Learning Python - II

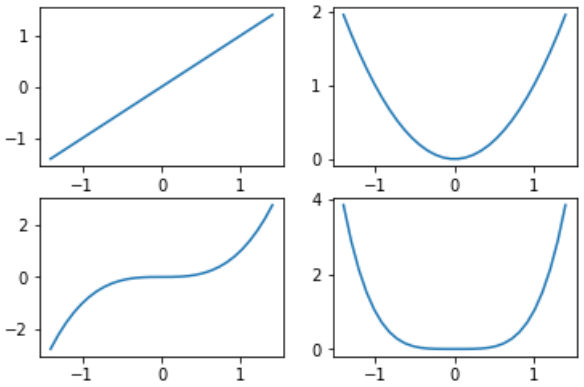
Hands-on Workshop @ Marwadi University, Rajkot

Instructor: Santosh Chapaneri

Matplotlib

Subplots

In [0]:
```
import numpy as np
import matplotlib.pyplot as plt

x  =  np.linspace(-1.4, 1.4, 30)
plt.subplot(2, 2, 1)  # top left
plt.plot(x, x)
plt.subplot(2, 2, 2)  # top right
plt.plot(x, x**2)
plt.subplot(2, 2, 3)  # bottow left
plt.plot(x, x**3)
plt.subplot(2, 2, 4)  # bottom right
plt.plot(x, x**4)
plt.show()
```



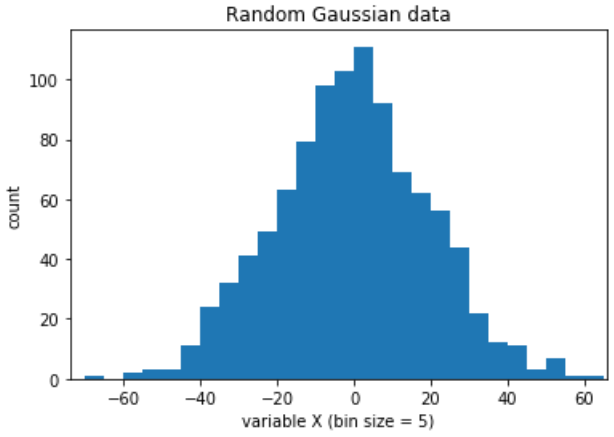
Histograms

The Gaussian PDF is given by:

$$X \sim \mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

In [0]:
```
import numpy as np
from matplotlib import pyplot as plt

data = np.random.normal(0, 20, 1000)    # mean, std, numSamples

# fixed bin size
bins = np.arange(-100, 100, 5)
print(bins)

plt.xlim([min(data)-5, max(data)+5])
plt.hist(data, bins=bins, alpha=1)
plt.title('Random Gaussian data')
plt.xlabel('variable X (bin size = 5)')
plt.ylabel('count')

plt.show()
```

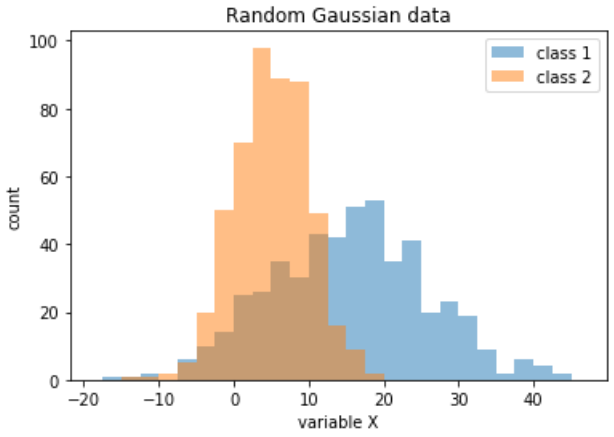
```
[-100  -95  -90  -85  -80  -75  -70  -65  -60  -55  -50  -45  -40  -35
  -30  -25  -20  -15  -10   -5    0    5   10   15   20   25   30   35
   40   45   50   55   60   65   70   75   80   85   90   95]
```



