ICDSS
Imperial College Data Science Society

Python 102

Titanic Challenge with Python

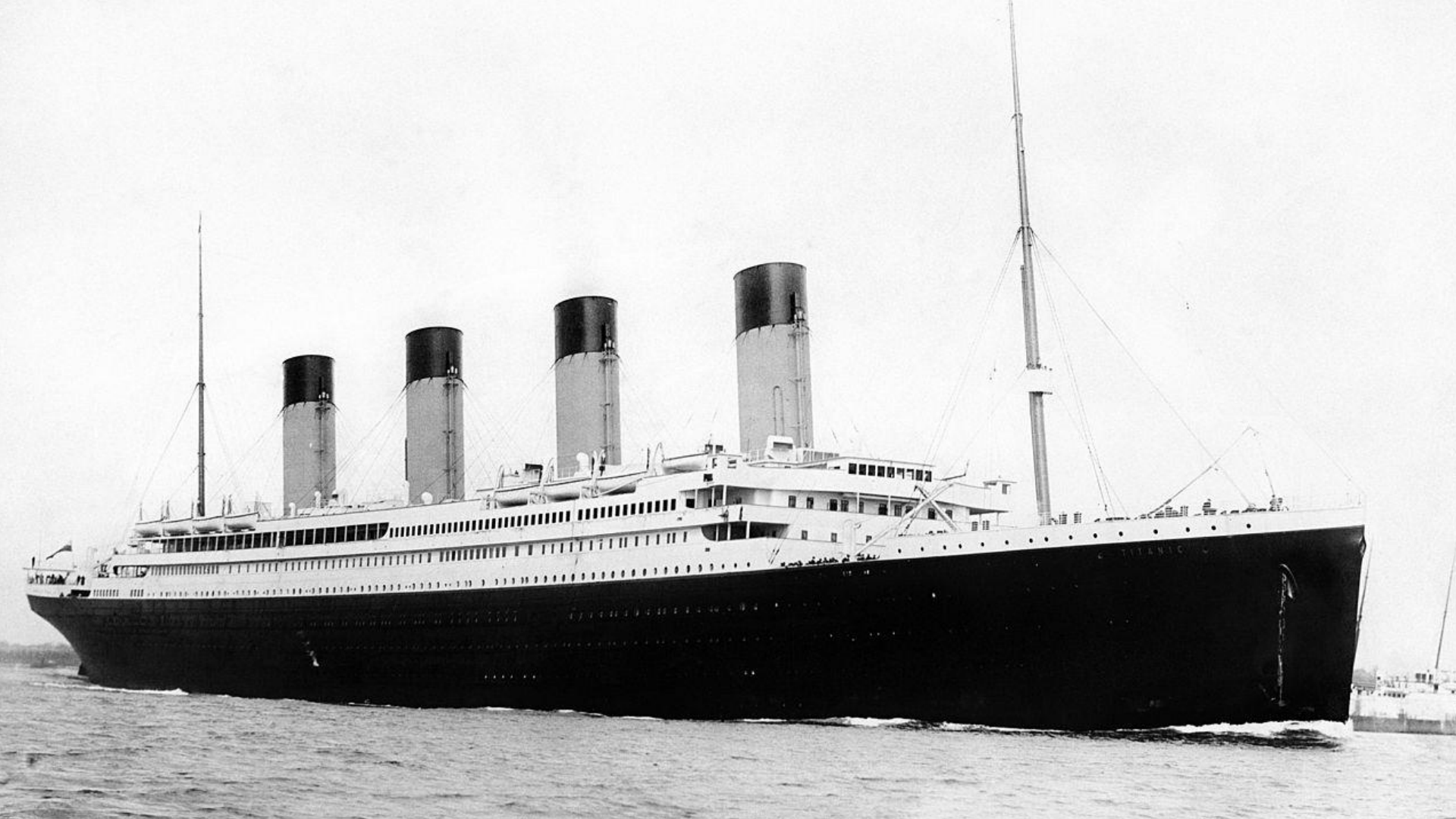
Kane Wu
Email: kcw115@ic.ac.uk

Agenda

- The Titanic Data
- Reading data into Python (Review)
- Transforming the Data
- Visualizing the Data

1

Titanic Data



Titanic Questions

- Who were the Titanic Passengers?
- Where were our passenger's cabins?
- How much did they pay for their tickets?
- What made people survive the sinking?

