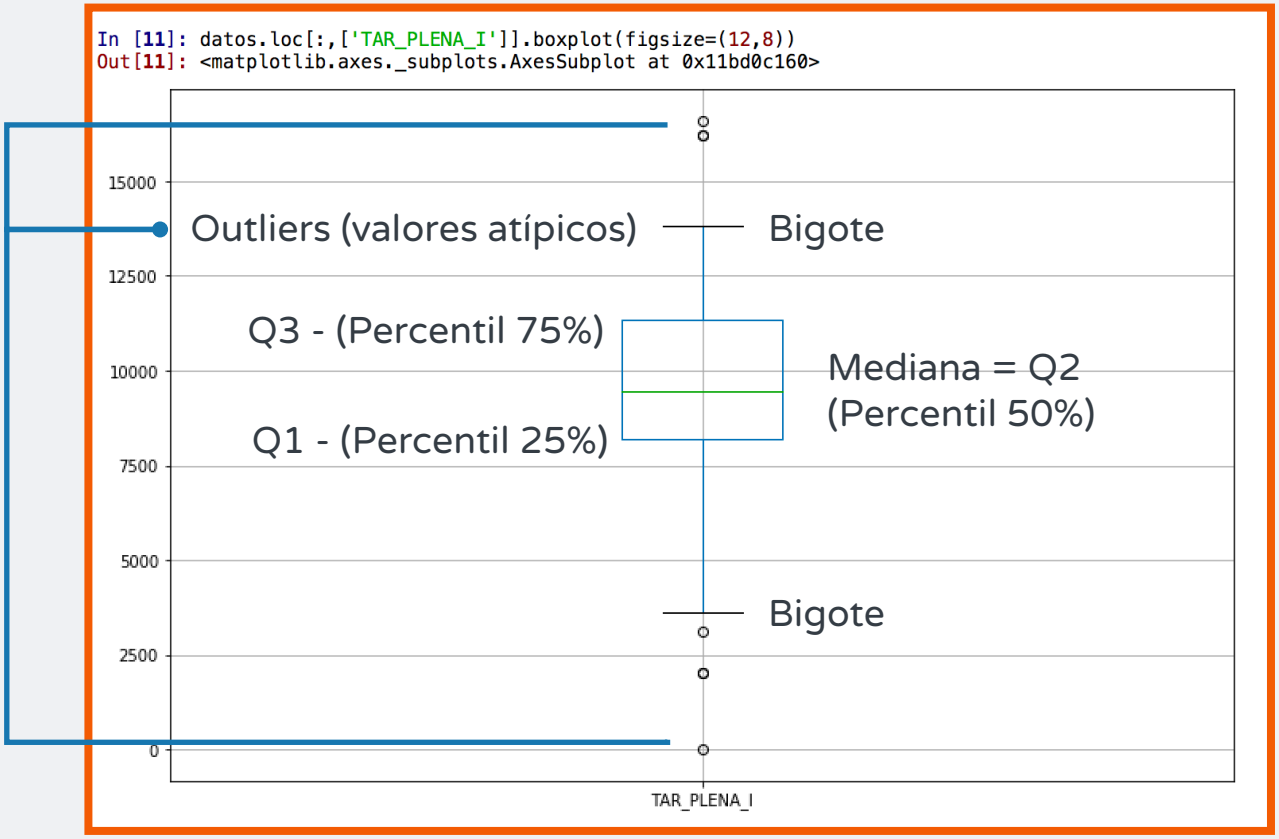
Terminal 1/A

```
In [7]: numerico.hist(figsize=(10,10), bins=3)
Out[7]:
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000000000C3555F8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000000000A28B748>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000000000A06C278>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000000000BD30630>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000000000B4167B8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000000000A390780>]],
      dtype=object)
```



EJEMPLO DE BOX - PLOT

Este gráfico es de tipo Box-Plot, también llamado diagrama de bigotes. Un Box-Plot es un tipo de gráfico que muestra mucha información de forma clara y resumida: muestra los valores máximos y mínimos, la desviación estándar (la caja) y los 4 cuartiles (los bigotes)