Histogram of 2 overlapping data sets

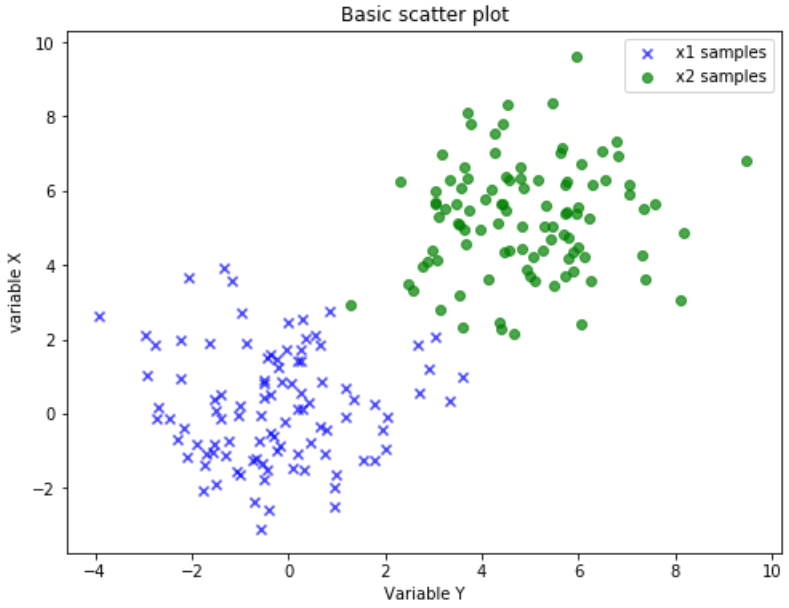
In [0]:
```
import random
%matplotlib inline

data1 = [random.gauss(15,10) for i in range(500)]
# 500 samples from Gaussian distribution, mu, sigma
data2 = [random.gauss(5,5) for i in range(500)]
# 500 samples from Gaussian distribution, mu, sigma

bins = np.arange(-60, 60, 2.5)
plt.xlim([min(data1+data2)-5, max(data1+data2)+5])

plt.hist(data1, bins=bins, alpha=0.5, label='class 1')
plt.hist(data2, bins=bins, alpha=0.5, label='class 2')
plt.title('Random Gaussian data')
plt.xlabel('variable X')
plt.ylabel('count')
plt.legend(loc='upper right')
```

Out[0]: <matplotlib.legend.Legend at 0x7fdf2c83ca90>



Scatter Plots

In [0]:
```
# Generating a Gaussion dataset:
# creating random vectors from the multivariate normal distribution
mu_vec1 = np.array([0,0])
cov_mat1 = np.array([[2,0],[0,2]])

x1_samples = np.random.multivariate_normal(mu_vec1, cov_mat1, 100)
x2_samples = np.random.multivariate_normal(mu_vec1+5, cov_mat1+0.8, 100)

# print x1_samples.shape
# (100, 2), 100 rows, 2 columns

plt.figure(figsize=(8,6))
plt.scatter(x1_samples[:,0], x1_samples[:,1], marker='x',
            color='blue', alpha=0.7, label='x1 samples')
plt.scatter(x2_samples[:,0], x2_samples[:,1], marker='o',
            color='green', alpha=0.7, label='x2 samples')
plt.title('Basic scatter plot')
plt.ylabel('variable X')
plt.xlabel('Variable Y')
plt.legend(loc='upper right')
plt.show()
```



Boxplot

Five-number summary:

- min = minimum value
- 25% = first quartile (Q1) = median of the lower half of the data
- 50% = second quartile (Q2) = median of the data
- 75% = third quartile (Q3) = median of the upper half of the data
- max = maximum value

