Professur Technische Informatik
Prof. Dr. Wolfram Hardt

Programming with pandas

Exercise 1.8

Shadi Saleh

Agenda

- Pandas
- Data frame
- Group by
- merge
- join

Data Frame methods

Unlike attributes, python methods have *parenthesis.*
All attributes and methods can be listed with a *dir()* function: **dir(df)**

df.method()	description
head([n]), tail([n])	first/last n rows
describe()	generate descriptive statistics (for numeric columns only)
max(), min()	return max/min values for all numeric columns
mean(), median()	return mean/median values for all numeric columns
std()	standard deviation
sample([n])	returns a random sample of the data frame
dropna()	drop all the records with missing values

Constructing data from: numpy ndarray

```
df2 = pd.DataFrame(np.array([[1, 2, 3],
                             [4, 5, 6],
                             [7, 8, 9]]),
                   columns=['a', 'b', 'c'])
```

Values from np array

Column names

Data Frames groupby method

- Using “group by” method we can:
  - Split the data into groups based on some criteria
  - Calculate statistics (or apply a function) to each group
  - Similar to dplyr() function in R
- *#Group data using rank*
- df_rank = df.groupby(['rank'])
- df_rank.mean()

rank	phd	service	salary
AssocProf	15.076923	11.307692	91786.230769
AsstProf	5.052632	2.210526	81362.789474
Prof	27.065217	21.413043	123624.804348

Group by

- Once groupby object is create we can calculate various statistics for each group:
- *#Calculate mean salary for each professor rank:*
- df.groupby('rank')[['salary']].mean()

rank	salary
AssocProf	91786.230769
AsstProf	81362.789474
Prof	123624.804348

Data Frame: filtering

- To subset the data we can apply Boolean indexing. This indexing is commonly known as a filter. For example if we want to subset the rows in which the salary value is greater than $120K:
- *#Calculate mean salary for each professor rank:*
- df_sub = df[df['salary'] > 120000]
- Any Boolean operator can be used to subset the data:
  - > greater; >= greater or equal;
  - < less; <= less or equal;
  - == equal; != not equal;
- *#Select only those rows that contain female professors:*
- df_f = df[df['sex'] == 'Female']

Data Frames: Slicing

- There are a number of ways to subset the Data Frame:
  - one or more columns
  - one or more rows
  - a subset of rows and columns

Rows and columns can be selected by their position or label

- When we need to select more than one column and/or make the output to be a DataFrame, we should use double brackets:
- *#Select column salary:*
- df[['rank','salary']]

Method loc and iloc

- If we need to select a range of rows, using their labels we can use method loc:

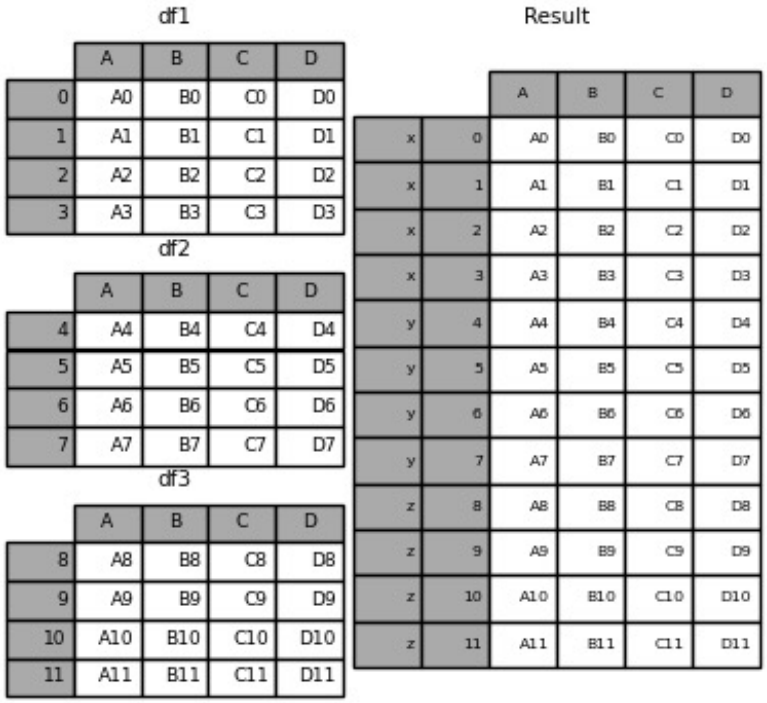
```
#Select rows by their labels:
df_sub.loc[10:20,['rank','sex','salary']]
```

- If we need to select a range of rows and/or columns, using their positions we can use method iloc:

```
#Select rows by their labels:
df_sub.iloc[10:20,[0, 3, 4, 5]]
```

Merge

- Merge is a **function in the pandas namespace**, and it is also available as a DataFrame instance method merge(), with the calling DataFrame being implicitly considered the left object in the join. (s)-on-index join.



