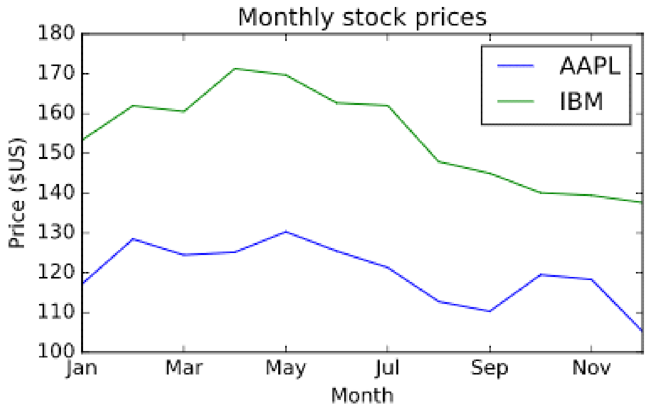
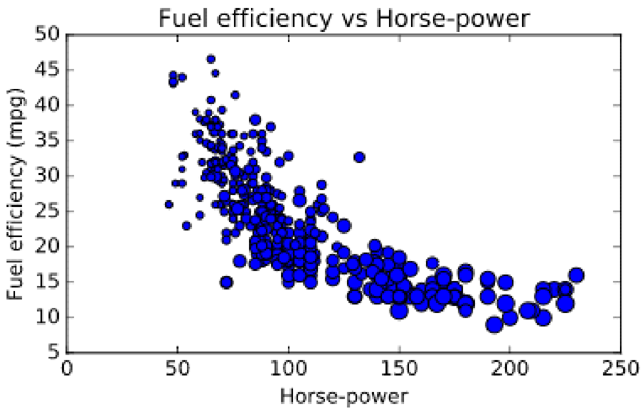
Ch 2 Exploratory data analysis

November-15-17 9:47 PM

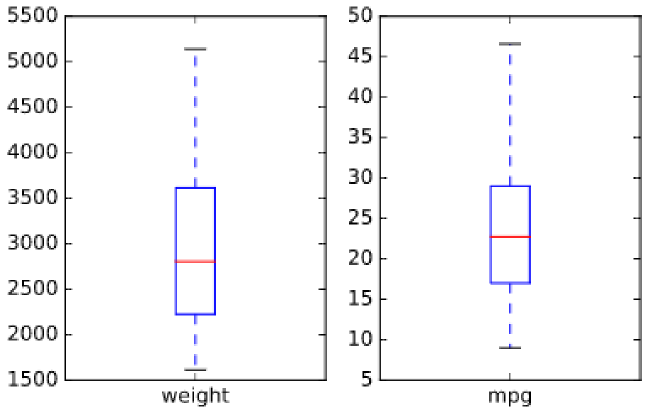
```
#pandas line plots
# Create a list of y-axis column names: y_columns
y_columns = ['AAPL', 'IBM']

# Generate a line plot
df.plot(x='Month', y=y_columns)

# Add the title
plt.title('Monthly stock prices')

# Add the y-axis label
plt.ylabel('Price ($US)')

# Display the plot
plt.show()
```



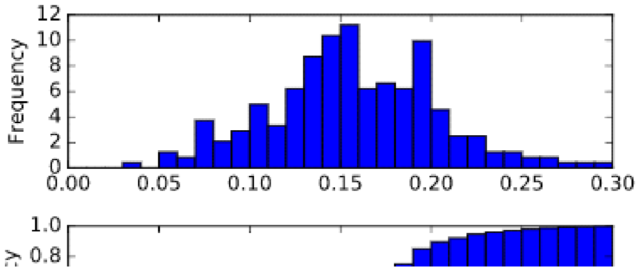
```
#pandas scatter plots
# Generate a scatter plot
df.plot(kind='scatter', x='hp', y='mpg', s=sizes)

# Add the title
plt.title('Fuel efficiency vs Horse-power')

# Add the x-axis label
plt.xlabel('Horse-power')

# Add the y-axis label
plt.ylabel('Fuel efficiency (mpg)')

# Display the plot
plt.show()
```



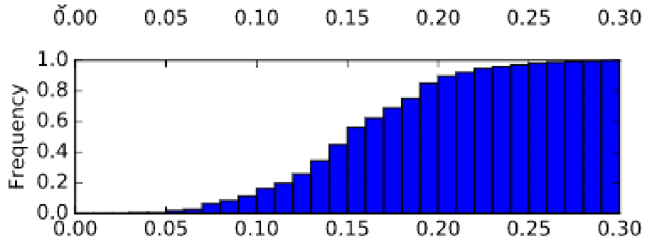
```
#pandas box plots
# Make a list of the column names to be plotted: cols
cols = ['weight', 'mpg']

# Generate the box plots
df[cols].plot (kind = 'box', subplots = True)

# Display the plot
plt.show()
```



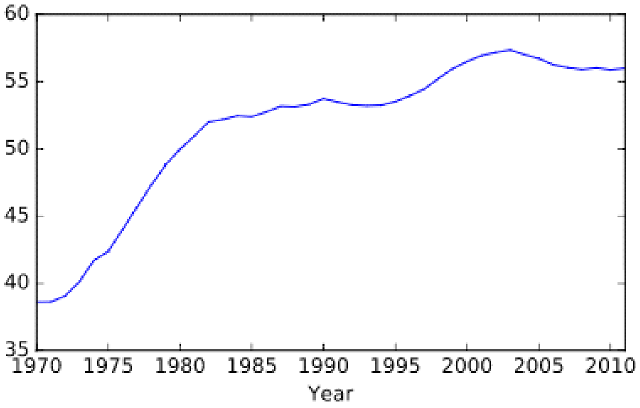
```
#pandas hist, pdf and cdf
# This formats the plots such that they appear on separate rows
fig, axes = plt.subplots(nrows=2, ncols=1)

# Plot the PDF
df.fraction.plot(ax=axes[0], kind='hist', normed=True, bins=30, range=(0,.3))
plt.show()
```