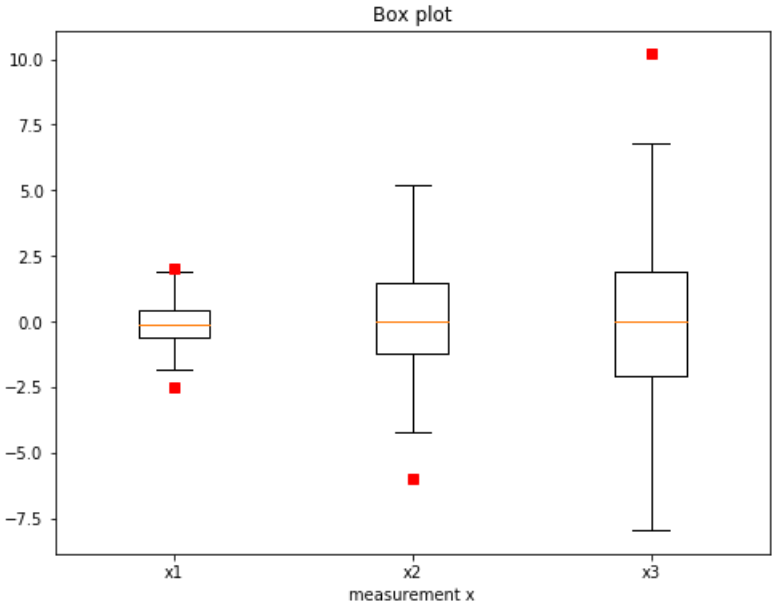
(More useful for describing skewed distributions)

Interquartile Range (IQR) = Q3 - Q1 **Outliers:**

- below Q1 - 1.5 * IQR
- above Q3 + 1.5 * IQR

In [0]:
```
all_data = [np.random.normal(0, std, 100) for std in range(1, 4)]

fig = plt.figure(figsize=(8,6))

plt.boxplot(all_data,
            notch=False, # box instead of notch shape
            sym='rs',    # red squares for outliers
            vert=True)   # vertical box aligmnent

plt.xticks([y+1 for y in range(len(all_data))], ['x1', 'x2', 'x3'])
plt.xlabel('measurement x')
plt.title('Box plot')
```

Out[0]: Text(0.5, 1.0, 'Box plot')



Drinks Dataset

WHO alcohol consumption data:

article: http://fivethirtyeight.com/datalab/dear-mona-followup-where-do-people-drink-the-most-beer-wine-and-spirits/ (http://fivethirtyeight.com/datalab/dear-mona-followup-where-do-people-drink-the-most-beer-wine-and-spirits/)

original data: https://github.com/fivethirtyeight/data/tree/master/alcohol-consumption (https://github.com/fivethirtyeight/data/tree/master/alcohol-consumption)

In [0]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [0]:
```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving drinks.csv to drinks.csv
```

In [0]:
```
import io

drink_cols = ['country', 'beer', 'spirit', 'wine', 'liters', 'continent']

drinks = pd.read_csv(io.BytesIO(uploaded['drinks.csv']),header=0,
                     names=drink_cols, na_filter=False)
```

- how many glasses of wine, cans of beer and shots of spirits were drunk per person in each country in 2010
- liters of pure alcohol consumed in 2010

In [0]:
```
drinks.head()
```

Out[0]:

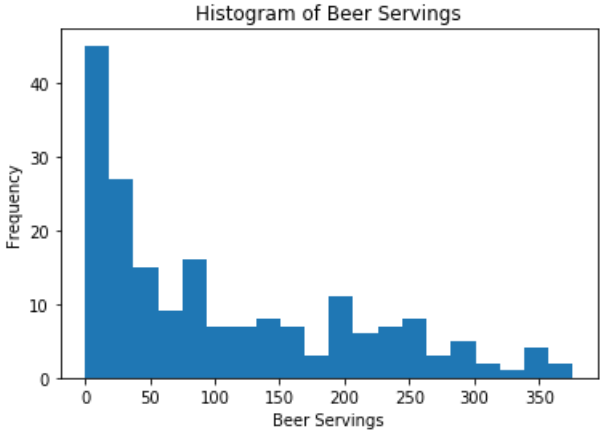
	country	beer	spirit	wine	liters	continent
0	Afghanistan	0	0	0	0.0	AS
1	Albania	89	132	54	4.9	EU
2	Algeria	25	0	14	0.7	AF
3	Andorra	245	138	312	12.4	EU
4	Angola	217	57	45	5.9	AF