The Titanic Data

A1 | PassengerId

	A	B	C	D	E	F	G	H	I	J	K	L
1	PassengerI	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, M	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, I	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen,	female	26	0	0	STON/O2.	7.925		S
5	4	1	1	Futrelle, M	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr.	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, M	male		0	0	330877	8.4583		Q
8	7	0	1	McCarthy,	male	54	0	0	17463	51.8625	E46	S
9	8	0	3	Palsson, M	male	2	3	1	349909	21.075		S
10	9	1	3	Johnson, I	female	27	0	2	347742	11.1333		S
11	10	1	2	Nasser, M	female	14	1	0	237736	30.0708		C
12	11	1	3	Sandstron	female	4	1	1	PP 9549	16.7	G6	S
13	12	1	1	Bonnell, M	female	58	0	0	113783	26.55	C103	S
14	13	0	3	Saunderc	male	20	0	0	A/5. 2151	8.05		S
15	14	0	3	Anderssor	male	39	1	5	347082	31.275		S
16	15	0	3	Vestrom, I	female	14	0	0	350406	7.8542		S
17	16	1	2	Hewlett, M	female	55	0	0	248706	16		S
18	17	0	3	Rice, Mast	male	2	4	1	382652	29.125		Q

The Titanic Data

Every row represents a passenger in Titanic

PassengerId	ID assigned to passenger
Survived	Whether the person survived the crash. 1 = survived, 0 = deceased
Pclass	What class deck the passenger was staying at
Name	Name of passenger
Sex	Gender of Passenger
Age	Age of passenger
SibSp	Number of siblings and spouses the passenger has on board

The Titanic Data

Parch	Number of parents and children the passenger has on board
Ticket	Ticket number of the passenger.
Fare	The fare the passenger paid for
Cabin	Which cabin the passenger was in.
Embarked	Where the passenger boarded the Titanic.

Slide on Pandas

Data Structures - DataFrames

Data Analysis – info, describe, head

Data Munging – loc, apply

Data Reading – read_csv

Data Writing – to_csv

Handling Missing Data – fillna, dropna

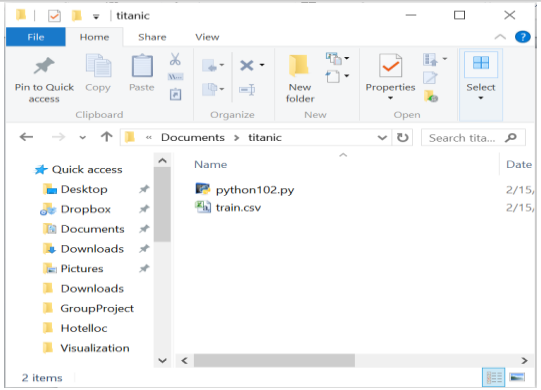
Merging and Joining Data - merge, concats

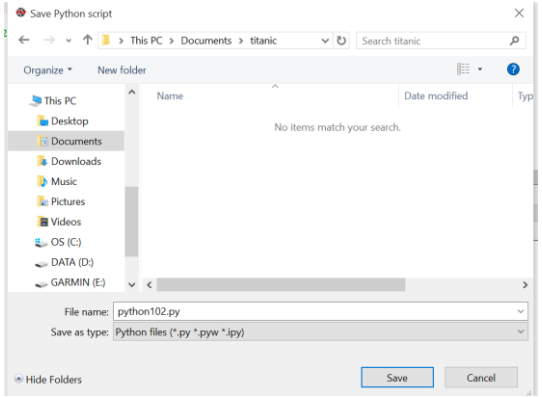
2

Reading Data Into Python

Reading Data Into Python

- We will be using pandas, the python package to read the file.
- import pandas as pd
- df = pd.read_csv("titanic.csv")
- df means dataframe. You can name it anything you want!!





