Visualizando errores y dispersión

En muchas ocasiones necesitaremos mostrar el error cometido respecto a una estimación o predicción o visualizar la dispersión de los valores de una variable. Para ello podemos utilizar barras de error en el primer caso o diagramas de cajas en el segundo caso.

Configurando el entorno.

Lo primero es configurar el entorno de ejecución. Véase el cuaderno "Primeros pasos con Matplotlib" para más detalles.

In [7]:
```
%matplotlib notebook
import matplotlib as mpl
import matplotlib.pyplot as plt
print('Matplotlib version: {}'.format(mpl.__version__))
import numpy as np
np.set_printoptions(floatmode='fixed', precision=3)
import pandas as pd
from io import StringIO
```

```
Matplotlib version: 3.5.2
```

Añadiendo líneas de error a un gráfico de líneas o puntos.

Para representar un gráfico de puntos que muestre barras de error asociadas a cada punto se utiliza la función `Axes.errorbar()`. Esta función añade a las opciones ya vistas de `Axes.plot()`, para dibujar un relación x|y, las opciones para dibujar las barras de error.

Estas barras se pueden dibujar de distintas formas según los valores de los parámetros `xerr`, para mostrar errores en el eje x y `yerr` para mostrar errores en el eje y.

Podemos mostrar un mismo error para todos los puntos, un error simétrico para cada punto, o un error por defecto y por exceso para cada punto.

Además también podemos especificar un formato línea para unir los puntos (como en plot()) y otro para las barras de error, o el mismo formato para todo.

Ejercicio 01: Se desea cargar desde un fichero CSV los datos de cotización de una empresa en un objeto Dataframe.

La salida esperada debería ser algo parecido a lo siguiente:

Fecha	Último	Apertura	Máximo	Mínimo	Vol.	% var.
2021-04-01	3.782	3.815	3.853	3.782	15,16M	-0,92%
2021-04-06	3.780	3.818	3.843	3.776	32,00M	-0,05%
2021-04-07	3.806	3.805	3.878	3.803	20,81M	0,69%

```
2021-04-08   3.818     3.828   3.841   3.773  12,96M   0,32%
2021-04-09   3.800     3.829   3.840   3.783   9,22M  -0,47%
2021-04-12   3.810     3.793   3.835   3.763  21,35M   0,26%
2021-04-13   3.737     3.800   3.814   3.711  16,68M  -1,92%
2021-04-14   3.736     3.750   3.750   3.682  16,08M  -0,03%
2021-04-15   3.694     3.765   3.786   3.692  12,89M  -1,11%
2021-04-16   3.752     3.700   3.752   3.680  14,80M   1,56%
2021-04-19   3.763     3.760   3.782   3.724  12,87M   0,29%
2021-04-20   3.685     3.761   3.761   3.667  15,49M  -2,07%
2021-04-21   3.703     3.695   3.736   3.657  10,18M   0,49%
2021-04-22   3.694     3.700   3.714   3.657  11,81M  -0,26%
2021-04-23   3.668     3.690   3.720   3.650  42,56M  -0,69%
2021-04-26   3.772     3.692   3.778   3.663  14,82M   2,82%
2021-04-27   3.806     3.774   3.826   3.760  14,84M   0,90%
2021-04-28   3.843     3.832   3.877   3.823   9,54M   0,99%
2021-04-29   3.845     3.863   3.874   3.829  11,83M   0,05%
2021-04-30   3.853     3.850   3.908   3.846  15,09M   0,22%
```

In [8]:

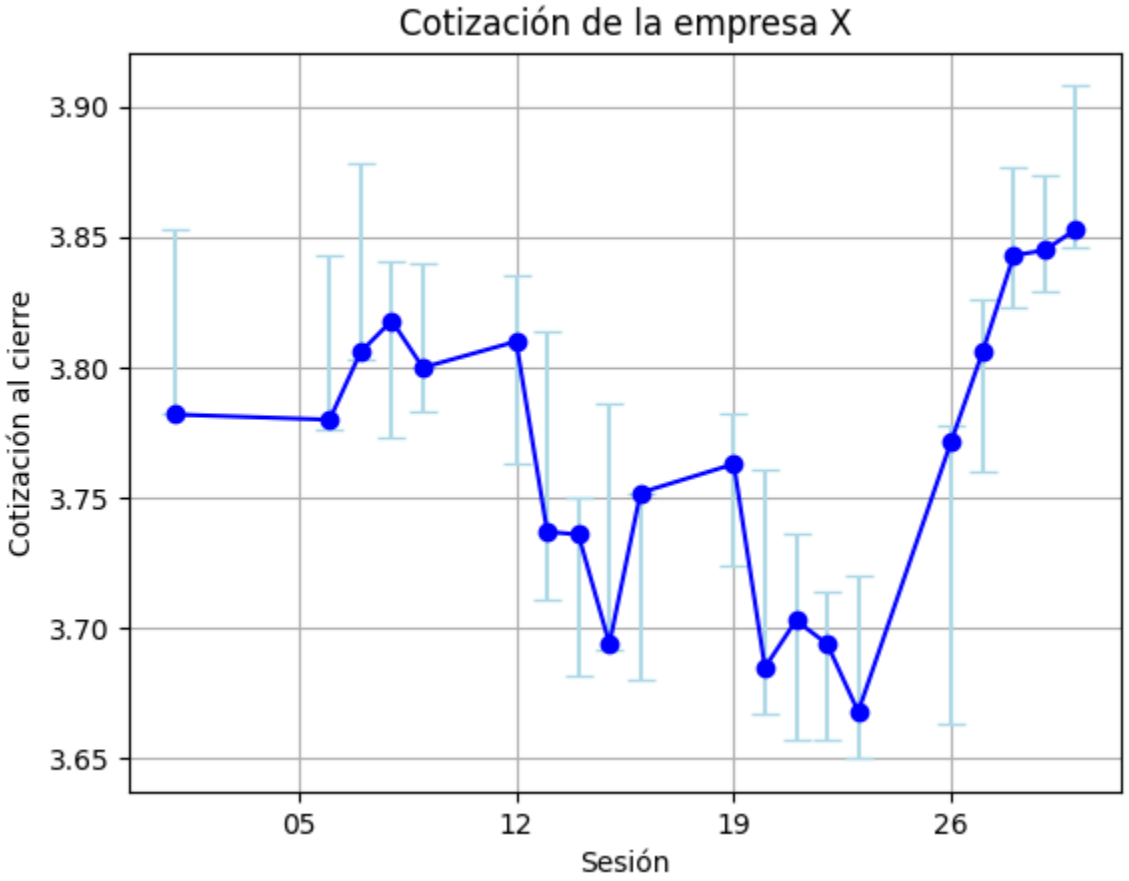
```
#Datos obtenidos desde es.investing.com
file=StringIO(
'''
Fecha   Último  Apertura        Máximo  Mínimo  Vol.    % var.
01.04.2021      3,782   3,815   3,853   3,782   15,16M  -0,92%
06.04.2021      3,780   3,818   3,843   3,776   32,00M  -0,05%
07.04.2021      3,806   3,805   3,878   3,803   20,81M  0,69%
08.04.2021      3,818   3,828   3,841   3,773   12,96M  0,32%
09.04.2021      3,800   3,829   3,840   3,783   9,22M   -0,47%
12.04.2021      3,810   3,793   3,835   3,763   21,35M  0,26%
13.04.2021      3,737   3,800   3,814   3,711   16,68M  -1,92%
14.04.2021      3,736   3,750   3,750   3,682   16,08M  -0,03%
15.04.2021      3,694   3,765   3,786   3,692   12,89M  -1,11%
16.04.2021      3,752   3,700   3,752   3,680   14,80M  1,56%
19.04.2021      3,763   3,760   3,782   3,724   12,87M  0,29%
20.04.2021      3,685   3,761   3,761   3,667   15,49M  -2,07%
21.04.2021      3,703   3,695   3,736   3,657   10,18M  0,49%
22.04.2021      3,694   3,700   3,714   3,657   11,81M  -0,26%
23.04.2021      3,668   3,690   3,720   3,650   42,56M  -0,69%
26.04.2021      3,772   3,692   3,778   3,663   14,82M  2,82%
27.04.2021      3,806   3,774   3,826   3,760   14,84M  0,90%
28.04.2021      3,843   3,832   3,877   3,823   9,54M   0,99%
29.04.2021      3,845   3,863   3,874   3,829   11,83M  0,05%
30.04.2021      3,853   3,850   3,908   3,846   15,09M  0,22%
'''
)
df = pd.DataFrame()

#Pon tu código aquí.
#Sugerencia: carga los datos con read_csv.
#Observa que el separador de columnas en un caracter especial 'tabulador'.
#La primera columna "Fecha" debe ser el índice de filas y contine fechas.
#Debes indicar que el format de fecha tiene el día al principio.
#Además los valores flotantes usan ',' para separar la parte decimal.

#

print(df)
```

```
Empty DataFrame
Columns: []
Index: []
```

Ejercicio 02: Usando el dataframe cargado en el ejemplo anterior queremos representaremos con barras de error los valores mínimos y máximos de cotización de la sesión (técnicamente estos no son errores,

pero podemos visualizarlos como tales).

Intenta ajustar los parámetros para que el resultado sea lo más parecido posible a la siguiente figura.