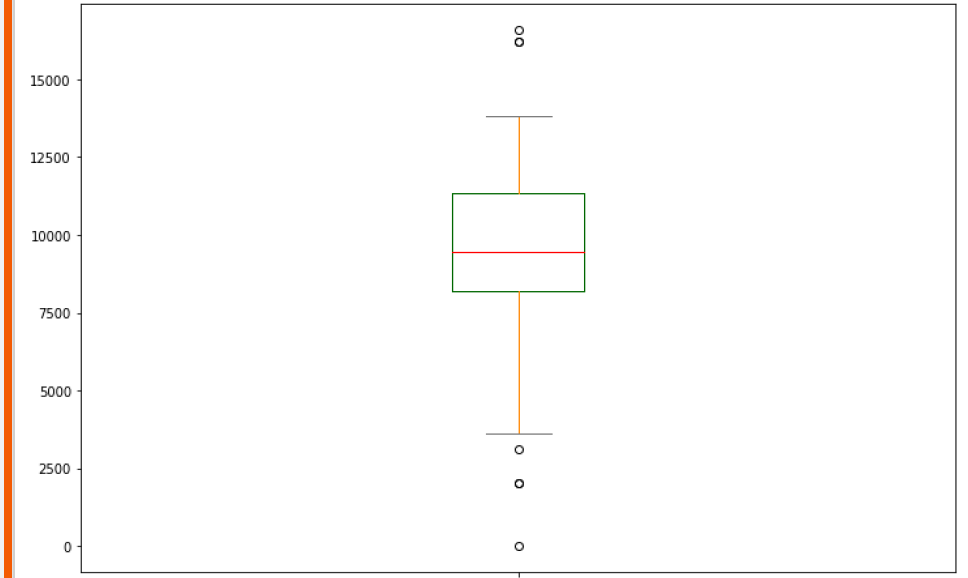


N4

PARA CAMBIAR LOS COLORES DEL BOX - PLOT

Construimos un diccionario con los valores boxes, whiskers, medians, caps y luego se lo pasamos al método box en el parámetro color

```
In [59]: color = {'boxes': 'DarkGreen', 'whiskers': 'DarkOrange', 'medians':'Red', 'caps': 'Gray'}

In [60]: datos.iloc[:,[4]].plot.box(color=color, figsize=(12,8))
Out[60]: <matplotlib.axes._subplots.AxesSubplot at 0x11c85be48>
```



Note que estamos usando la función **iloc** y debemos indicar las columnas que nos interesan usando una lista con sus posiciones; y **figsize** fija el tamaño de la figura

N4

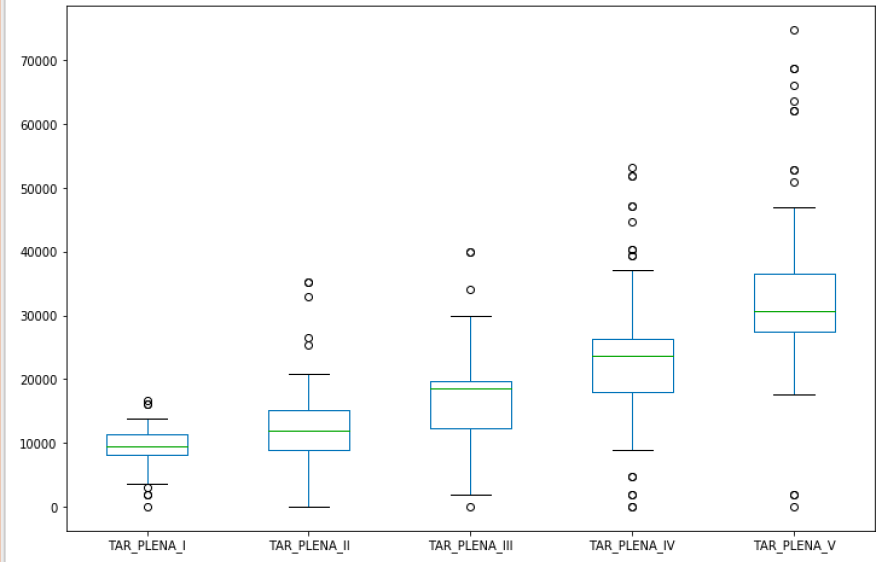
EJEMPLO DE BOX – PLOT CON VARIAS COLUMNAS

La función loc filtra la información del dataframe datos:

- Se indican los registros que se van a utilizar: “ : ” - todas las filas de **datos**
- Se indican las columnas con una lista de cadenas de caracteres

Resultado: un dataframe con la misma cantidad de registros que **datos**, pero solo tienen las columnas correspondientes a las tarifas I a V

```
In [5]: datos.loc[:,["TAR_PLENA_I","TAR_PLENA_II", "TAR_PLENA_III", "TAR_PLENA_IV", "TAR_PLENA_V"]].plot.box(figsize=(12,8))
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x11bf860b8>
```