Inclass Exercises

- Reading CSV
  - Read the titanic CSV file.
  - Run the command head() on the file
  - Run the command describe() on the file
  - Run the command info() on the file
  - Print out row 58 from the file.
  - Find the sum of all the fare

Understanding the data

- Age column has a count of 714
- All other columns has 891
- Missing values in the age count
- Print df[“Age”]

```
In [8]: print df["Age"]
0      22
1      38
2      26
3      35
4      35
5     NaN
6      54
7       2
8      27
9      14
10      4
11     58
12     20
13     39
14     14
15     55
16      2
17    NaN
18     31
19    NaN
20     35
21     34
22     15
23     28
24      8
25     38
26    NaN
27     19
28    NaN
29    NaN
```

4

Transforming the Data

Transforming the data

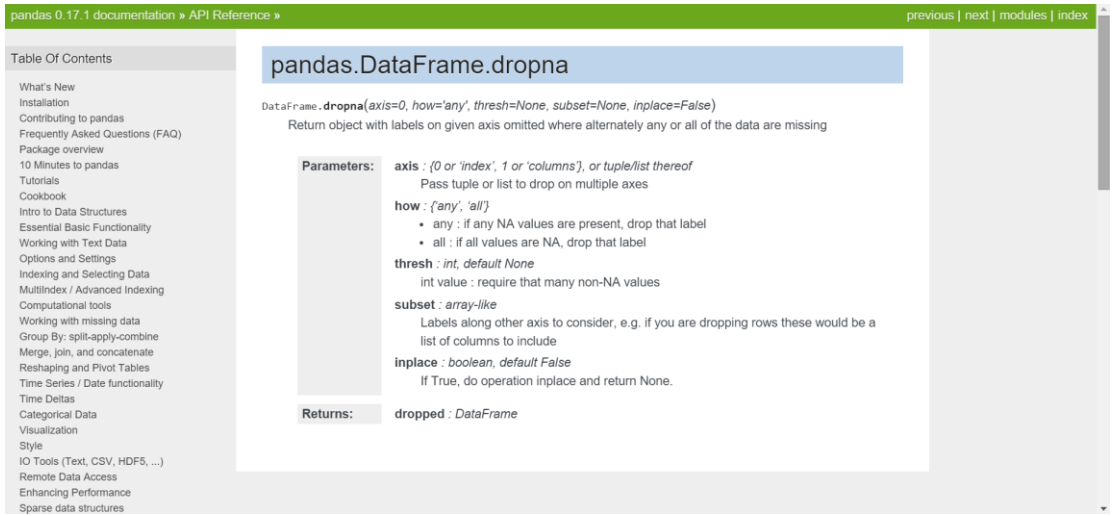
- dropna – a function that drop missing values

```
In [69]: df2 = df

In [70]: df2.head()
Out[70]:
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex  Age  SibSp  \
0                            Braund, Mr. Owen Harris    male   22      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female   38      1
2                             Heikkinen, Miss. Laina  female   26      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female   35      1
4                           Allen, Mr. William Henry    male   35      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

```
In [71]: df2 = df2.dropna()

In [72]: df2.head()
Out[72]:
    PassengerId  Survived  Pclass  \
1             2         1       1
3             4         1       1
6             7         0       1
10           11         1       3
11           12         1       1

                                                 Name     Sex  Age  SibSp  \
1   Cumings, Mrs. John Bradley (Florence Briggs Th...  female   38      1
3        Futrelle, Mrs. Jacques Heath (Lily May Peel)  female   35      1
6                             McCarthy, Mr. Timothy J    male   54      0
10                   Sandstrom, Miss. Marguerite Rut  female    4      1
11                          Bonnell, Miss. Elizabeth  female   58      0