df1

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

df2

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

df3

	A	B	C	D
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

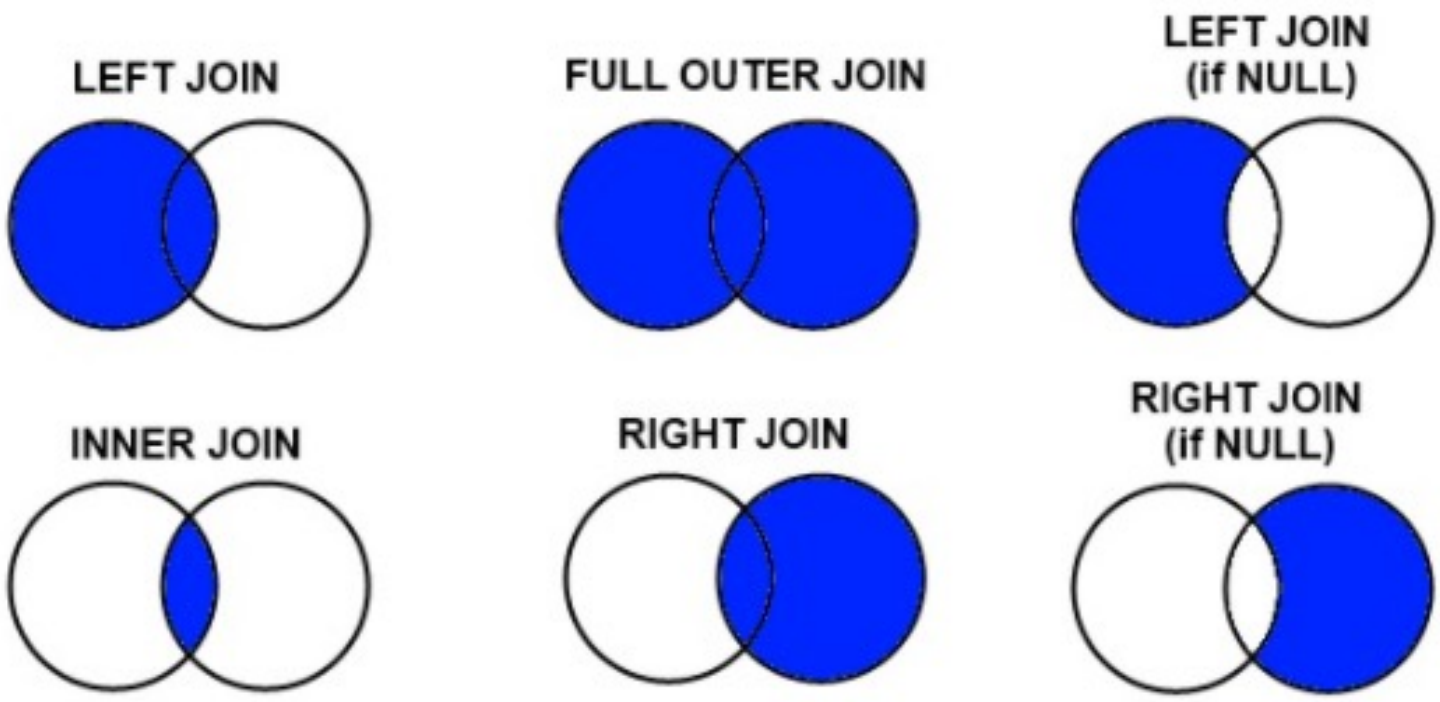
Result

		A	B	C	D
x	0	A0	B0	C0	D0
x	1	A1	B1	C1	D1
x	2	A2	B2	C2	D2
x	3	A3	B3	C3	D3
y	4	A4	B4	C4	D4
y	5	A5	B5	C5	D5
y	6	A6	B6	C6	D6
y	7	A7	B7	C7	D7
z	8	A8	B8	C8	D8
z	9	A9	B9	C9	D9
z	10	A10	B10	C10	D10
z	11	A11	B11	C11	D11

Merge Types

- 1. Inner Merge / Inner join – The default Pandas behaviour, only keep rows where the merge “on” value exists in both the left and right dataframes.
- 2. Left Merge / Left outer join – (aka left merge or left join) Keep every row in the left dataframe. Where there are missing values of the “on” variable in the right dataframe, add empty / NaN values in the result.
- 3. Right Merge / Right outer join – (aka right merge or right join) Keep every row in the right dataframe. Where there are missing values of the “on” variable in the left column, add empty / NaN values in the result.
- 4. Outer Merge / Full outer join – A full outer join returns all the rows from the left dataframe, all the rows from the right dataframe, and matches up rows where possible, with NaNs elsewhere.

Different merge and join types



Merge/Join types as used in Pandas, R, SQL, and other data-orientated languages and libraries. Source: Stack Overflow.

