Plotting

Learn how to plot DataFrames using the pyplot API from Matplotlib.

Chapter Goals:

- Learn how to plot DataFrames using the pyplot API

A. Basics

The main function used for plotting DataFrames is `plot (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.html)`. This function is used in tandem with the `show (https://matplotlib.org/api/_as_gen/matplotlib.pyplot.show.html)` function from the pyplot API, to produce plot visualizations. We import the pyplot API with the line:

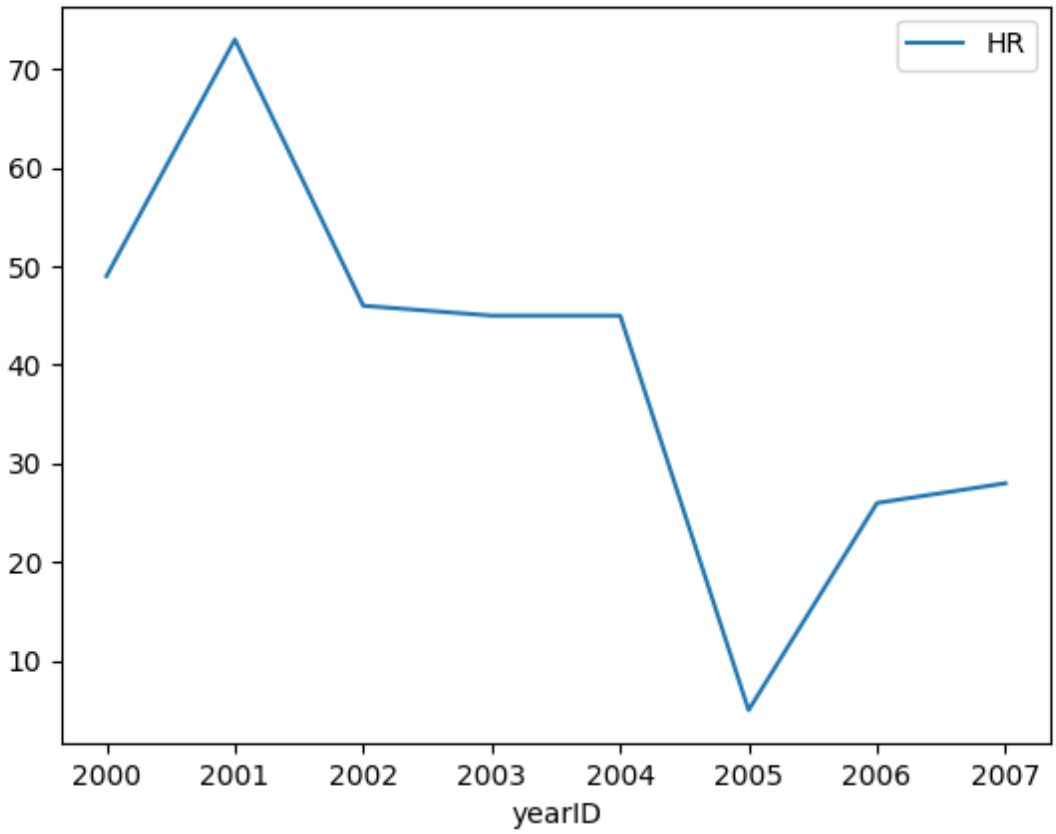
```
import matplotlib.pyplot as plt
```

```
1  # predefined df
2  print('{}\n'.format(df))
3
4  df.plot()
5  plt.show()
```

Output 1.172s

```
   HR  yearID
0  49    2000
1  73    2001
2  46    2002
3  45    2003
4  45    2004
5   5    2005
```

The above code results in this plot:



After calling `df.plot`, which creates our line plot, we then use `plt.show` to open a separate window containing the visualization of the plot. You can also use `plt.savefig` (`https://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.savefig`) to save the plot to a PNG or PDF file.

```
# predefined df
print('{}\n'.format(df))

df.plot()
plt.savefig('df.png')  # save to PNG file
```