In [0]:
```
drinks.shape
```

Out[0]:
```
(193, 6)
```

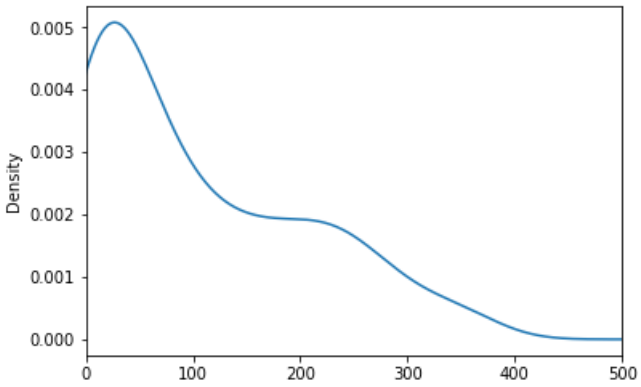
In [0]:
```
# Histogram of Beer
drinks.beer.plot(kind='hist', bins=20, title='Histogram of Beer Servings')
plt.xlabel('Beer Servings')
plt.ylabel('Frequency')
```

Out[0]:
```
Text(0, 0.5, 'Frequency')
```



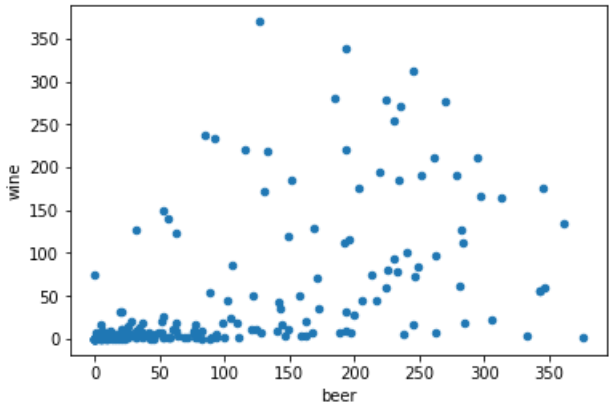
In [0]:
```
# compare with density plot (smooth version of a histogram)
drinks.beer.plot(kind='density', xlim=(0, 500))
```

Out[0]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdf2c8dc160>
```



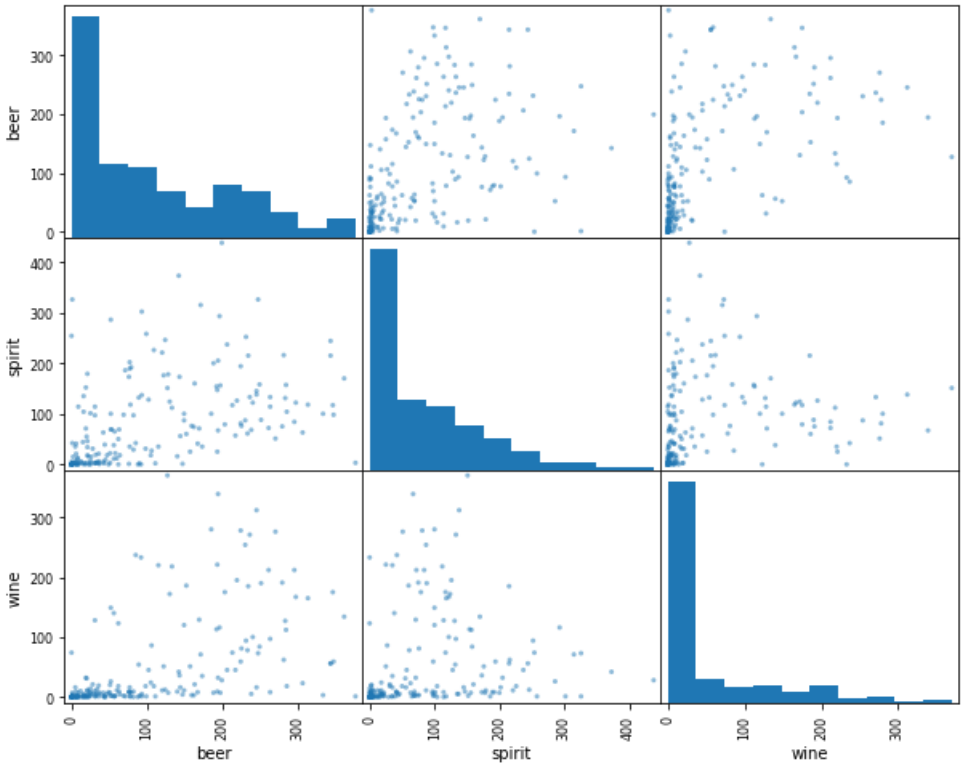
In [0]:
```
# Scatter Plot: show the relationship between two numerical variables
drinks.plot(kind='scatter', x='beer', y='wine')
```

Out[0]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdf240dfef0>
```



