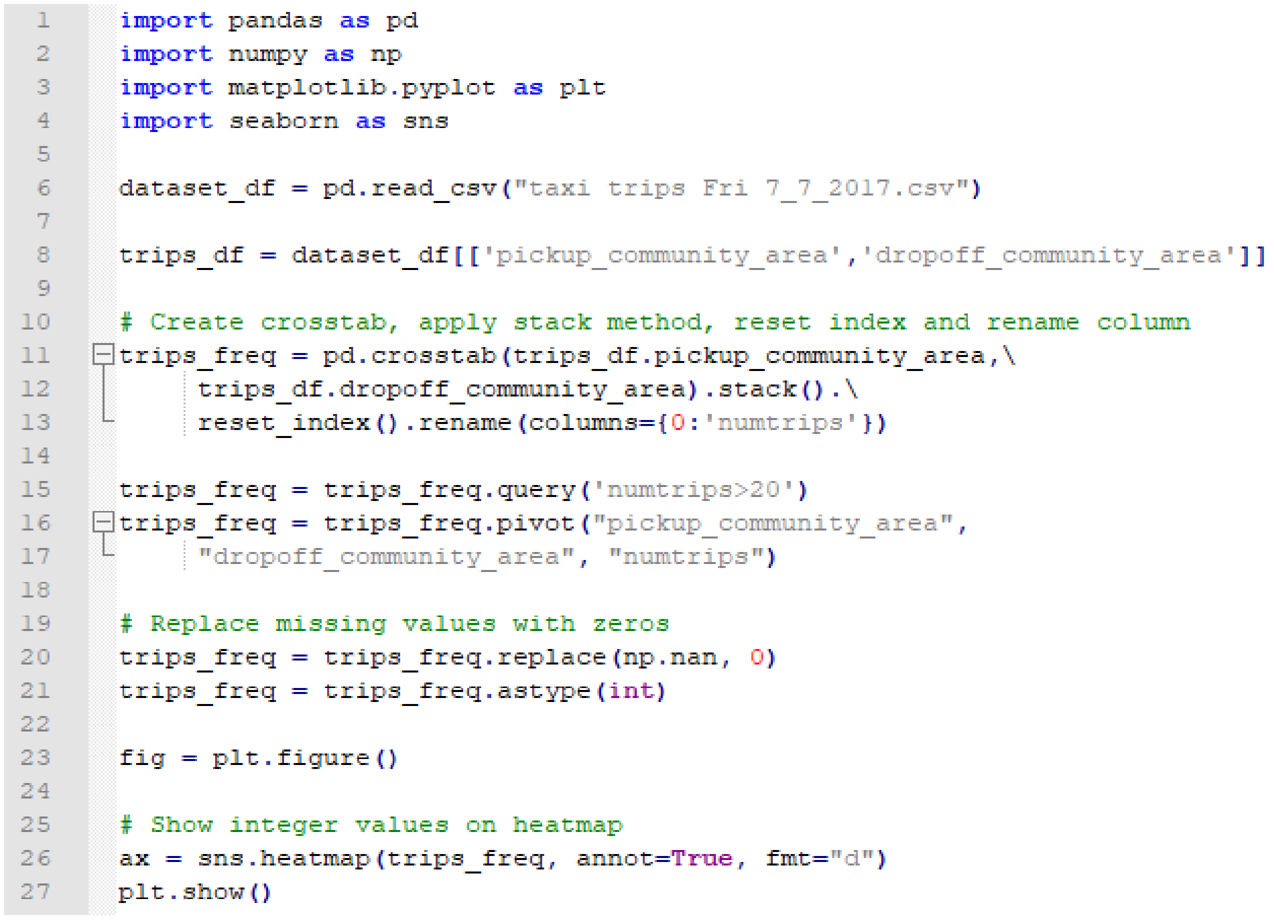
Errata

The following table lists corrections to content in the book that were found after the book was published.

