

Solutions 2

Jumping Rivers

DataFrames

For this set of questions we will use the movies data from the IMDB database. This data is contained in the course package `jrpyintroduction`. To load the movies data as a `DataFrame` called `movies` you can use the following code:

```
import jrpyintroduction.datasets as dat
movies = dat.movies.load_data()
```

1. Use the `.head()` method to inspect the top of the data. This can help give you a feel for what the data looks like and what variables are contained within the data.

```
print(movies.head())
```

```
##              title  year  length  budget  rating  votes   r1   r2   r3   r4   r5   ...
## 0              $  1971    121    NaN     6.4    348   4.5   4.5   4.5   4.5  14.5  ...
## 1      $1000 a Touchdown  1939     71    NaN     6.0     20   0.0  14.5   4.5  24.5  14.5  ...
## 2    $21 a Day Once a Month  1941     7    NaN     8.2     5   0.0   0.0   0.0   0.0   0.0  ...
## 3          $40,000  1996     70    NaN     8.2     6  14.5   0.0   0.0   0.0   0.0  ...
## 4  $50,000 Climax Show, The  1975     71    NaN     3.4    17  24.5   4.5   0.0  14.5  14.5  ...
##
## [5 rows x 24 columns]
```

2. How many films and variables are there in this dataset?

```
print(movies.shape)
# 58788 films, 24 variables
## (58788, 24)
```

3. What is the mean and median film length?

```
# either
print(
    movies.length.mean()
)

# or
## 82.33787507654624

import numpy as np
print(
    np.mean(movies.length)
)
```

```
## 82.33787507654624
```

4. What year is the oldest film in the data set from?

```
print(
    movies.year.min()
)
```

```
## 1893
```

5. How long are the longest and shortest films?

```
# I have used different syntax here to highlight there is more than
# one way to extract columns from a data frame.
```

```
print(
    movies['length'].max()
)
```

```
## 5220
```

```
print(
    movies.loc[:, 'length'].min()
)
```

```
## 1
```

6. Calculate the standard deviation of the ratings by using the **numpy** `std()` function.

```
print(
    np.std(movies.rating)
)
```

```
## 1.5530175913581754
```

7. Now calculate the standard deviation using the **DataFrame** member method `.std()`. Is there a difference? If so, why do you think that is?

```
print(
    movies.rating.std()
)
# There is a small difference. This is because np.std calculates
# population standard deviation by default whereas DataFrame.std
# calculates sample standard deviation.
```

```
## 1.553030800154227
```

8. How many action films are in the data? (There is a 1 in the Action column whenever a film belongs to that genre.)

```
print(  
    movies.Action.sum()  
)
```

```
## 4688
```