Solutions 3

Jumping Rivers

Graphics

We will continue to investigate the movies data from earlier. To begin we will load the data and then take a random sample of 500 values to help keep the plots a bit cleaner.

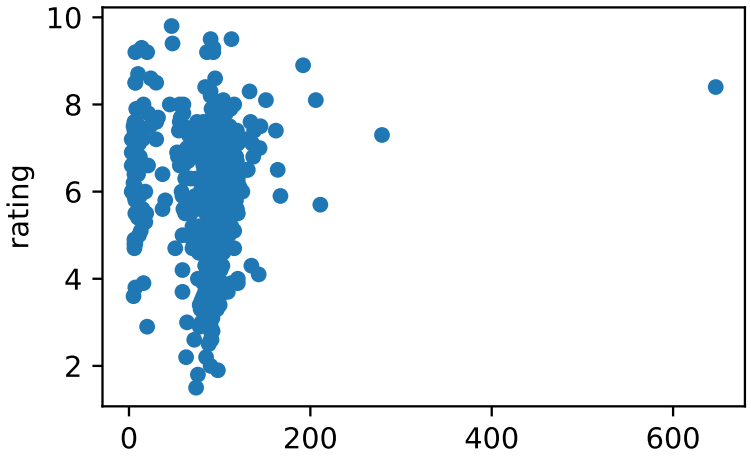
```
import jrpyml.datasets as dat
movies = dat.load_movies()
movies = movies.sample(500)
```

Also load all of the packages that we might need for this practical

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

1. Start with a simple scatter plot of movie ratings against lengths. It should look something like the one below.

```
p1 = movies.plot.scatter(x = 'length', y = 'rating')
plt.show(p1)
```



2. Use the `xlim` and `ylim` arguments to change the axis ranges to (0,200) and (0,10) respectively.

```
p2 = movies.plot.scatter(x = 'length', y = 'rating',
  ylim = [0,10], xlim = [0,200]
)
plt.show(p2)
```

3. Change the colours of points such that we have one colour for Comedy films and another colour for non Comedy films.

```
p3 = movies.plot.scatter(
  x = 'length', y = 'rating',
  ylim = [0,10], xlim = [0,200],
  c = 'Comedy', cmap = 'autumn_r'
)
plt.show(p3)
```

4. Relabel your axes with the units given on the x axis

```
p4 = movies.plot.scatter(
  x = 'length', y = 'rating',
  ylim = [0,10], xlim = [0,200],
  c = 'Comedy', cmap = 'autumn_r'
)
plt.xlabel('Length (minutes)')
plt.ylabel('Rating')
plt.show(p4)
```

5. Finally give your graph a title.

```
p5 = movies.plot.scatter(
  x = 'length', y = 'rating',
  ylim = [0,10], xlim = [0,200],
  c = 'Comedy', cmap = 'autumn_r'
)
plt.xlabel('Length (minutes)')
plt.ylabel('Rating')
plt.title('Investigating whether long movies are good.')
plt.show(p5)
```

6. Create a boxplot of movie lengths using the `.boxplot()` method.

```
p6 = movies.boxplot('length')
plt.show(p6)
```

7. You can amend the axes after the plot with `plt.ylim()` and the corresponding `.xlim()`.

```
p7 = movies.boxplot('length' )
plt.ylim(
# calculate the axes limits from the data
  movies.length.min(),
  movies.length.max()
)
plt.show(p7)
```

8. Add a new column to your movies `DataFrame` which corresponds to the decade the film was released. Hint: You can do this by taking the year value, dividing by 10, rounding down using `np.floor()` and then multiplyng by 10 again.

```
movies['decade'] = np.floor(movies.year/10 ) * 10
# To be really good you could then change the type of the
# variable to be an integer
movies.decade = movies.decade.astype('int32')
```

9. Use the `by` argument of the `.boxplot()` method to create a separate boxplot for each decade.

```
p8 = movies.boxplot('length', by = 'decade')
plt.ylim(
# calculate the axes limits from the data
  movies.length.min(),
  movies.length.max()
)
plt.show(p8)
```

If you finish

If you finish, feel free to explore some of the other graphics that we have discussed.