In [9]:
```
import matplotlib.dates as mdates #necesario para gestionar ejes con datos temporales.
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters() #Requerido por matplotlib si usamos objetos pandas con

fig = plt.figure()
ax = plt.axes()

#Pon tu código aquí.
#Sugerencia: especifica el título, etiquetas de ejes y activa la rejilla.
#en el objeto Axes.

#

#Estas sentencias configuran el eje X para mostrar fechas.
#Esto se estudiará más adelante.
ax.xaxis.set_major_locator(mdates.WeekdayLocator(byweekday=mdates.MO))#Un tick cada lune
dayFmt = mdates.DateFormatter("%d")#Inidicamos un formato de fecha que imprime sólo el d
ax.xaxis.set_major_formatter(dayFmt)#Indicamos que los tick usen el formato de fecha dad

#Pon tu código aquí.
#Sugerencia: utiliza el comando errorbar().
#Utiliza el parámetro yerr para indicar en un array (2,N) el error por defeto,
#y el error por exceso.
#Recuerda: El índice del dataframe es el eje X.
#   Las columnas, Último, (Último-Mínimo) y (Máximo-Último) dos dan los valores
#   a mostrar.

#
```

Gráfico de caja.

Un gráfico "caja" `Axes.boxplot()` (también conocido como gráfico de bigotes "*whisker plot*") permite condensar mucha información sobre la distribución de valores de una variable.

En un gráfico de caja se representan cinco valores de una distribución (ver Figura) y además permite mostrar valores raros (*outlier*). Para una descripción más completa se recomienda leer esta referencia.

```
      Q1-1.5IQR   Q1   median  Q3   Q3+1.5IQR
                   |-----:-----|
  o      |--------|     :     |--------|    o  o
                   |-----:-----|
flier              <----------->            fliers
                        IQR
```

La caja se dibuja desde el primer cuartil al tercero. Esta información permite analizar rápidamente la distribución de valores de una variable y también comparar entre las distribuciones de varias variables.

Ejercicio 03: Cargar un fichero donde se almacena la exactitud (*Accuracy*) conseguida por varios clasificadores cuando se han aplicado en varios conjuntos de test.

La salida generada debería ser algo parecido a los siguiente:

```
                C1         C2          C3
Test
Test 0  41.195124  58.179442  100.000000
Test 1  40.084025  45.197243   74.461500
```

```
   Test 2  53.916186  65.116233   24.809018
   Test 3  52.708748  59.078461   94.342665
   Test 4  50.342881  56.653179   51.577038
```