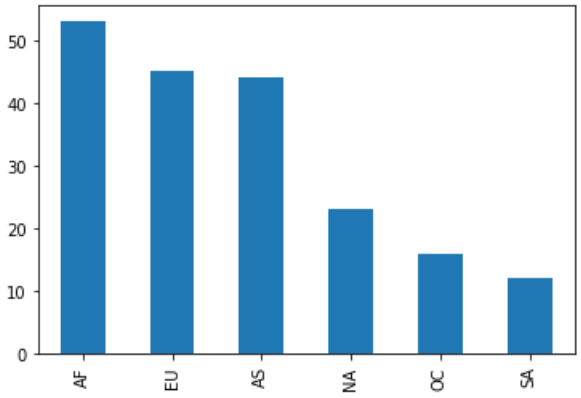
In [0]:
```
# scatter matrix of three numerical columns
pd.plotting.scatter_matrix(drinks[['beer', 'spirit', 'wine']], figsize=(10, 8))
```

Out[0]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fdf240bb4a8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdf24069550>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdf2408fac8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fdf24044080>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdf23fea5f8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdf24011b70>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fdf23fc5128>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdf23f6b6d8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdf23f6b710>]],
      dtype=object)
```



In [0]:
```
# Bar Plot: show a numerical comparison across different categories
drinks.continent.value_counts().plot(kind='bar')
```

Out[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7fdf23e85c18>



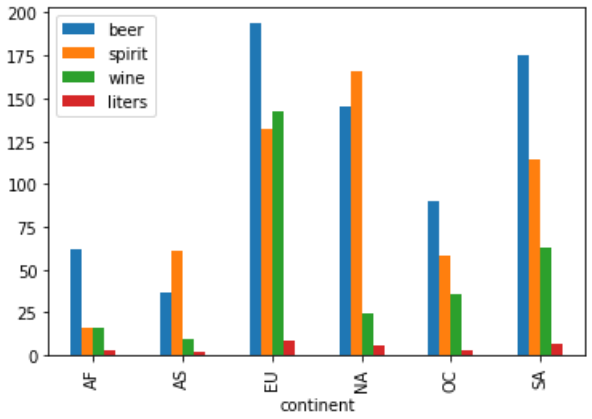
In [0]:
```
# calculate the mean alcohol amounts for each continent
drinks.groupby('continent').mean()
```

Out[0]:

	beer	spirit	wine	liters
continent				
AF	61.471698	16.339623	16.264151	3.007547
AS	37.045455	60.840909	9.068182	2.170455
EU	193.777778	132.555556	142.222222	8.617778
NA	145.434783	165.739130	24.521739	5.995652