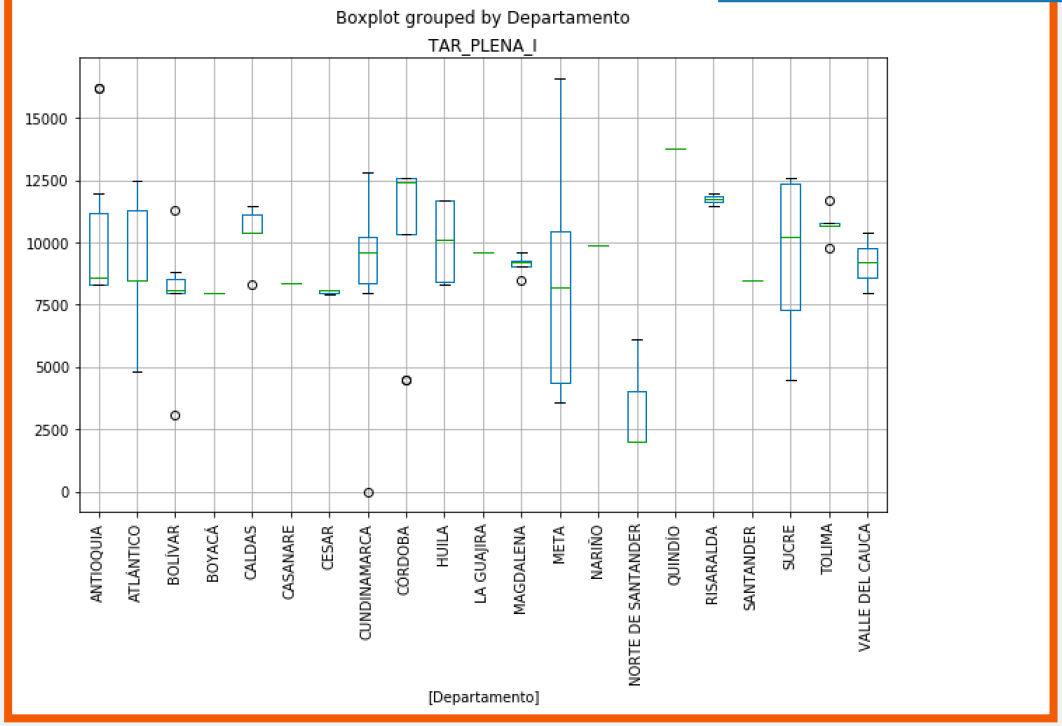
La función **plot.box** construye un diagrama de bigotes por cada columna numérica en el dataframe

N4

EJEMPLO DE BOX – PLOT COMPARACIÓN

Queremos ver cómo varían los precios de los peajes entre departamentos limitándonos a una sola categoría (categpría I). Usamos la función **boxplot** usa con el parámetro **by** para indicar que queremos tener una caja por cada departamento (por eso Departamento hace parte de las columnas del dataframe) y **rot** se usa para rotar las etiquetas 90 grados

```
In [63]: datos.loc[:,['Departamento','TAR_PLENA_I']].boxplot(by="Departamento",rot=90,figsize=(10,6))
Out[63]: <matplotlib.axes._subplots.AxesSubplot at 0x11c166ef0>
```