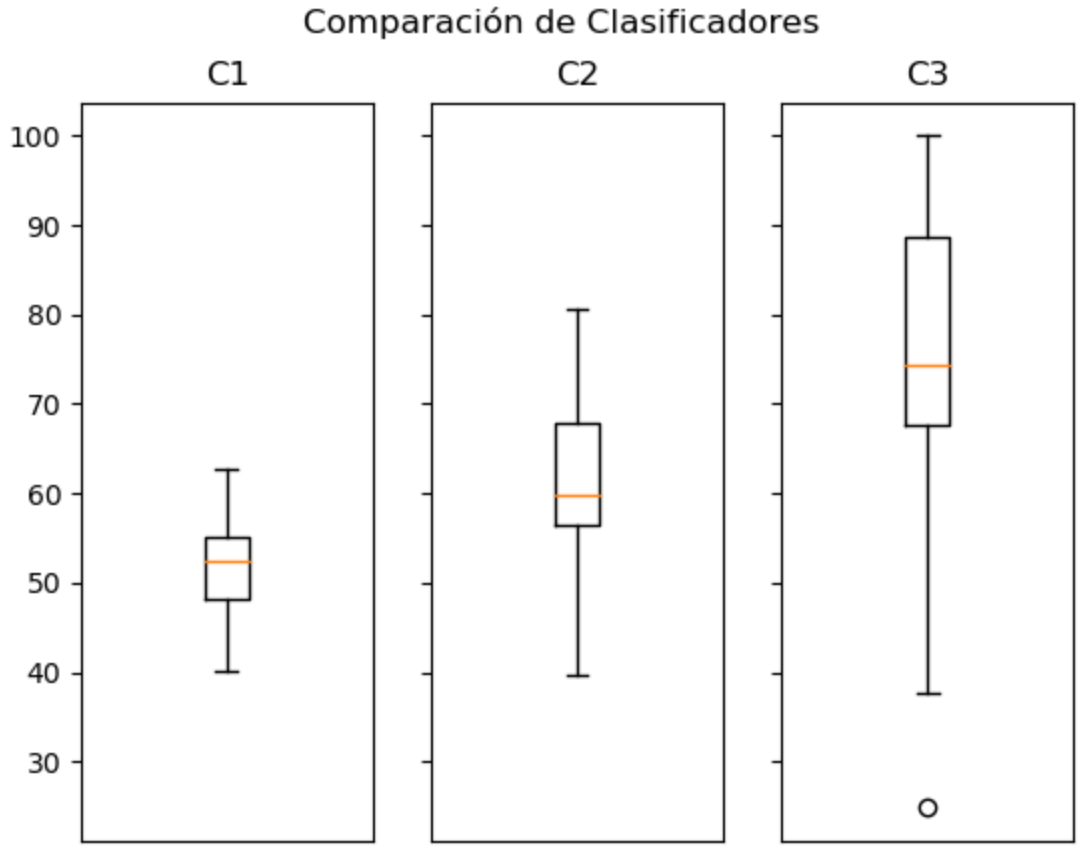
In [10]:

```
file = StringIO(
'''Test:C1:C2:C3
Test 0:41,19512356955531:58,179442494402025:100,0
Test 1:40,08402469006468:45,19724281341427:74,4615002447282
Test 2:53,91618612768901:65,11623326085783:24,80901846223321
Test 3:52,708747647224136:59,07846091107256:94,34266485081307
Test 4:50,342880673053536:56,65317858383716:51,57703793403086
Test 5:51,959935977730815:80,68135528917625:87,4590004894867
Test 6:52,24205720830521:42,332475597667695:37,64575931058347
Test 7:47,427919175371:61,511637301107825:66,96385536869067
Test 8:49,617861440681594:70,27561007879322:74,35708060746806
Test 9:49,11398357057106:65,7168523881314:62,25817400384894
Test 10:44,14103980544816:49,327244083211916:72,36207265600606
Test 11:46,68275259969946:68,25807361839361:60,1318751882874
Test 12:47,67815785141936:39,6501555822916:69,63002646083551
Test 13:56,76939062603222:57,33328425189499:100,0
Test 14:53,44828205382989:71,36776106665252:85,63759728377201
Test 15:57,28406855178828:66,512889462018:100,0
Test 16:54,04194461978242:66,10210103675635:59,39404418511116
Test 17:55,22557864922575:53,10807747676824:76,60699134349763
Test 18:56,66875103159891:43,01689392094074:100,0
Test 19:54,20354953792935:78,69260105551776:73,1564726825055
Test 20:54,76420040067305:57,835047681417954:83,08581312871534
Test 21:62,82683408828065:54,687662540642286:100,0
Test 22:52,637469757179446:78,24335907879723:73,67410490403417
Test 23:48,25376630323033:76,03333377247561:67,35923558035688
Test 24:59,020170222641816:66,10579661523427:94,96895701360887
Test 25:52,43224918046058:58,99899504200043:89,08047870314059
Test 26:57,12400015736799:57,762480669617446:68,62488455015536
Test 27:44,730802915806365:69,9043065778114:85,89147493932023
Test 28:48,24838546864551:60,7043607693469:71,36495180198067
Test 29:59,73166557734882:56,31331571534148:75,40694698784418
''')

df = pd.DataFrame()
#Pon tu código aquí.
#Sugerencia: usa el método pandas read_cvs.

#

print(df.head())
```

```
Empty DataFrame
Columns: []
Index: []
```

Ejercicio 04: Dibuja un gráfico de caja para cada clasificador representado en el dataframe cargado en el ejercicio anterior.

Ajusta los parámetros para que la figura generada se aproxime lo más posible a la siguiente:



In [11]:
```
import matplotlib.ticker as ticker #para manipular los ticks de los ejes.
fig = None
axs = None
#Pon tu código aquí.
#Sugerencia queremos generar tres graficos en horizontal. Usa el comando subplots().
#Utiliza los argumentos 'sharey' para que los tres gráficos usen la misma escala en el
#eje Y.

#

#Pon tu código aquí.
#Sugerencia: usa el método suptitle del objeto Figure para añadir un título general.

#

#Pon tu código aquí.
#Dibuja cada caja para el C1 en su Axes correspondiente. Ponle el título.
#Para conseguir que no se utilicen ticks en el eje X debes indicar

#
if axs is not None:
    axs[0].xaxis.set_major_locator(ticker.NullLocator()) #Desactiva los ticks en el eje

#Pon tu código aquí.
#Dibuja cada caja para el C2 en su Axes correspondiente. Ponle el título.
#Para conseguir que no se utilicen ticks en el eje X debes indicar

#
if axs is not None:
    axs[1].xaxis.set_major_locator(ticker.NullLocator())

#Pon tu código aquí.
```

```
#Dibuja cada caja para el C2 en su Axes correspondiente. Ponle el título.
#Para conseguir que no se utilicen ticks en el eje X debes indicar

#
if axs is not None:
    axs[2].xaxis.set_major_locator(ticker.NullLocator())
```

In [ ]: