my_analysis

February 1, 2022

1 Health Data Notebook

by Jonathan Ferrari, for Data 198 Spring 2022

1.1 Introduction

In this notebook, I will explore and do some basic data analysis of a filed called `health.csv`. This file contains some data about my 2021 year in the context of my health. Let's dive into it!

1.2 Imports

First, we need to make some basic imports so we can look at this data!

[56]:
```
import pandas as pd
import numpy as np
import plotly
import plotly.express as px
import matplotlib
import matplotlib.pyplot as plt
from IPython.display import display
plt.style.use("fivethirtyeight")
```

1.3 The Data

Let's load the data and look at it as a `pd.DataFrame`

[57]:
```
health = pd.read_csv("/Users/jonathanferrari/Downloads/Spring 2022/Data Science␣
↪198 - Instructional Support Seminar/Data198-SP22/data/Health.csv")
display(health)
display(health.describe().iloc[:, 1:3])
```

```
     date month        day     calories     miles
0       1   Jan     Friday   692.881761  3.722718
1       2   Jan   Saturday   536.691408  1.896977
2       3   Jan     Sunday   276.351000  1.696723
3       4   Jan     Monday   270.923147  1.502654
4       5   Jan    Tuesday   397.484041  2.173992
..    …   …          …           …         …
360    27   Dec     Monday     0.164000  1.537434
361    28   Dec    Tuesday   372.944000  1.694043
362    29   Dec  Wednesday  1043.590545  5.737930
363    30   Dec   Thursday   542.194505  2.420097
364    31   Dec     Friday   615.976073  2.928637

[365 rows x 5 columns]

          calories       miles
count   365.000000  365.000000
mean    610.901924    3.397468
std     247.724318    1.892034
min       0.164000    1.020860
25%     434.875116    2.104381
50%     581.203988    2.787204
75%     751.915536    4.196303
max    1625.688000   10.193427
```

1.4 Data Cleaning

I didn't wear my watch some days, but my phone tracked my miles, so I'll filter out the days where my watch wasn't on so our analysis will be more accurate.

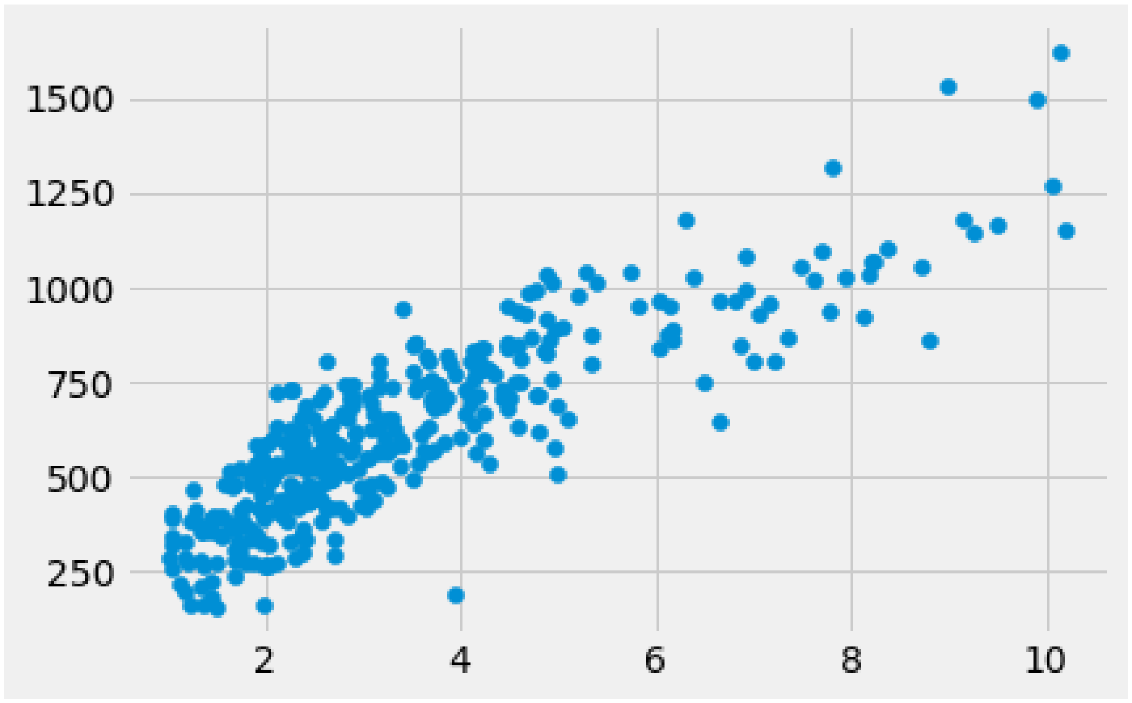
[58]:
```
health = health.query("calories > 100")
```

1.5 Graphs, Graphs, and more Graphs!

Now lets graph this and do some simple linear regression!

1.5.1 `matplotlib` Plots

[59]:
```
plt.scatter(data = health, x = 'miles', y = 'calories');
```



1.5.2 `plotly` Plots

To be frank, I'm not a fan of `matplotlib`, the visualizations are boring! Instead let's look at it using an interactive (and user-friendly) library named `plotly`

[60]:
```
(slope, intercept), r = (np.round(np.polyfit(health['miles'],␣
↪health['calories'], 1), 2),
                         np.round(np.corrcoef(health['miles'],␣
↪health['calories'])[0][1], 4))
px.scatter(data_frame = health, x = 'miles', y = 'calories' , range_y = (0,␣
↪1650),
           trendline = 'ols', title=f"Miles vs. Calories\nTrednline: R^2: {r}␣
↪Calories = {slope}*miles + {intercept}")
```

1.6 Data Visualization Explorations

Lastly, lets get a little fancy and look at the data by `month` or even how far into the year we were! Ya know what, lets do them all!

[61]:
```
def my_year(colorize):
    rep = 'index'
    if type(colorize) == str:
        rep = colorize
    display(px.scatter(data_frame = health, x = 'miles', y = 'calories' ,␣
↪range_y = (0, 1650),
                title=f"Miles vs. Calories [Colored by {rep}]\nTrednline: R^2:␣
↪{r} Calories = {slope}*miles + {intercept}",
               color = colorize))
color_cols = [health.index, 'date', 'month', 'day', 'calories', 'miles']
```

[62]:
```
for group in color_cols:
    my_year(group)
```

1.7 Conclusion

1.7.1 Thats all for now!

Thanks for reading and I hope you enjoyed this demo!

[ ]:
```

```