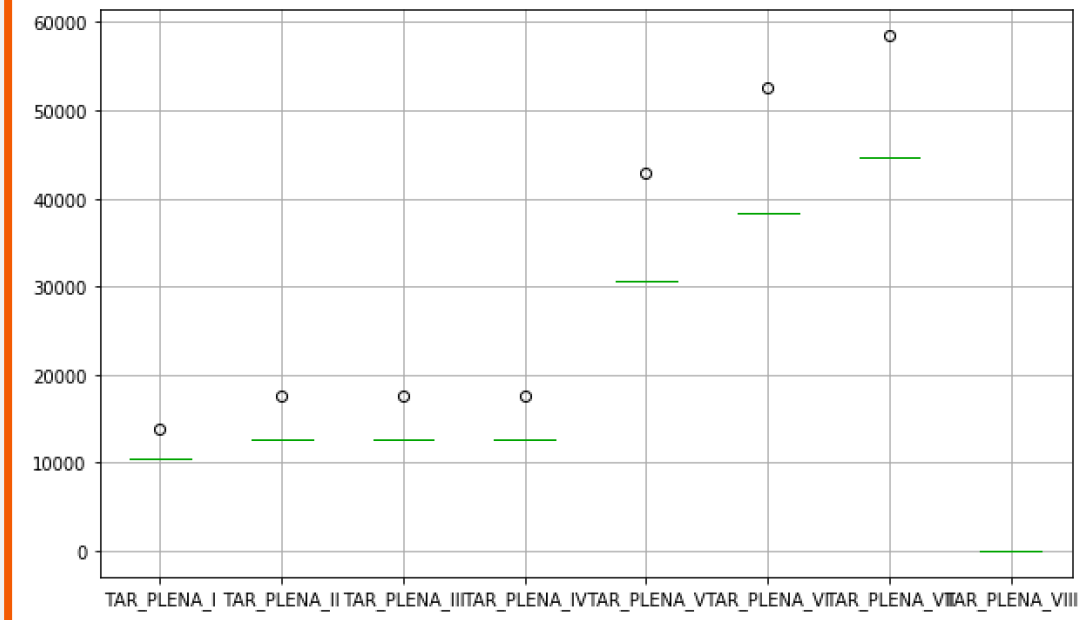
Universidad de los Andes

N4

EJEMPLO DE BOX – PLOT CON POCA INFORMACIÓN

- ✓ En el ejemplo anterior pudo haber notado que si generamos un box-plot sobre el subconjunto con poca información (Boyacá, Casanare, etc) podemos ver que hay muy poca información para que se pueda construir una gráfica
- ✓ Veamos este ejemplo con un dataframe con solo 5 registros

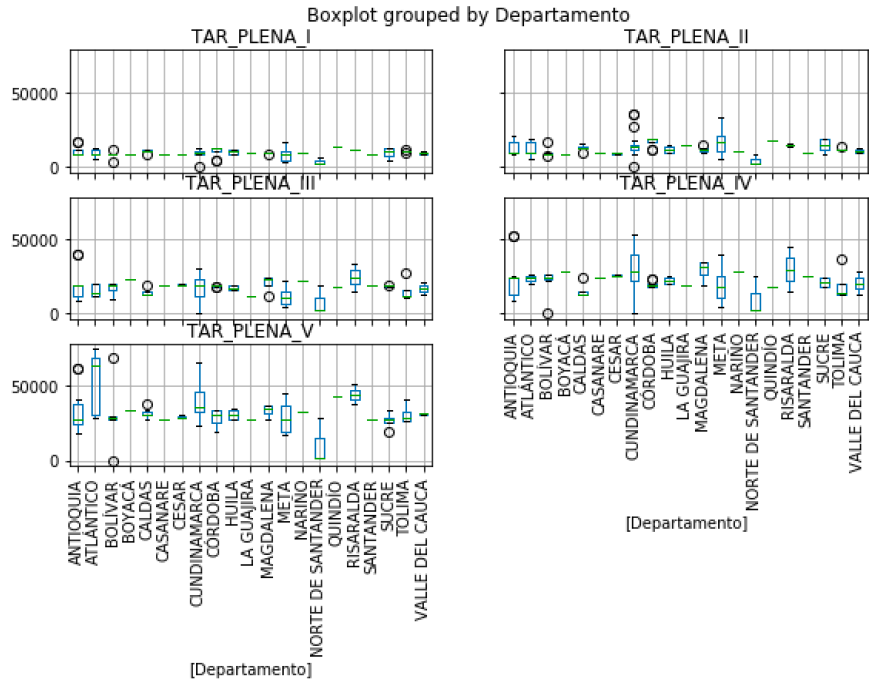
```
In [72]: solo_cinco = datos.head()

In [73]: solo_cinco.boxplot(figsize=(10,6))
Out[73]: <matplotlib.axes._subplots.AxesSubplot at 0x11d7b4630>
```



N4

EJEMPLO DE BOX – PLOT COMPARACIÓN USANDO GRUPOS DE GRÁFICAS

```
In [76]: datos.loc[:,["Departamento", "TAR_PLENA_I","TAR_PLENA_II", "TAR_PLENA_III", "TAR_PLENA_IV",
"TAR_PLENA_V"]].boxplot(by="Departamento",rot=90,figsize=(9,5))
Out[76]:
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x11e83a240>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x11f0f4c88>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x11f177e10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x11f126128>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x11f19c400>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x11f1c66d8>]],
      dtype=object)
```