Output

2.903s

```
   HR  yearID
0  49    2000
1  73    2001
2  46    2002
3  45    2003
4  45    2004
5   5    2005
6  26    2006
7  28    2007
```

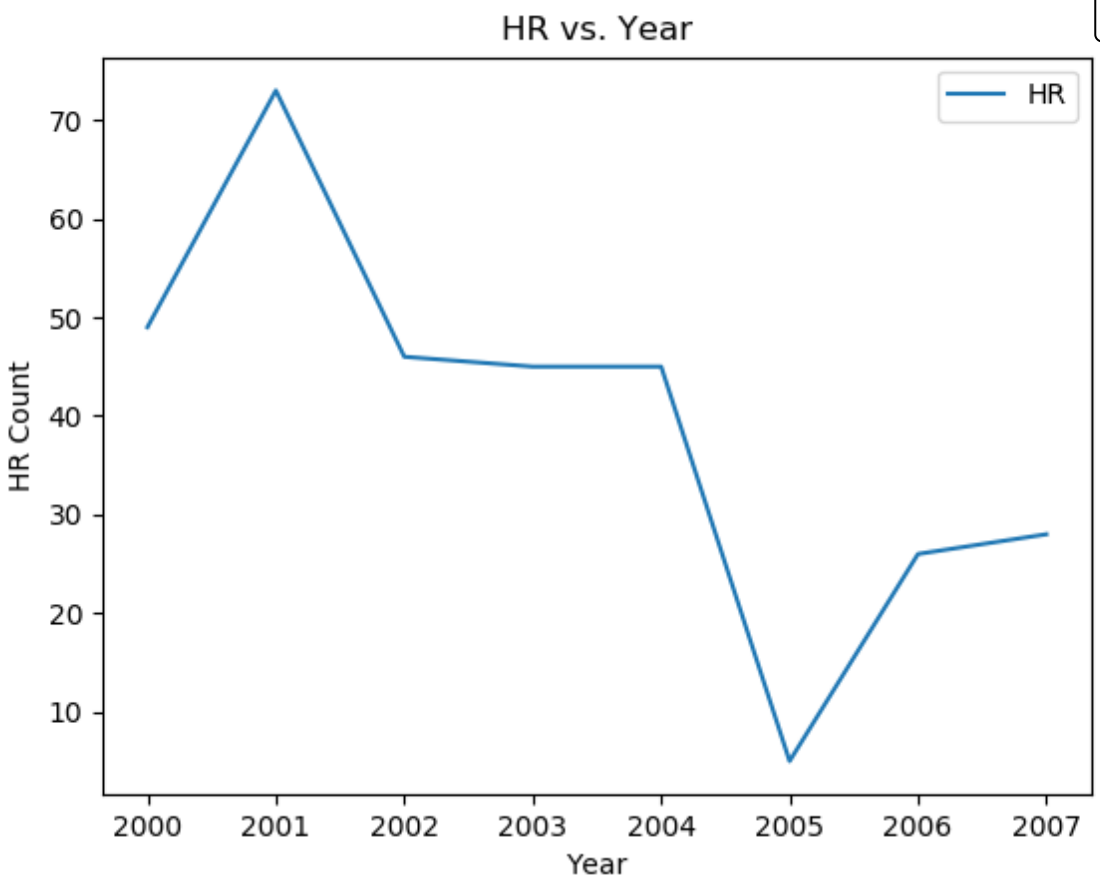
The plot we created has no title or *y*-axis label. We can manually set the plot's title and axis labels using the pyplot API.

```
1  # predefined df
2  print('{}\n'.format(df))
3
4  df.plot()
5  plt.title('HR vs. Year')
6  plt.xlabel('Year')
7  plt.ylabel('HR Count')
8  plt.show()
```

Output 1.628s

```
   HR  yearID
0  49    2000
1  73    2001
2  46    2002
3  45    2003
4  45    2004
5   5    2005
6  26    2006
7  28    2007
```

The above code results in this plot:



We use the `title` (https://matplotlib.org/api/_as_gen/matplotlib.pyplot.title.html) function to set the title of our plot, and the `xlabel` (https://matplotlib.org/api/_as_gen/matplotlib.pyplot.xlabel.html) and `ylabel` (https://matplotlib.org/api/_as_gen/matplotlib.pyplot.ylabel.html) functions to set the axis labels.

