Creating, Loading and Selecting data with Pandas

June 4, 2020

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
df = pd.DataFrame([
  ['January', 100, 100, 23, 100],
  ['February', 51, 45, 145, 45],
  ['March', 81, 96, 65, 96],
  ['April', 80, 80, 54, 180],
  ['May', 51, 54, 54, 154],
  ['June', 112, 109, 79, 129]],
  columns=['month', 'clinic_east',
           'clinic_north', 'clinic_south',
           'clinic_west']
)
```

[3]:
```
df.head()
```

[3]:

	month	clinic_east	clinic_north	clinic_south	clinic_west
0	January	100	100	23	100
1	February	51	45	145	45
2	March	81	96	65	96
3	April	80	80	54	180
4	May	51	54	54	154

[6]:
```
# Selecting rows
df.iloc[2] #Selecing 2 index row
```

[6]:
```
month           March
clinic_east        81
clinic_north       96
clinic_south       65
clinic_west        96
Name: 2, dtype: object
```

[9]:
```
df.iloc[2, 1] # Third row and second column
```

[9]:
```
81
```

[13]:
```
df.loc[2, "clinic_north"] # Here we can use column name as an index
```

[13]:
```
96
```

[14]:
```
df.loc[0:3, "month":"clinic_south"]
```

[14]:

	month	clinic_east	clinic_north	clinic_south
0	January	100	100	23
1	February	51	45	145
2	March	81	96	65
3	April	80	80	54

[16]:
```
# Selecting multiple rows
df.iloc[3:]
```

[16]:

	month	clinic_east	clinic_north	clinic_south	clinic_west
3	April	80	80	54	180
4	May	51	54	54	154
5	June	112	109	79	129

[18]:
```
# Selecting columns
df['month']
```

[18]:
```
0     January
1    February
2       March
3       April
4         May
5        June
Name: month, dtype: object
```

[19]:
```
# or
df.clinic_west
```

[19]:
```
0    100
1     45
2     96
3    180
4    154
5    129
Name: clinic_west, dtype: int64
```

[20]:
```
# Selecting multiple columns
df[['clinic_north', 'clinic_south']]
```

[20]:

	clinic_north	clinic_south
0	100	23
1	45	145
2	96	65
3	80	54
4	54	54
5	109	79

[22]:
```
# Selecing rows with logic I
df[df.month == 'January']
```

[22]:

	month	clinic_east	clinic_north	clinic_south	clinic_west
0	January	100	100	23	100

[28]:
```
# Select rows with logic II
df[(df.month == 'January') | (df.month == 'March') ]
```

[28]:

	month	clinic_east	clinic_north	clinic_south	clinic_west
0	January	100	100	23	100
2	March	81	96	65	96

[27]:
```
df[(df["month"] == "March") | (df["month"] == "April")]
```

[27]:

	month	clinic_east	clinic_north	clinic_south	clinic_west
2	March	81	96	65	96
3	April	80	80	54	180

[33]:
```
# Select rows with logic III
df[df.month.isin(['January', 'February', 'March'])]
```

[33]:

	month	clinic_east	clinic_north	clinic_south	clinic_west
0	January	100	100	23	100
1	February	51	45	145	45
2	March	81	96	65	96

[37]:
```
# Subset of rows or df.iloc[[0,3,5]]
df2 = df.loc[[0,3,5]]
df2
```

[37]:

	month	clinic_east	clinic_north	clinic_south	clinic_west
0	January	100	100	23	100
3	April	80	80	54	180
5	June	112	109	79	129

[40]:
```
df2.reset_index(inplace = True, drop = True)
df2
```

[40]:

	month	clinic_east	clinic_north	clinic_south	clinic_west
0	January	100	100	23	100
1	April	80	80	54	180
2	June	112	109	79	129

[41]:
```
pwd
```

[41]:
```
'/home/roshan/Desktop/data/Data Manipiulation with Pandas'
```

In this example, you'll be the data analyst for ShoeFly.com, a fictional online shoe store. You've seen this data; now it's your turn to work with it!

[44]:
```
# Load the data from shoefly.csv into the variable orders.
orders = pd.read_csv('shoefly.csv')
orders
```

[44]:

	id	first_name	last_name	email	shoe_type	shoe_material	shoe_color
0	54791	Rebecca	Lindsay	RebeccaLindsay57@hotmail.com	clogs	faux-leather	black
1	53450	Emily	Joyce	EmilyJoyce25@gmail.com	ballet flats	faux-leather	navy
2	91987	Joyce	Waller	Joyce.Waller@gmail.com	sandals	fabric	black
3	14437	Justin	Erickson	Justin.Erickson@outlook.com	clogs	faux-leather	red
4	79357	Andrew	Banks	AB4318@gmail.com	boots	leather	brown
5	52386	Julie	Marsh	JulieMarsh59@gmail.com	sandals	fabric	black
6	20487	Thomas	Jensen	TJ5470@gmail.com	clogs	fabric	navy
7	76971	Janice	Hicks	Janice.Hicks@gmail.com	clogs	faux-leather	navy
8	21586	Gabriel	Porter	GabrielPorter24@gmail.com	clogs		
9	62083	Frances	Palmer	FrancesPalmer50@gmail.com	wedges		
10	91629	Jessica	Hale	JessicaHale25@gmail.com	clogs		
11	98602	Lawrence	Parker	LawrenceParker44@gmail.com	wedges		
12	45832	Susan	Dennis	SusanDennis58@gmail.com	ballet flats		
13	33862	Diane	Ochoa	DO2680@gmail.com	sandals		
14	73431	Rebecca	Charles	Rebecca.Charles@gmail.com	boots		
15	93889	Jacqueline	Crane	JC2072@hotmail.com	wedges		
16	39888	Vincent	Stephenson	VS4753@outlook.com	boots		
17	35961	Roy	Tillman	RoyTillman20@gmail.com	boots		
18	24560	Thomas	Roberson	Thomas.Roberson@gmail.com	wedges		
19	28559	Angela	Newton	ANewton1977@outlook.com	wedges		