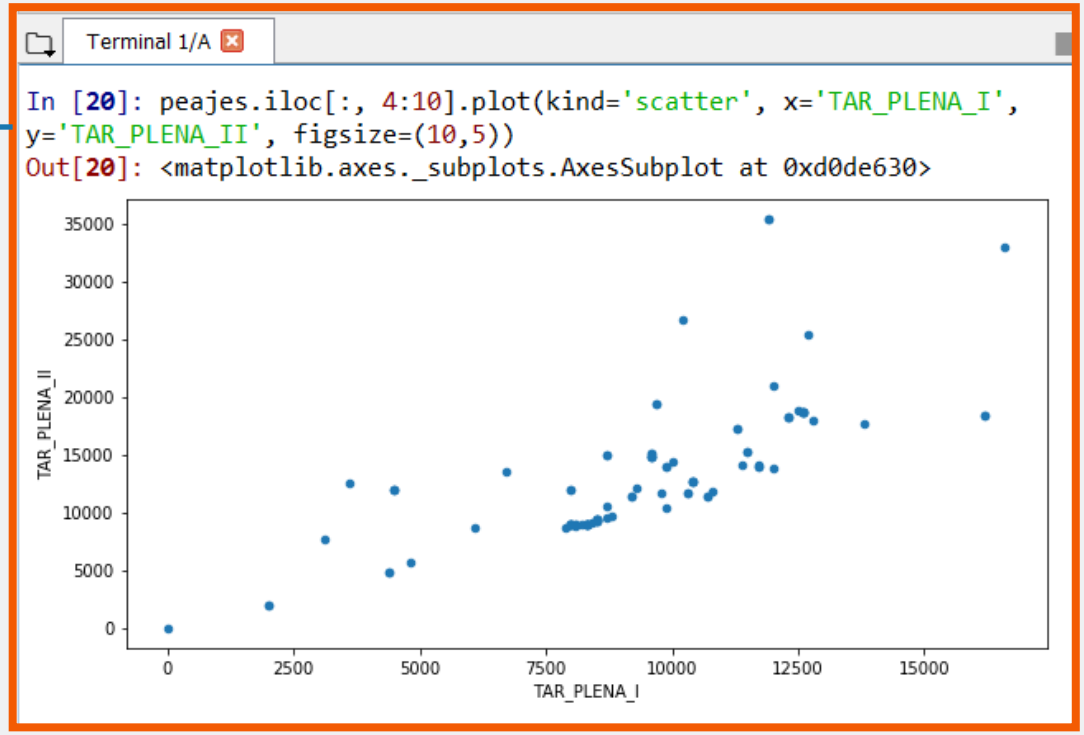


Podemos ver que pandas generó las 5 gráficas (una por columna) pero se encargó de ajustarlas para que tuvieran la misma escala y estuvieran alineadas, facilitando así su comparación

EJEMPLO DE SCATTER

Un diagrama de tipo Scatter compara el comportamiento de dos variables, marcando en un diagrama puntos para cada pareja de valores de estas variables. Hay que indicar cuáles son las columnas que nos interesan (es decir, las 2 variables)

Aquí decimos que en el eje **x** vamos a usar la columna **TAR_PLENA_I** y para el eje **y** vamos a usar **TAR_PLENA_II**