    Parch    Ticket     Fare Cabin Embarked
1       0  PC 17599  71.2833   C85        C
3       0    113803  53.1000  C123        S
6       0     17463  51.8625   E46        S
10      1   PP 9549  16.7000    G6        S
11      0    113783  26.5500  C103        S
```

Transforming the data

- dropna(subset = ['Age'])
- This drops all the null values in the file



pandas 0.17.1 documentation » API Reference »

previous | next | modules | index

Table Of Contents

pandas.DataFrame.dropna

`DataFrame.`**dropna**(*axis=0, how='any', thresh=None, subset=None, inplace=False*)

Return object with labels on given axis omitted where alternately any or all of the data are missing

Parameters:

axis : *{0 or 'index', 1 or 'columns'}, or tuple/list thereof*
Pass tuple or list to drop on multiple axes

how : *{'any', 'all'}*
- any : if any NA values are present, drop that label
- all : if all values are NA, drop that label

thresh : *int, default None*
int value : require that many non-NA values

subset : *array-like*
Labels along other axis to consider, e.g. if you are dropping rows these would be a list of columns to include

inplace : *boolean, default False*
If True, do operation inplace and return None.

Returns:

dropped : *DataFrame*

Transforming the data

- Pandas loc function selects the data you want
- notnull() returns rows that does not have null
- df4 = df4.loc[df4.Age.notnull()]

```
In [78]: print df4.head(7)
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3
6            7         0       1
7            8         0       3

                                                Name     Sex  Age  SibSp  \
0                            Braund, Mr. Owen Harris    male   22      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female   38      1
2                             Heikkinen, Miss. Laina  female   26      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female   35      1
4                           Allen, Mr. William Henry    male   35      0
6                            McCarthy, Mr. Timothy J    male   54      0
7                     Palsson, Master. Gosta Leonard    male    2      3

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
6      0             17463  51.8625   E46        S
7      1            349909  21.0750   NaN        S
```

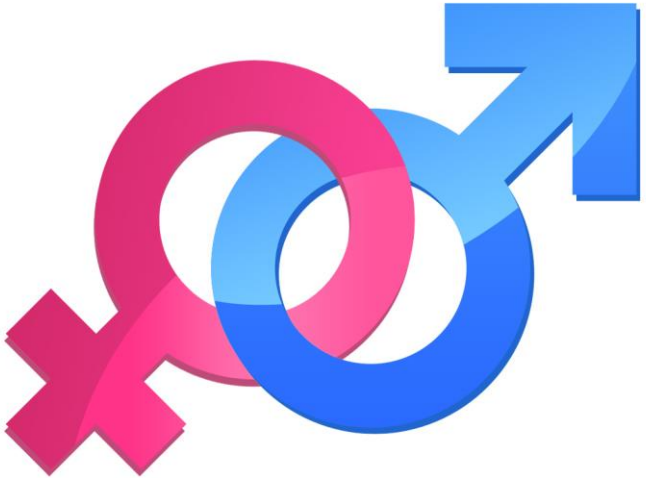
Transforming the data

- fillna – a function that replace missing values with things you want
- df[“Age”].median() = returns the median of the columns
- df[“Age”] = df[“Age”].fillna(df["Age"].median())

```
In [15]: print df["Age"].median()
28.0

In [16]: df["Age"] = df["Age"].fillna(df["Age"].median())