OC	89.687500	58.437500	35.625000	3.381250
SA	175.083333	114.750000	62.416667	6.308333

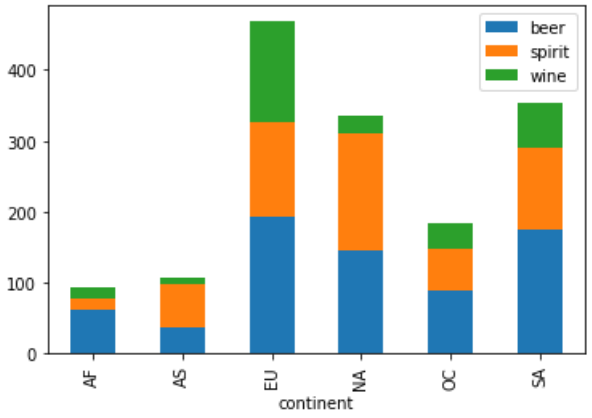
In [0]:
```
# side-by-side bar plots
drinks.groupby('continent').mean().plot(kind='bar')
```

Out[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7fdf225ceac8>



In [0]:
```
# Stacked bar plots (without liters attribute)
drinks.groupby('continent').mean().drop('liters', axis=1).plot(kind='bar', stacked=True)
```

Out[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7fdf224e47f0>

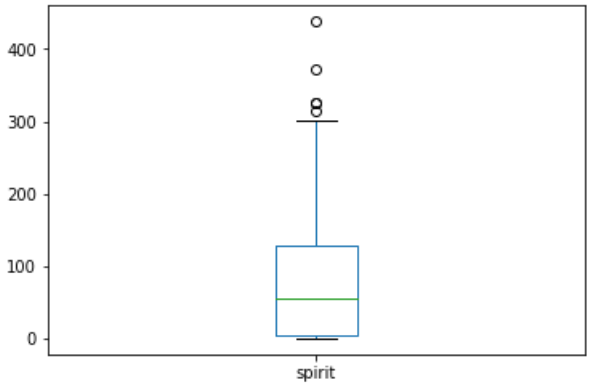


In [0]:
```
# Box Plot: show quartiles (and outliers) for one or more numerical variables
# show "five-number summary" for spirit
drinks.spirit.describe()
```

Out[0]:
```
count    193.000000
mean      80.994819
std       88.284312
min        0.000000
25%        4.000000
50%       56.000000
75%      128.000000
max      438.000000
Name: spirit, dtype: float64
```

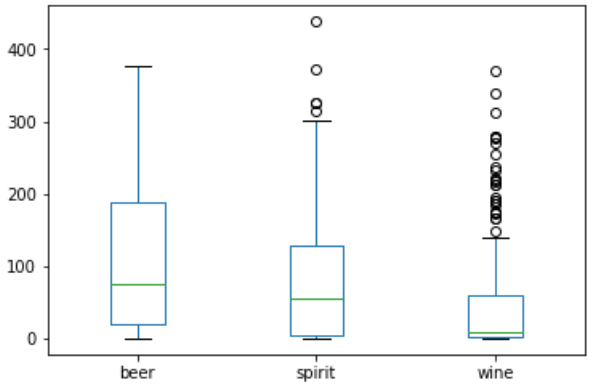
In [0]:
```
drinks.spirit.plot(kind='box')
```

Out[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7fdf23de5da0>



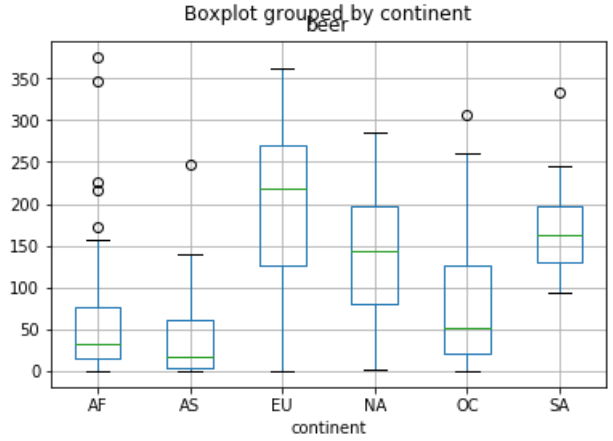
In [0]:
```
# include multiple variables
drinks.drop('liters', axis=1).plot(kind='box')
```

Out[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7fdf223d83c8>



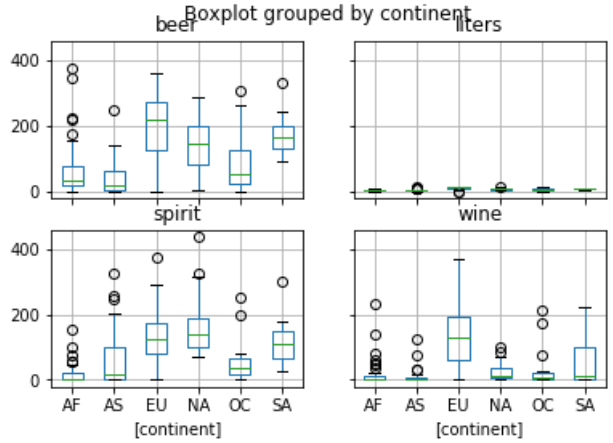
In [0]:
```
# Grouped Box Plots: show one box plot for each group