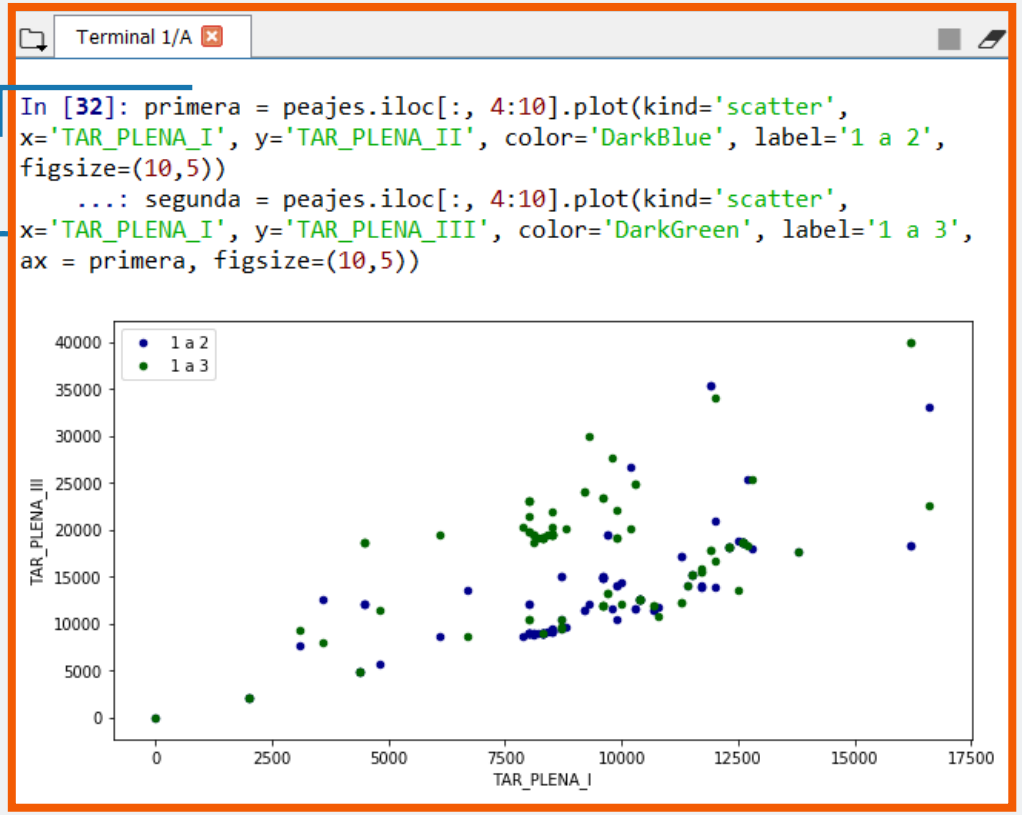


N4

EJEMPLO DE SCATTER CON 2 CONJUNTOS DE PAREJAS

Las siguientes 2 instrucciones, si se ejecutan en el intérprete de Spyder, deben ejecutarse JUNTAS!

- ✓ La primera línea crea una gráfica igual a la del ejemplo anterior, usando el color DarkBlue, y la almacena en la variable 'primera'
- ✓ La segunda línea crea una gráfica comparando TAR_PLENA_I con TAR_PLENA_III y usando el color DarkGreen. Además incluye la primera gráfica usando el parámetro ax



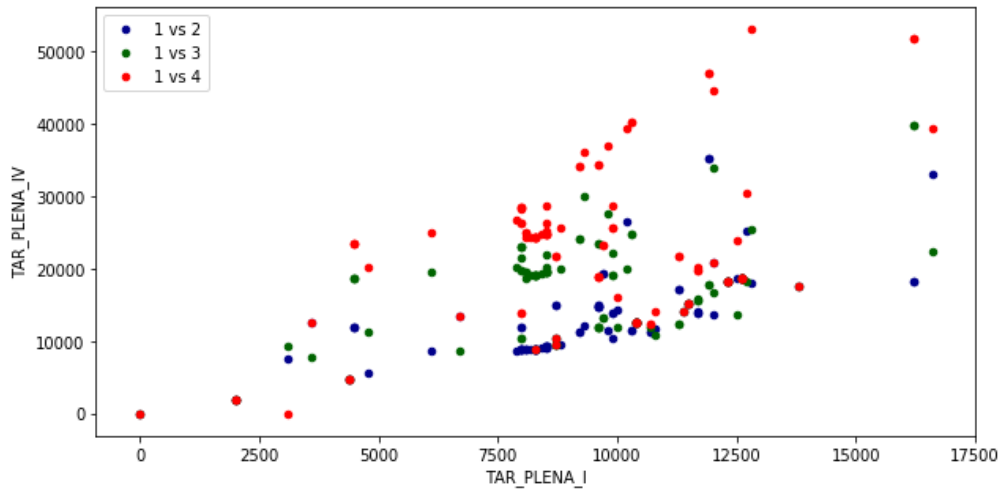
N4

EJEMPLO DE SCATTER CON 3 CONJUNTOS DE PAREJAS

Estas tres instrucciones hacen algo similar al ejemplo anterior, combinando la primera gráfica con la segunda, y la gráfica resultante con una tercera gráfica

Terminal 1/A

```
In [34]: primera = peajes.iloc[:, 4:10].plot(kind='scatter',
x='TAR_PLENA_I', y='TAR_PLENA_II', color='DarkBlue', label='1 vs 2',
figsize=(10,5))
    ...: segunda = peajes.iloc[:, 4:10].plot(kind='scatter',
x='TAR_PLENA_I', y='TAR_PLENA_III', color='DarkGreen', label='1 vs 3',
ax = primera, figsize=(10,5))
    ...: tercera = peajes.iloc[:, 4:10].plot(kind='scatter',
x='TAR_PLENA_I', y='TAR_PLENA_IV', color='Red', label='1 vs 4', ax =
segunda, figsize=(10,5))
```



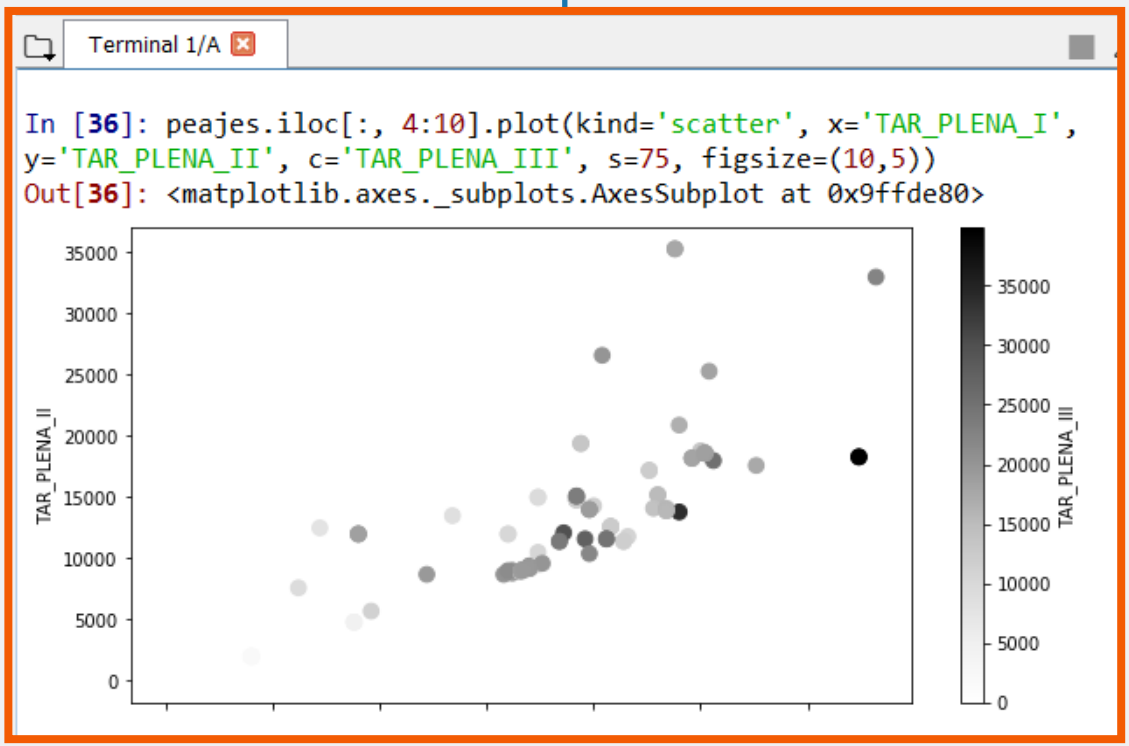
N4

EJEMPLO DE OTRO TIPO DE SCATTER CON 3 VARIABLES

Esta gráfica combina tres variables:

- ✓ Para el eje x, se usa TAR_PLENA_I
- ✓ Para ej eje y, se usa TAR_PLENA_II
- ✓ Para el color de los puntos (c), se usa TAR_PLENA_III

El parámetro s indica el tamaño de los puntos y en este caso le pusimos el valor 75 para que se puedan ver con facilidad