In [17]: print df["Age"]
0      22
1      38
2      26
3      35
4      35
5      28
6      54
7       2
8      27
9      14
10      4
11     58
12     20
13     39
14     14
15     55
16      2
17     28
18     31
19     28
20     35
21     34
```

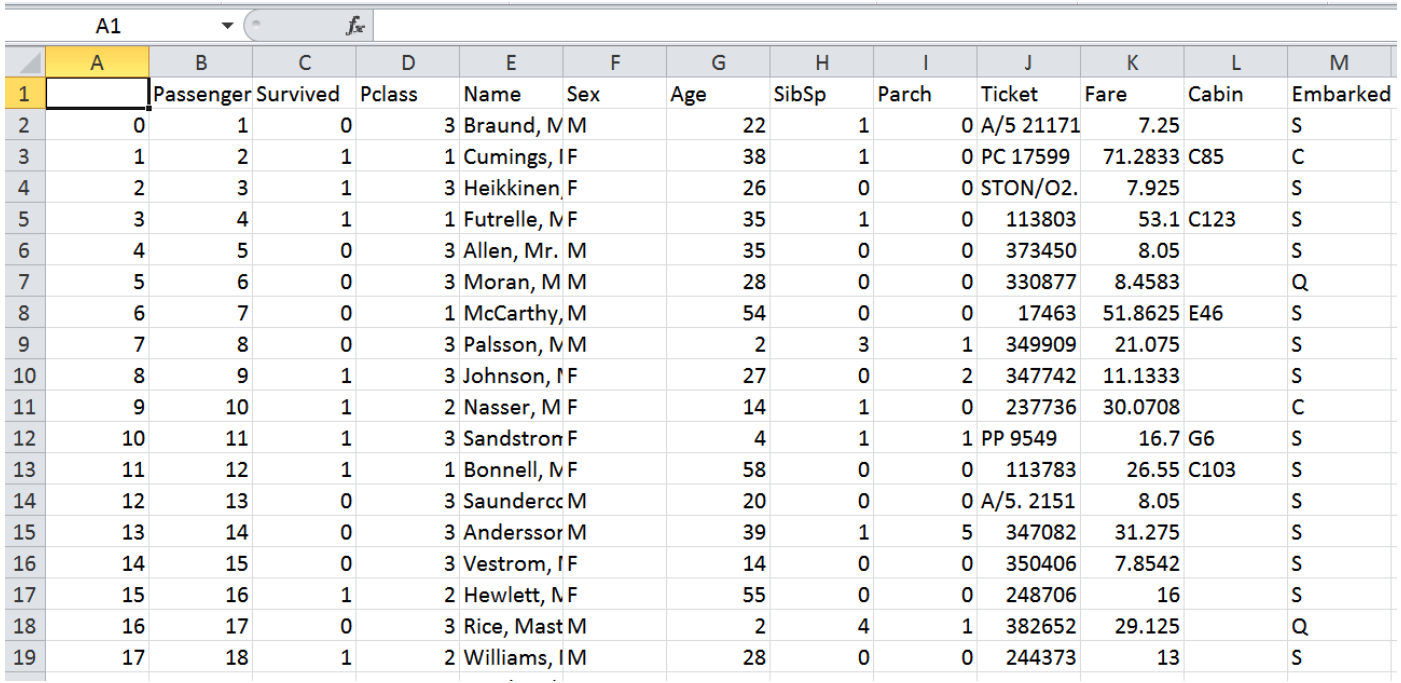
Transforming the data

- Now we want to change all the values a column to something else
- Male into M
- Female into F



Writing the Data

- df.to_csv('out.csv.')



A1

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		Passenger	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	0	1	0	3	Braund, M	M	22	1	0	A/5 21171	7.25		S
3	1	2	1	1	Cumings, I	F	38	1	0	PC 17599	71.2833	C85	C
4	2	3	1	3	Heikkinen,	F	26	0	0	STON/O2.	7.925		S
5	3	4	1	1	Futrelle, M	F	35	1	0	113803	53.1	C123	S
6	4	5	0	3	Allen, Mr.	M	35	0	0	373450	8.05		S
7	5	6	0	3	Moran, M	M	28	0	0	330877	8.4583		Q
8	6	7	0	1	McCarthy,	M	54	0	0	17463	51.8625	E46	S
9	7	8	0	3	Palsson, M	M	2	3	1	349909	21.075		S
10	8	9	1	3	Johnson, I	F	27	0	2	347742	11.1333		S
11	9	10	1	2	Nasser, M	F	14	1	0	237736	30.0708		C
12	10	11	1	3	Sandstron	F	4	1	1	PP 9549	16.7	G6	S
13	11	12	1	1	Bonnell, M	F	58	0	0	113783	26.55	C103	S
14	12	13	0	3	Saundercc	M	20	0	0	A/5. 2151	8.05		S
15	13	14	0	3	Anderssor	M	39	1	5	347082	31.275		S
16	14	15	0	3	Vestrom, I	F	14	0	0	350406	7.8542		S
17	15	16	1	2	Hewlett, M	F	55	0	0	248706	16		S
18	16	17	0	3	Rice, Mast	M	2	4	1	382652	29.125		Q
19	17	18	1	2	Williams, I	M	28	0	0	244373	13		S

Inclass Exercise

Inclass Exercises

- Do the same for Embarked
  - Replace the missing value in the embarked column with S
  - Assign the code “Southampton” to S
  - Assign the code “Cherbourg” to C
  - Assign the code “Queenstown” to Q
  - Write the file to “embarked.csv”

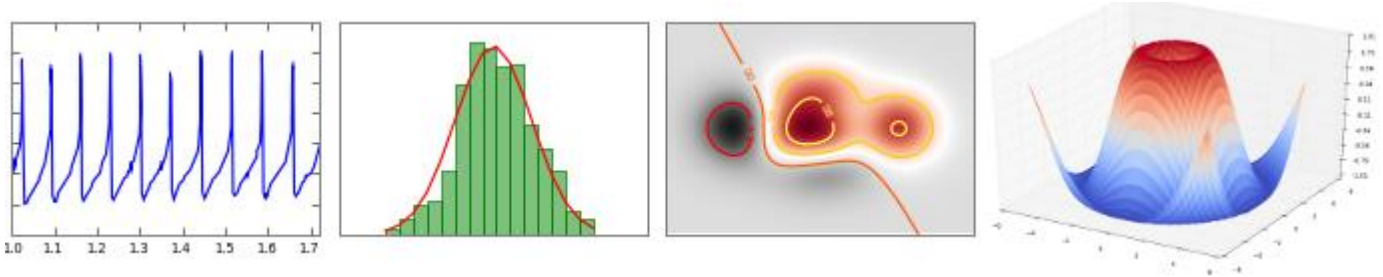
5

Visualizing the data

(and more transformations!!)

Seaborn

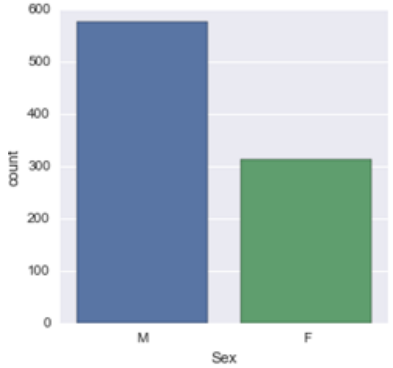
Visualization library based on matplotlib
Visualization more appealing
Complicated plots simple to create
Integrate well with pandas

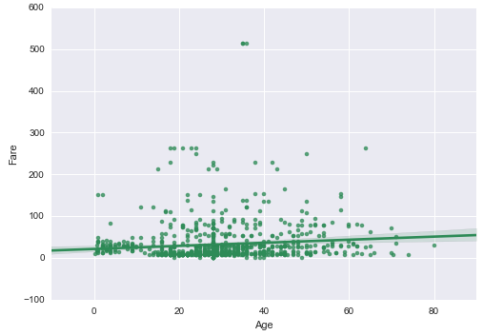


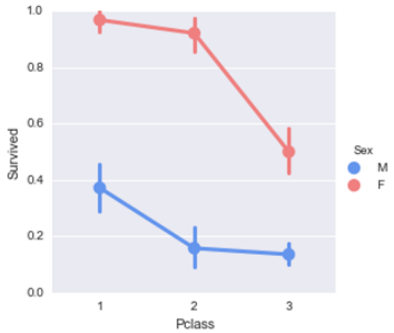
Seaborn

FactorPlot

- A factor plot is simply the same plot generated for different factor variables
- Default is a point plot
- kind : {point, bar, count, box, violin, strip}







Who were the Titanic Passengers?

Inclass Exercises

- Make a factorplot based on where the individuals embarked from
- Make a factor plot based on where the individual embarked from but group it by Pclass using colors.

What about the children?

Pandas Apply Function

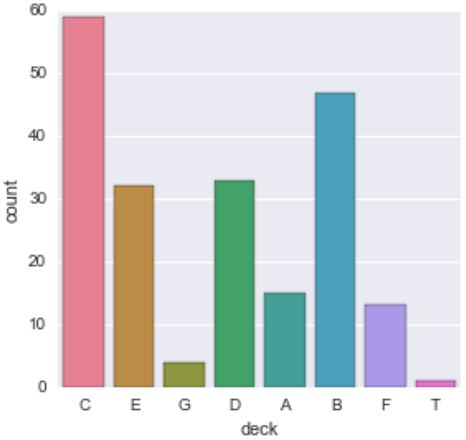
- Applies function along input axis of DataFrame.