```
8        leather      brown
9        leather      white
10       leather        red
11        fabric      brown
12        fabric      white
13        fabric        red
14  faux-leather      white
15        fabric        red
16       leather      black
17       leather      white
18        fabric        red
19        fabric        red
```

[46]:
```
# Inspect the first 5 lines of the data
orders.head()
```

[46]:
```
      id first_name last_name                         email     shoe_type  \
0  54791    Rebecca   Lindsay  RebeccaLindsay57@hotmail.com         clogs
1  53450      Emily     Joyce        EmilyJoyce25@gmail.com  ballet flats
2  91987      Joyce    Waller        Joyce.Waller@gmail.com       sandals
3  14437     Justin  Erickson   Justin.Erickson@outlook.com         clogs
4  79357     Andrew     Banks               AB4318@gmail.com         boots

  shoe_material shoe_color
0  faux-leather      black
1  faux-leather       navy
2        fabric      black
3  faux-leather        red
4       leather      brown
```

[48]:
```
# Your marketing department wants to send out an email blast to everyone who␣
 ↪ordered shoes!
# Select all of the email addresses from the column email and save them to a␣
 ↪variable called emails.
emails = orders.email
emails
```

[48]:
```
0     RebeccaLindsay57@hotmail.com
1           EmilyJoyce25@gmail.com
2           Joyce.Waller@gmail.com
3      Justin.Erickson@outlook.com
4                AB4318@gmail.com
5           JulieMarsh59@gmail.com
6                TJ5470@gmail.com
7           Janice.Hicks@gmail.com
8        GabrielPorter24@gmail.com
9        FrancesPalmer50@gmail.com
```