B. Other plots

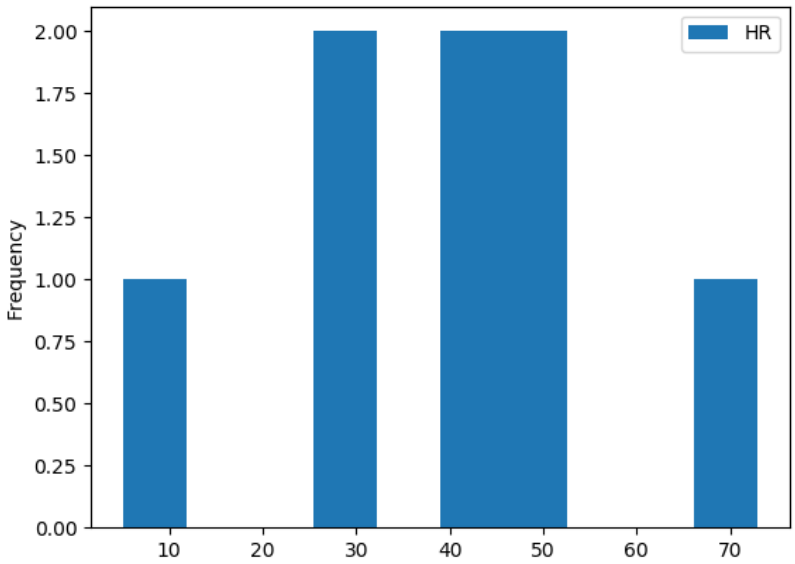
In addition to basic line plots, we can create other plots like histograms or boxplots by setting the `kind` keyword argument in `plot`.

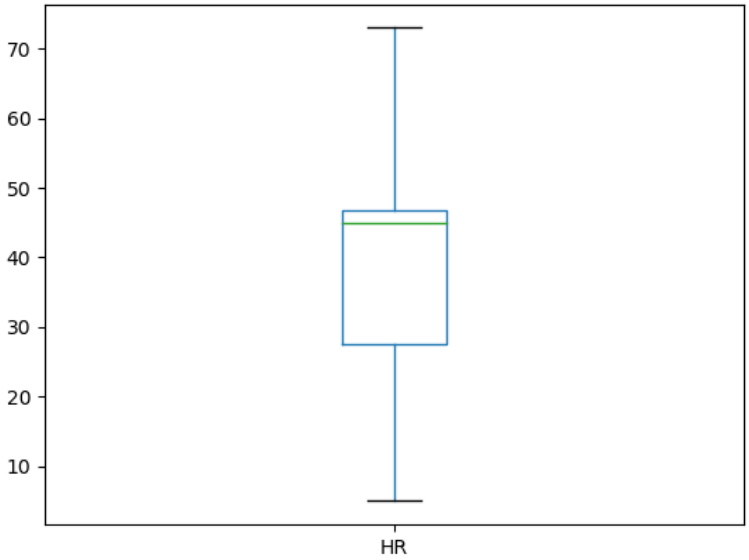
```
# predefined df
print('{}\n'.format(df))

df.plot(kind='hist')
df.plot(kind='box')
plt.show()
```

Output 1.200s

	HR	yearID
0	49	2000
1	73	2001
2	46	2002
3	45	2003
4	45	2004
5	5	2005
6	26	2006
7	28	2007

The above code results in these plots:





There are numerous different kinds of plots we can create by setting the `kind` keyword argument. A list of the accepted values for `kind` can be found in the documentation (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.html#pandas.DataFrame.plot) for `plot`.