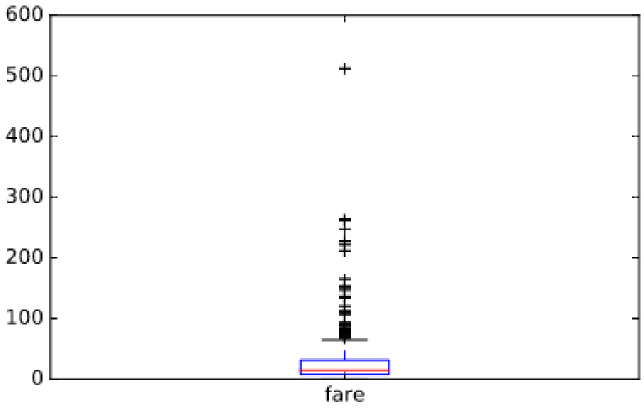
```
# Plot the PDF
df.fraction.plot(ax=axes[0], kind='hist', normed=True, bins=30,
range=(0,.3))
plt.show()

# Plot the CDF
df.fraction.plot(ax = axes[1], kind='hist', normed=True, bins=
30, cumulative =True, range=(0,.3))
plt.show()
```



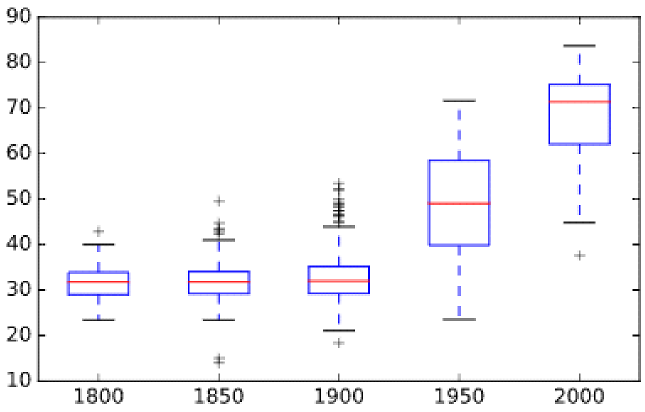
```
#Bachelor's degrees awarded to women
# Print the minimum value of the Engineering column
print(df ['Engineering'].min())

# Print the maximum value of the Engineering column
print(df ['Engineering'].max())

# Construct the mean percentage per year: mean
mean = df.mean(axis='columns')

# Plot the average percentage per year
mean.plot ()

# Display the plot
plt.show()
```



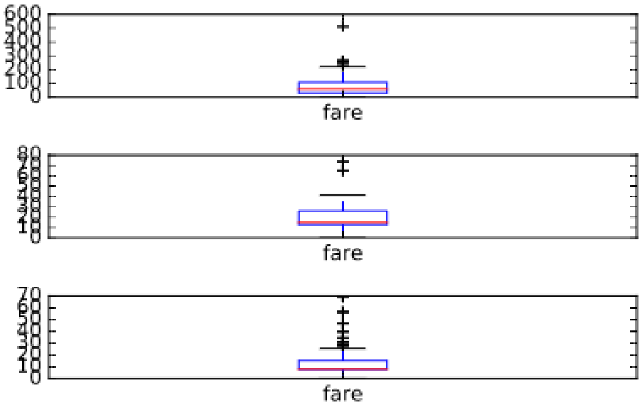
```
#Median vs mean
# Print summary statistics of the fare column with .describe()
print(df['fare'].describe())

# Generate a box plot of the fare column
df['fare'].plot (kind = 'box')

# Show the plot
plt.show()
```



```
#Quantiles
# Print the number of countries reported in 2015
print(df['2015'].count())

# Print the 5th and 95th percentiles
print(df.quantile ([0.05, 0.95]))

# Generate a box plot
years = ['1800','1850','1900','1950','2000']
df[years].plot(kind='box')
plt.show()
```



```
#Standard deviation of temperature
# Print the mean of the January and March data
print(january.mean(), march.mean ())

# Print the standard deviation of the January and March data
print (january.std())
print (march.std())



df[df['origin'] == 'US']


#Separate and summarize
# Compute the global mean and global standard deviation:
global_mean, global_std
global_mean = df.mean()
global_std = df.std()

# Filter the US population from the origin column: us
us = df[df['origin'] =='US']

# Compute the US mean and US standard deviation: us_mean,
us_std
us_mean = us.mean()
us_std = us.std()

# Print the differences
print(us_mean - global_mean)
print(us_std - global_std)



#Separate and plot
# Display the box plots on 3 separate rows and 1 column
fig, axes = plt.subplots(nrows=3, ncols=1)

# Generate a box plot of the fare prices for the First passenger
class
titanic.loc[titanic['pclass'] == 1].plot(ax=axes[0], y='fare',
kind='box')

# Generate a box plot of the fare prices for the Second
passenger class
titanic.loc[titanic['pclass'] == 2].plot(ax=axes[1], y='fare',
kind='box')

# Generate a box plot of the fare prices for the Third
passenger class
titanic.loc[titanic['pclass'] == 3].plot(ax=axes[2], y='fare',
kind='box')

# Display the plot
plt.show()
```


fare
fare
fare