

Day 14: *Combining Functions in Pandas*

Just as Dot left the zoo in Melbourne, chaos reared its head once more. A loud horn blared from over the entrance to the zoo, stopping Dot in their tracks. A crackly loudspeaker clicked on, and from within it, an alarmed voice sent out a message: "All hands on deck! All hands on deck! The wallabies are loose! I repeat, the wallabies are loose!" Around Dot, the zoo's other patrons began to yell and scatter in all directions, running and sprinting, abandoning their bags and strollers. "It's the end!" yelled a frantic, wheezing man as he ran by Dot. A uniformed zoo employee stuck her head out of the gate and pointed toward Dot. "You there! You look calm and trustworthy. Give us a hand in here, will you?" She grabbed Dot by the arm and yanked them back into the zoo. The employee began to run, gesturing for Dot to run with her. "What's happening?" Dot yelled at the woman, and she shook her head as they ran. "You'll see in a minute," she said. "Wait – right there!" Dot followed the woman's pointing finger. In the middle of the walking path, right ahead of them, were two mid-sized wallabies standing upright and rigid, gazing straight at Dot.

"You take the one on the left, and I got the one on the right," the employee said, tip-toeing stealthily towards the wallabies. "What do you mean?" Dot asked fearfully. "Just grab him!" the woman yelled, lunging at her wallaby. Without thinking, Dot pounced at the wallaby, who hopped casually out of the way and aimed a kick at Dot's leg. Dot rolled, and the kick missed them by an inch. They sprung back to their feet and lunged at the wallaby, capturing the fuzzy creature in a bear hug. "What do I do with him?" Dot yelled, struggling with the animal. "Here, friend, here!" The woman yelled. She had the other wallaby wrapped in a giant net; Dot hopped over with their wallaby and nudged him into it. After the wallabies were safe, Dot collapsed to the floor, panting with exertion. "Thanks a lot, kind stranger," the employee said, slapping Dot on the back. "You look like you need a break!" Dot nodded. "This has been a hectic day, but my next destination might be more tranquil – I'm headed over to New Zealand soon." The woman advised Dot to take solace in the New Zealand countryside.

Dot wants to know where the calmest, most relaxing place in New Zealand is. Can you help them figure out where that is by examining New Zealand's road traffic data?

Tutorial

In today's challenge, you will be using and combining the various functions you have learned over the last few challenges.

1. **Dataframe column functions like .min() or .max()**

- introduced in the Day 7 challenge.

2. **Pandas Group By**

- introduced in the Day 10 and 11 challenges.

3. **Pandas Unstack() function**

- introduced in the Day 12 challenge.

4. Plotting in Pandas

- introduced in the Day 13 challenge.

If you have forgotten how to use some of these functions don't worry, data scientists and analysts always have a cheat sheet handy. You can check out the pandas user guide in the [pandas documentation](#) or refer to the [forum](#) or Google for help.

```
In [2]: # We will be using data that contains traffic information for New Zealand during the ye
import pandas as pd
df = pd.read_csv('nz_car_traffic.csv')
df.head()
```

```
Out[2]:
```

	startDate	regionName	classWeight	flowDirection	trafficCount	month
0	2020-01-01	03 - Waikato	Heavy	2	436.0	2020-01
1	2020-01-01	03 - Waikato	Light	2	9700.0	2020-01
2	2020-01-01	03 - Waikato	Light	1	9398.5	2020-01
3	2020-01-01	03 - Waikato	Heavy	1	486.5	2020-01
4	2020-01-01	11 - Canterbury	Light	2	494.5	2020-01

Challenge

- Find the regions (variable `regionName`) which had the lowest amount (sum of `trafficCount`) of each light and heavy traffic during the year 2020. (one region for each `classWeight`)
- Using a bar chart, which month had the lowest amount of total traffic in 2020?
- What was the % of heavy traffic (using variable `trafficCount`) in New Zealand in 2020?

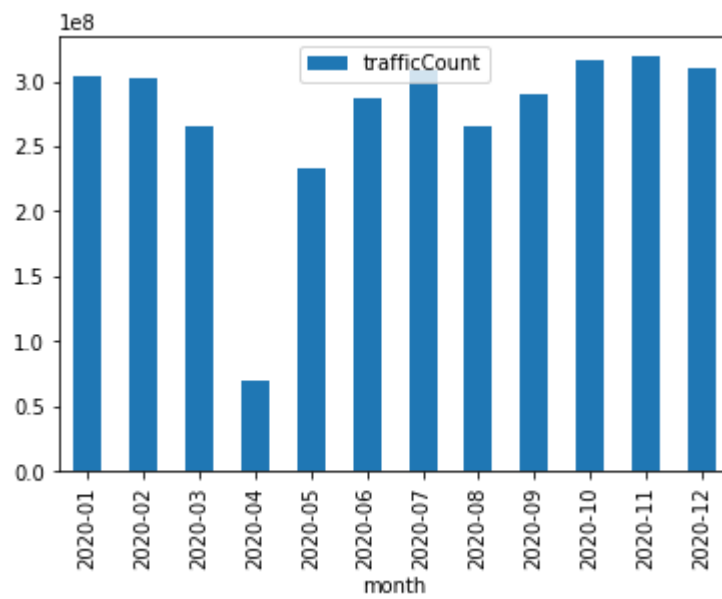
```
In [3]: # SOLUTION

# 1
grouped = df.groupby(['regionName', 'classWeight'])['trafficCount'].sum().unstack()
print(grouped.iloc[:,0].idxmin()) # heavy
print(grouped.iloc[:,1].idxmin()) # light

12 - West Coast
12 - West Coast
```

```
In [4]: # 2
df.groupby("month").agg({"trafficCount": 'sum'}).plot(kind='bar')
```

```
Out[4]: <AxesSubplot:xlabel='month'>
```



In [5]:

```
# 3
df[df.classWeight == 'Heavy'].trafficCount.sum() / df.trafficCount.sum()
```

Out[5]:

0.043097315190595085