```
10         JessicaHale25@gmail.com
11      LawrenceParker44@gmail.com
12         SusanDennis58@gmail.com
13                DO2680@gmail.com
14       Rebecca.Charles@gmail.com
15              JC2072@hotmail.com
16              VS4753@outlook.com
17          RoyTillman20@gmail.com
18       Thomas.Roberson@gmail.com
19         ANewton1977@outlook.com
Name: email, dtype: object
```

[51]:
```
# Frances Palmer claims that her order was wrong. What did Frances Palmer order?
# Use logic to select that row of orders and save it to the variable␣
↪frances_palmer.
frances_palmer = orders[(orders.first_name == "Frances") & (orders.last_name ==␣
↪"Palmer")]
frances_palmer
```

[51]:
```
      id first_name last_name                      email shoe_type  \
9  62083    Frances    Palmer  FrancesPalmer50@gmail.com    wedges

  shoe_material shoe_color
9       leather      white
```

[56]:
```
# We need some customer reviews for our comfortable shoes. Select all orders␣
↪for shoe_type:
# clogs, boots, and ballet flats and save them to the variable comfy_shoes.
comfy_shoes = orders[orders.shoe_type.isin(["clogs", "boots", "ballet flats"])]
```

[57]:
```
comfy_shoes
```

[57]:
```
       id first_name   last_name                        email     shoe_type  \
0   54791    Rebecca     Lindsay  RebeccaLindsay57@hotmail.com         clogs
1   53450      Emily       Joyce        EmilyJoyce25@gmail.com  ballet flats
3   14437     Justin    Erickson   Justin.Erickson@outlook.com         clogs
4   79357     Andrew       Banks              AB4318@gmail.com         boots
6   20487     Thomas      Jensen              TJ5470@gmail.com         clogs
7   76971     Janice       Hicks        Janice.Hicks@gmail.com         clogs
8   21586    Gabriel      Porter     GabrielPorter24@gmail.com         clogs
10  91629    Jessica        Hale       JessicaHale25@gmail.com         clogs
12  45832      Susan      Dennis       SusanDennis58@gmail.com  ballet flats
14  73431    Rebecca     Charles     Rebecca.Charles@gmail.com         boots
16  39888    Vincent  Stephenson            VS4753@outlook.com         boots
17  35961        Roy     Tillman        RoyTillman20@gmail.com         boots

   shoe_material shoe_color
```

```
0   faux-leather      black
1   faux-leather       navy
3   faux-leather        red
4        leather      brown
6         fabric       navy
7   faux-leather       navy
8        leather      brown
10       leather        red
12        fabric      white
14  faux-leather      white
16       leather      black
17       leather      white
```

[59]:
```
clinic_df = pd.DataFrame([
  ['January', 100, 100, 23, 100],
  ['February', 51, 45, 145, 45],
  ['March', 81, 96, 65, 96],
  ['April', 80, 80, 54, 180],
  ['May', 51, 54, 54, 154],
  ['June', 112, 109, 79, 129]],
  columns=['month', 'clinic_east',
           'clinic_north', 'clinic_south',
           'clinic_west'])
```

[60]:
```
clinic_df.head()
```

[60]:

	month	clinic_east	clinic_north	clinic_south	clinic_west
0	January	100	100	23	100
1	February	51	45	145	45
2	March	81	96	65	96
3	April	80	80	54	180
4	May	51	54	54	154

[62]:
```
# If you wanted to select the row including all of the data for the month of␣
 ↪May, which of the following lines of code would you use?
```

[64]:
```
clinic_df.loc[4]
```

[64]:
```
month           May
clinic_east      51
clinic_north     54
clinic_south     54
clinic_west     154
Name: 4, dtype: object
```

[65]:
```
clinic_df[clinic_df.month == 'May']
```

[65]:

	month	clinic_east	clinic_north	clinic_south	clinic_west
4	May	51	54	54	154

[66]:
```
customers = pd.DataFrame([
  ['Jesse Sternberg', '193 6th Avenue', 31],
  ['Amy Lauder', '546 Marblehead Way', 43],
  ['Gerri Sanderson', '65 New York Street', 35],
  ['Austin Barnes', '2888 North Ogden Avenue', 28]],
  columns = ['name', 'address', 'age'])
```

[67]:
```
customers
```

[67]:

	name	address	age
0	Jesse Sternberg	193 6th Avenue	31
1	Amy Lauder	546 Marblehead Way	43
2	Gerri Sanderson	65 New York Street	35
3	Austin Barnes	2888 North Ogden Avenue	28

[69]:
```
customers.age
```

[69]:
```
0    31
1    43
2    35
3    28
Name: age, dtype: int64
```

[ ]:
```
```