```
In [104]: df[['Age','Sex']]
Out[104]:
     Age Sex
0     22   M
1     38   F
2     26   F
3     35   F
4     35   M
5    NaN   M
6     54   M
7      2   M
8     27   F
9     14   F
10     4   F
11    58   F
12    20   M
13    39   M
14    14   F
15    55   F
16     2   M
17   NaN   M
18    31   F
19   NaN   F
20    35   M
```

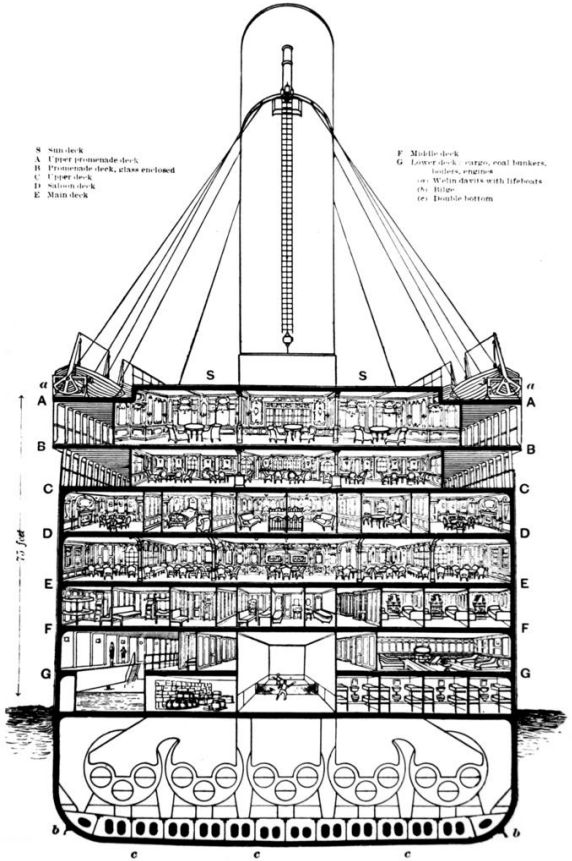