Merge

Left data frame

Right data frame

Left data frame

Only 3 columns

Common column between left and right

```
In [13]: result = pd.merge(user_usage,
                           user_device[['use_id', 'platform', 'device']],
                           on='use_id')
         result.head()
```

Out[13]:

	outgoing_mins_per_month	outgoing_sms_per_month	monthly_mb	use_id	platform	device
0	21.97	4.82	1557.33	22787	android	GT-I9505
1	1710.08	136.88	7267.55	22788	android	SM-G930F
2	1710.08	136.88	7267.55	22789	android	SM-G930F
3	94.46	35.17	519.12	22790	android	D2303
4	71.59	79.26	1557.33	22792	android	SM-G361F

Data Frames: Sorting

- We can sort the data by a value in the column. By default the sorting will occur in ascending order and a new data frame is return.

```
# Create a new data frame from the original sorted by the column Salary
df_sorted = df.sort_values( by ='service')
df_sorted.head()
```

	rank	discipline	phd	service	sex	salary
55	AsstProf	A	2	0	Female	72500
23	AsstProf	A	2	0	Male	85000
43	AsstProf	B	5	0	Female	77000
17	AsstProf	B	4	0	Male	92000
12	AsstProf	B	1	0	Male	88000

Data Frames: Sorting

- We can sort the data using 2 or more columns:
- df_sorted = df.sort_values(by =['service', 'salary'], ascending = [True, False])
- df_sorted.head(10)

	rank	discipline	phd	service	sex	salary
52	Prof	A	12	0	Female	105000
17	AsstProf	B	4	0	Male	92000
12	AsstProf	B	1	0	Male	88000
23	AsstProf	A	2	0	Male	85000
43	AsstProf	B	5	0	Female	77000
55	AsstProf	A	2	0	Female	72500
57	AsstProf	A	3	1	Female	72500
28	AsstProf	B	7	2	Male	91300
42	AsstProf	B	4	2	Female	80225
68	AsstProf	A	4	2	Female	77500

Missing Values

- # Select the rows that have at least one missing value
- flights[flights.isnull().any(axis=1)].head()

	year	month	day	dep_time	dep_delay	arr_time	arr_delay	carrier	tailnum	flight	origin	dest	air_time	distance	hour	minute
330	2013	1	1	1807.0	29.0	2251.0	NaN	UA	N31412	1228	EWR	SAN	NaN	2425	18.0	7.0
403	2013	1	1	NaN	NaN	NaN	NaN	AA	N3EHAA	791	LGA	DFW	NaN	1389	NaN	NaN
404	2013	1	1	NaN	NaN	NaN	NaN	AA	N3EVAA	1925	LGA	MIA	NaN	1096	NaN	NaN
855	2013	1	2	2145.0	16.0	NaN	NaN	UA	N12221	1299	EWR	RSW	NaN	1068	21.0	45.0
858	2013	1	2	NaN	NaN	NaN	NaN	AA	NaN	133	JFK	LAX	NaN	2475	NaN	NaN

Missing Values

- There are a number of methods to deal with missing values in the data frame:

df.method()	Description
dropna()	Drop missing observations
dropna(how='all')	Drop observations where all cells is NA
dropna(axis=1, how='all')	Drop column if all the values are missing
dropna(thresh = 5)	Drop rows that contain less than 5 non-missing values
fillna(0)	Replace missing values with zeros
isnull()	returns True if the value is missing
notnull()	Returns True for non-missing values

Missing Values

- When summing the data, missing values will be treated as zero
- If all values are missing, the sum will be equal to NaN
- cumsum() and cumprod() methods ignore missing values but preserve them in the resulting arrays
- Missing values in GroupBy method are excluded (just like in R)
- Many descriptive statistics methods have *skipna* option to control if missing data should be excluded . This value is set to *True* by default (unlike R)

Graphics

Function	Description
distplot	histogram
barplot	estimate of central tendency for a numeric variable
violinplot	similar to boxplot, also shows the probability density of the data
jointplot	Scatterplot
regplot	Regression plot
pairplot	Pairplot
boxplot	boxplot
swarmplot	categorical scatterplot
factorplot	General categorical plot

Exercise 1

- Write a Pandas program to display a summary of the basic information about a specified DataFrame and its data.
- *Sample DataFrame*:
  exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
  'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
  'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
  'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
  labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Exercise 2

- Write a Pandas program to select the rows where the number of attempts in the examination is greater than 2.

  Sample DataFrame:
  exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
  'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
  'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
  'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
  labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Thank you for your attention

Shadi Saleh

Professur Technische Informatik
Fakultät für Informatik

Technische Universität Chemnitz
Straße der Nationen 62
09111 Chemnitz
Germany

shadi.saleh@informatik.tu-chemnitz.de