Page	**Issue and Correction**
123	The first learning objective for Chapter 6 should be “Use the NumPy package and the NumPy ndarray object”, not “Explain good programming practice guidelines”.
188	In the first sentence following Figure 8.2, “… covered in Chapter 3.”, should say “… covered in Chapter 4.”.
195	The sentence “The `linregress` function returns five values, one for the slope of the regression line, one for the intercept, one for the *R*-squared value for the model, one for the *p* value, and one for the standard error.”, should be changed to “The `linregress` function returns five values, one for the slope of the regression line, one for the intercept, one for the correlation coefficient for the model (the pearsons *r* value), one for the *p* value, and one for the standard error.” Later, the sentence “The `r_value` is 0.78, indicating that 78% of the change in fare amount is explained by how long the taxi trip was in miles.”, should be changed to “The `r_value` is 0.78, so the *R*-squared value is 0.78 x 0.78 = 0.608, indicating that 60.8% of the change in fare amount is explained by how long the taxi trip was in miles.”
199	In the very last sentence on the page, “Lines 7 to 8 use the `.fit()` method…”, should be “Lines 8 to 9 use the `.fit()` method…”.
208	In line 6, “`fare_series`” should be “`fareseries`”. In line 12, “… and the horizontal axis displays the frequency …”, should be “… and the vertical axis displays the frequency …”.
210	In the sentence before Figure 9.3, “Figure 9.3 uses the matplotlib `plot` function …” should be “Figure 9.3 uses the matplotlib `bar` function …”.
211	In line 4 of the section “Bar Chart Specific Code Instructions”, “… categories for the histogram.” should be “… categories for the bar chart.”
227	Line 10 was cut off at the bottom of Figure 9.17. The complete figure is:

```
 1  import pandas as pd
 2  import matplotlib.pyplot as plt
 3  import seaborn as sns
 4
 5  dataset_df = pd.read_csv("AnscombeData.csv")
 6  fig = plt.figure()
 7
 8  # Use seaborn lmplot to create multiple plots, one for each dataset
 9  ax = sns.lmplot(x = 'x', y = "y", col = "DataSet", data = dataset_df, ci = None)
10  plt.savefig("AnscombeDataCharted.png", dpi = 300)
```

239 Lines 9-27 were cut off at the bottom of Figure 9.31. The complete figure is:



```
1   import pandas as pd
2   import numpy as np
3   import matplotlib.pyplot as plt
4   import seaborn as sns
5
6   dataset_df = pd.read_csv("taxi trips Fri 7_7_2017.csv")
7
8   trips_df = dataset_df[['pickup_community_area','dropoff_community_area']]
9
10  # Create crosstab, apply stack method, reset index and rename column
11  trips_freq = pd.crosstab(trips_df.pickup_community_area,\
12       trips_df.dropoff_community_area).stack().\
13       reset_index().rename(columns={0:'numtrips'})
14
15  trips_freq = trips_freq.query('numtrips>20')
16  trips_freq = trips_freq.pivot("pickup_community_area",
17       "dropoff_community_area", "numtrips")
18
19  # Replace missing values with zeros
20  trips_freq = trips_freq.replace(np.nan, 0)
21  trips_freq = trips_freq.astype(int)
22
23  fig = plt.figure()
24
25  # Show integer values on heatmap
26  ax = sns.heatmap(trips_freq, annot=True, fmt="d")
27  plt.show()
```

262 In the first paragraph in the section k-Means Clustering Example with Taxi Trips Data Set, “Line 8 of the code …” should be “Line 7 of the code …”, “Line 9 of the code …” should be “Line 8 of the code …”, and “Line 10 of the code …” should be “Line 9 of the code …”.

266 The discussion of the output shown in Figure 10.15 specifies values that are not consistent with the values in Figure 10.15. This discrepancy resulted from our re-execution of the code in Figure 10.14 when preparing the final illustrations for the book. The k-means clustering algorithm uses random starting points, so each time the code in Figure 10.14 is executed different results can occur. To ensure obtaining the same results each time, the scikit-learn `KMeans` function has a keyword argument `random_state` which can be assigned a specific value (just as we used the `random_state` keyword for the `train_test_split` function used in lines 14-15 in the code in Figure 10.1 on page 249). For example, line 18 in Figure 10.14 on page 265 can be changed to the following:

```
kmeans_model = KMeans(n_clusters = 5, random_state = 7).fit(data_transformed)
```