```
In [92]: df.who
Out[92]:
0      M
1      F
2      F
3      F
4      M
5      M
6      M
7      C
8      F
9      C
10     C
11     F
12     M
13     M
14     C
15     F
16     C
17     M
18     F
19     F
20     M
```

Inclass Exercises

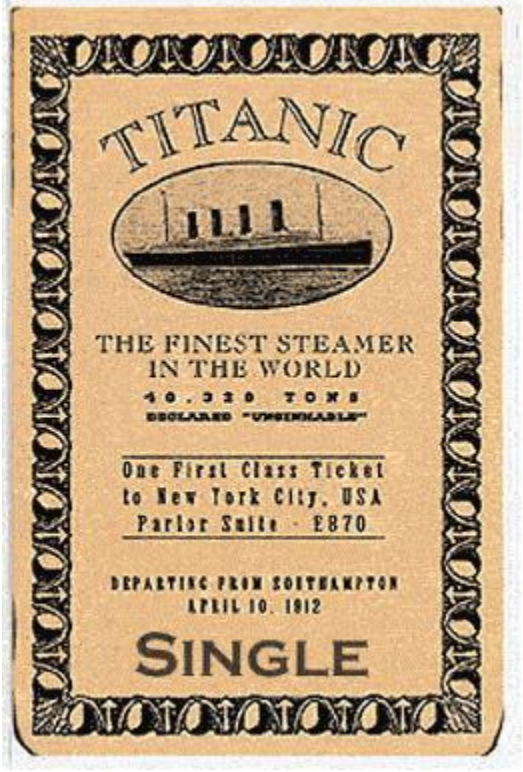
- Use df.loc to find all rows for people older than 60
- Use apply and update the who column to label people who are older than 60 with E ('Elderly')
- Plot the graph we did earlier with elderly. Assign the color yellow for elderly.
- Using the apply function create a column called wealthy_old that will have people that paid a fare of more than 30 and who are older than 50 labeled as rich and old. Otherwise will be labeled normal.