# Box plot of beer servings grouped by continent
drinks.boxplot(column='beer', by='continent')
```

Out[0]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdf223ca7f0>
```



In [0]:
```
# Box plot of all numeric columns grouped by continent
drinks.boxplot(by='continent')
```

Out[0]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fdf222dfc88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdf222c22e8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fdf2226b588>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdf22292860>]],
      dtype=object)
```

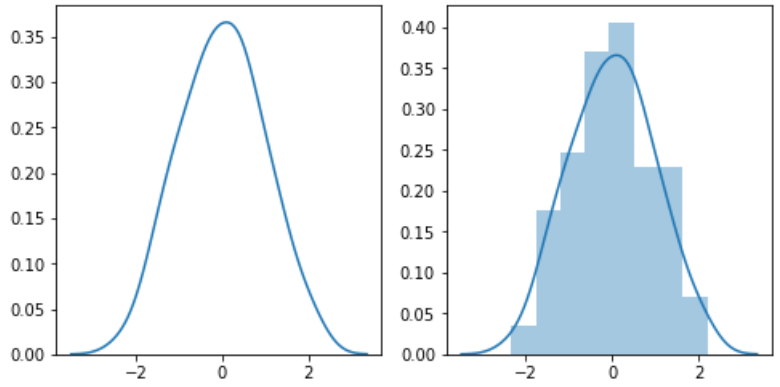


Seaborn

If Matplotlib “tries to make easy things easy and hard things possible”, Seaborn tries to make a well-defined set of hard things easy too.

In [0]:
```
import seaborn as sns
import numpy.random as rng
import matplotlib.pyplot as plt
%matplotlib inline

xs = rng.normal(0,1,100)

fig, axes = plt.subplots(1, 2, figsize=(8,4))
sns.distplot(xs, hist=False, ax=axes[0]);
sns.distplot(xs, hist=True, ax=axes[1])
```

Out[0]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdf2172b358>
```



In [0]:
```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving iris.csv to iris.csv
```

In [0]:
```
import io
iris = pd.read_csv(io.BytesIO(uploaded['iris.csv']))
```

In [0]:
```
iris.head()
```

Out[0]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

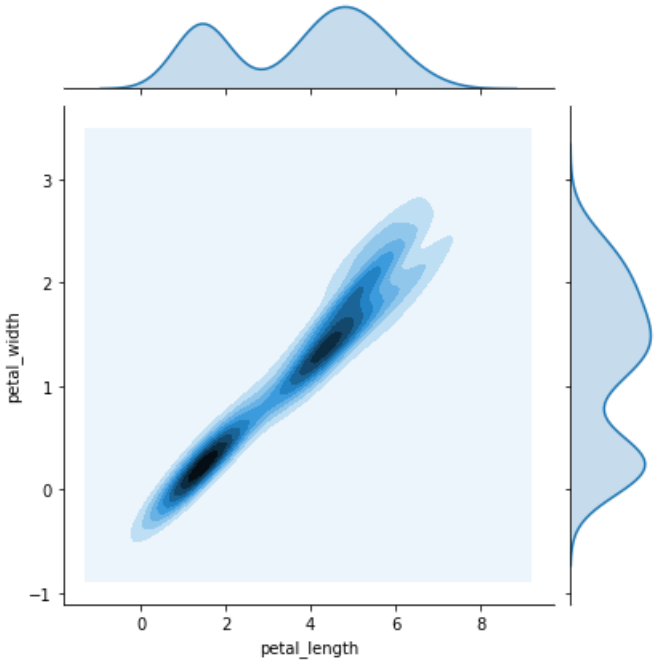
In [0]:
```
# Let's see how many examples we have of each species
iris['species'].value_counts()
```