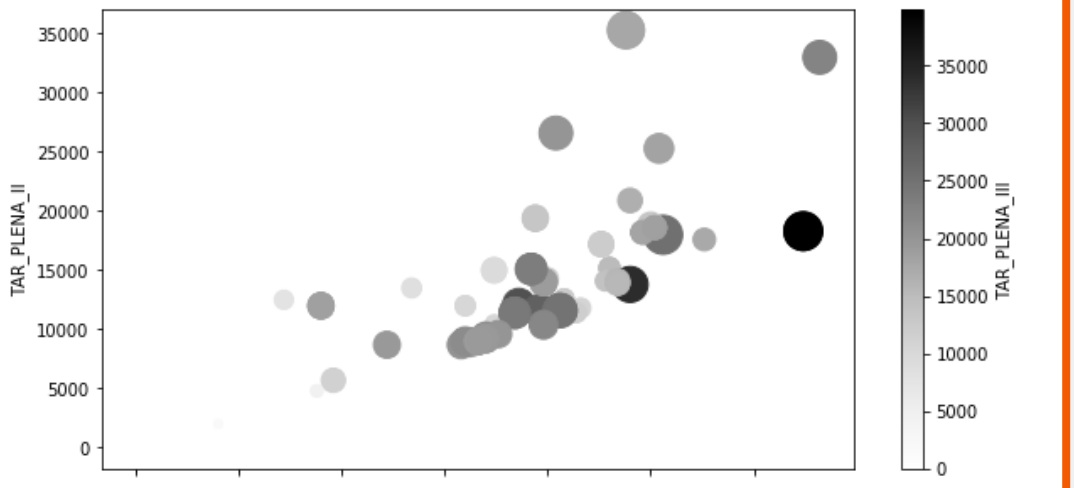
N4

EJEMPLO DE OTRO TIPO DE SCATTER CON 4 VARIABLES

- ✓ En esta gráfica se agregó una nueva variable que se está usando para calcular el tamaño de cada punto
- ✓ De esta forma entre mayor sea el valor de TAR_PLENA_IV, mayor será el tamaño de cada punto

Terminal 1/A

```
In [39]: peajes.iloc[:, 4:10].plot(kind='scatter', x='TAR_PLENA_I',
y='TAR_PLENA_II', c='TAR_PLENA_III', s=peajes['TAR_PLENA_IV']/100,
figsize=(10,5))
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x986c7f0>
```



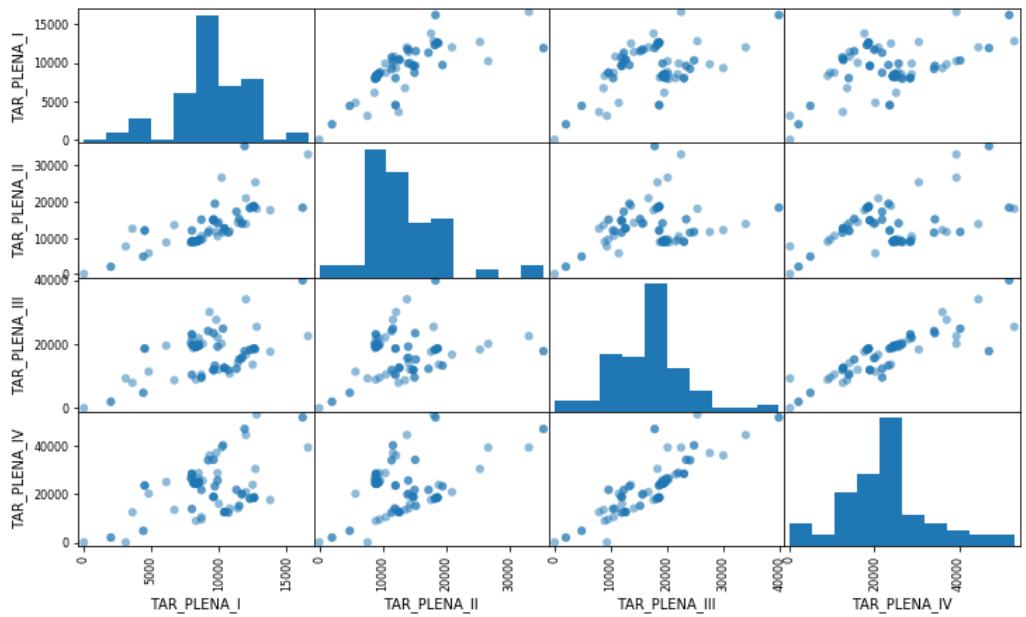
N4

EJEMPLO DE UNA MATRIZ DE GRÁFICAS DE DISPERSIÓN

Una matriz de gráficas de dispersión permite comparar varias variables entre sí, de forma simultanea (en un solo gráfico)

Terminal 1/A

```
In [45]: scatter_matrix(peajes.iloc[:, 4:8], figsize=(12,7), diagonal='hist', s=150)
Out[45]:
```



En este caso estamos usando sólo 4 columnas (variables) y con el parámetro diagonal estamos indicando qué gráfica usar cuando "comparamos" a una columna con ella misma