C. Multiple features

We can also plot multiple features on the same graph. This can be extremely useful when we want visualizations to compare different features.

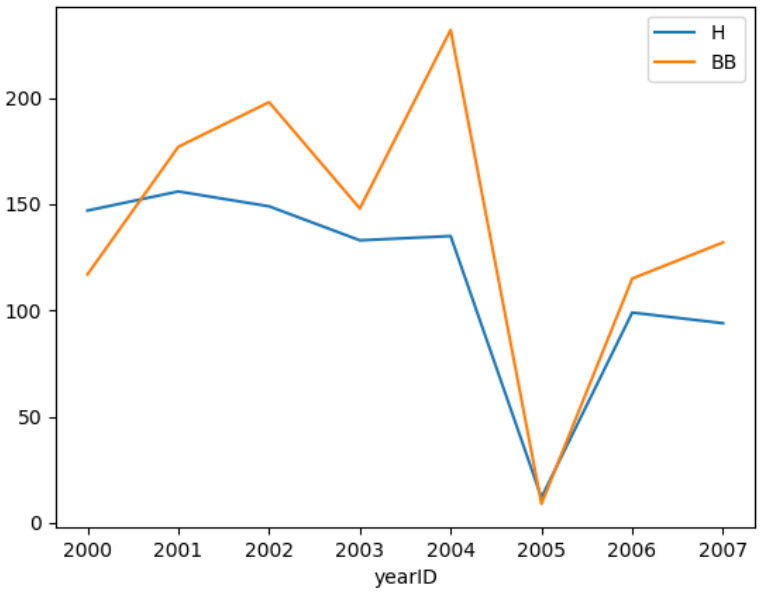
```
1  # predefined df
2  print('{}\n'.format(df))
```

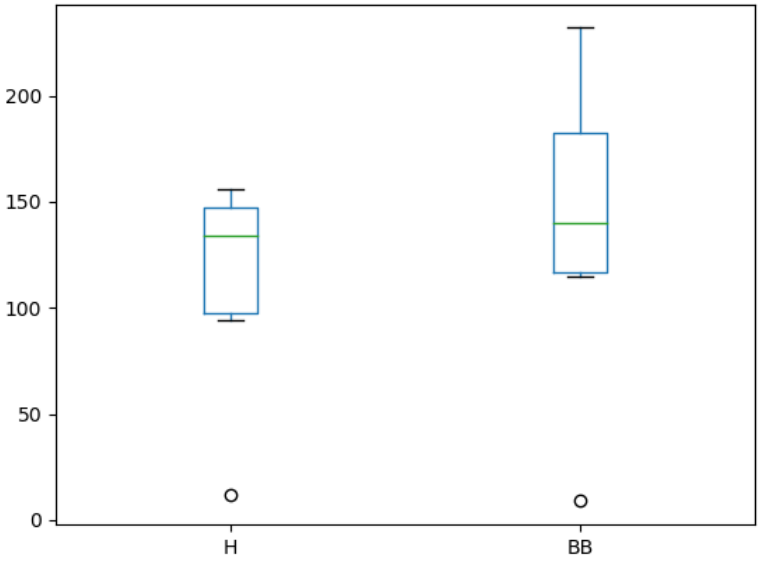
```
3
4  df.plot()
5  df.plot(kind='box')
6  plt.show()
```

Output 1.197s

```
     BB    H  yearID
0   117  147    2000
1   177  156    2001
2   198  149    2002
3   148  133    2003
4   232  135    2004
5     9   12    2005
6   115   99    2006
7   132   94    2007
```

The above code results in these plots:





These are a line plot and boxplot showing both hits (`H`) and walks (`BB`). Note that the circles in the boxplot represent outlier values.

Back

Metrics

Next

To NumPy

Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/plotting__data-analysis-with-pandas__machine-learning-for-software-engineers)