Out[0]:
```
versicolor    50
setosa        50
virginica     50
Name: species, dtype: int64
```

Joint distribution plot - Seaborn

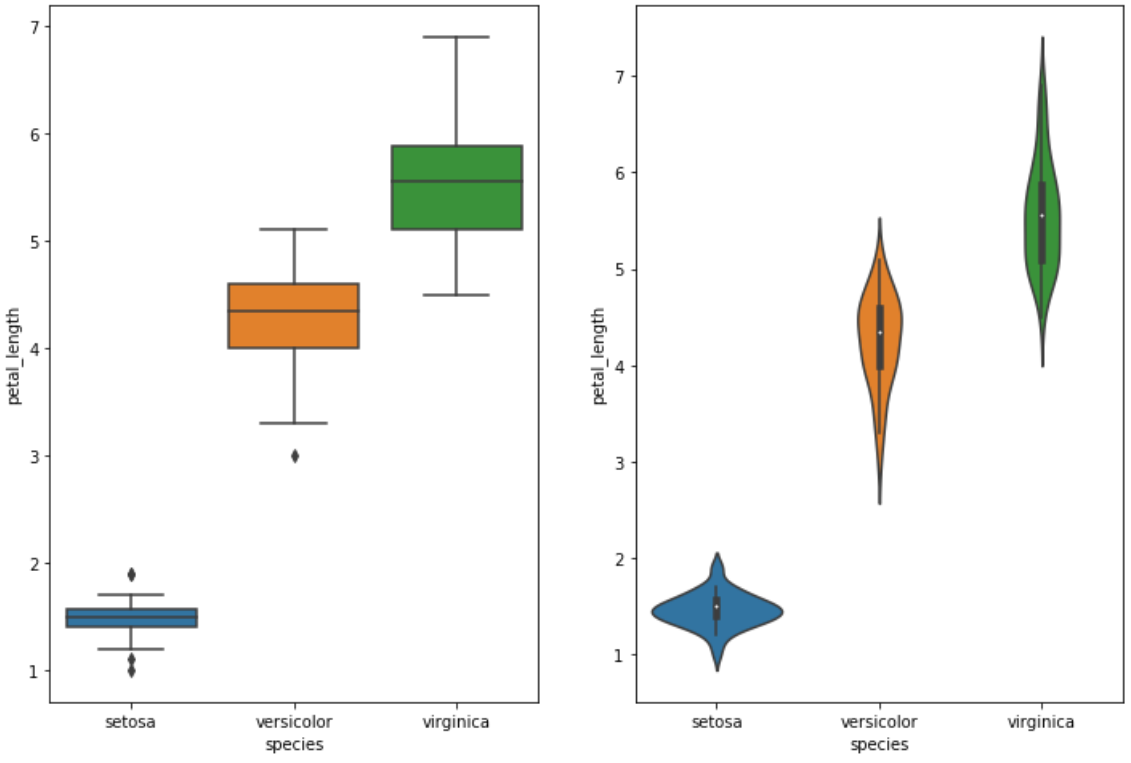
In [0]:
```
sns.jointplot(x='petal_length', y='petal_width', data=iris, kind='kdeplot')
```

Out[0]: <seaborn.axisgrid.JointGrid at 0x7fdf21f97a90>



In [0]:
```
fig, axes = plt.subplots(1, 2, figsize=(12,8))

sns.boxplot(x='species', y='petal_length', data=iris, ax=axes[0])
sns.violinplot(x='species', y='petal_length', data=iris, ax=axes[1])
```

Out[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7fdf21563f28>



In [0]:
```
# Use different markers for each level of the hue variable:
sns.pairplot(iris, hue='species', markers=['o', 's', 'D'])
```

Out[0]: <seaborn.axisgrid.PairGrid at 0x7fdf21fcf128>