```
In [140]: df2.loc[df2.wealthy_old == 'Rich and Old']
Out[140]:
     Age      Fare   wealthy_old
6     54   51.8625  Rich and Old
54    65   61.9792  Rich and Old
124   54   77.2875  Rich and Old
155   51   61.3792  Rich and Old
195   58  146.5208  Rich and Old
262   52   79.6500  Rich and Old
268   58  153.4625  Rich and Old
275   63   77.9583  Rich and Old
366   60   75.2500  Rich and Old
438   64  263.0000  Rich and Old
```



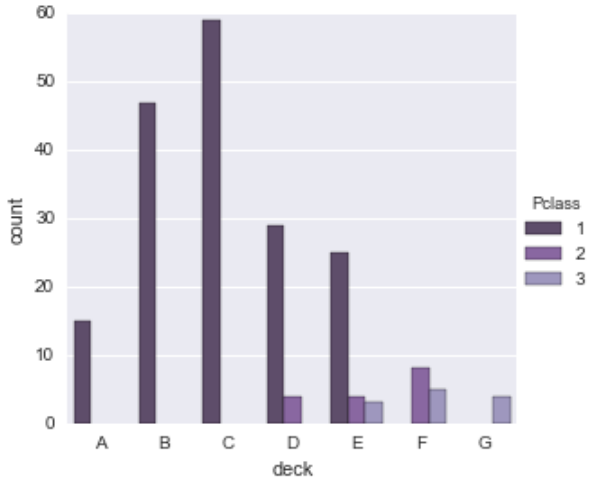
Where were our passenger's cabins?

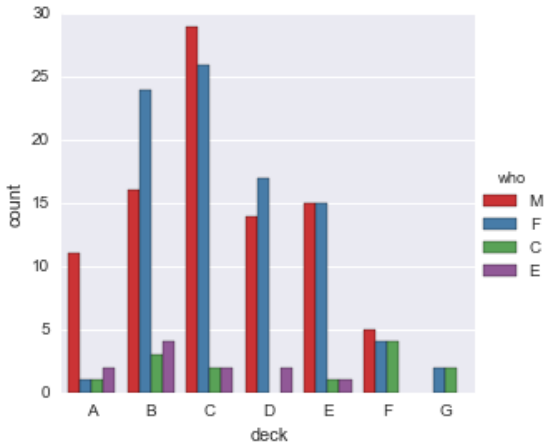


How much did they pay for their tickets?

Inclass Exercises

- Make a factorplot with deck on x axis and Age on the yaxis
- Check if the deck B bar is correct by using describe()
- Plot the graph of each deck with the class they belong in
- Plot the graph of each deck with the who column using a different palette





What made people survive the sinking?

Inclass Exercises

- How do we visualize those who survived based on the who column?
- How do we visualize those who survived based on the deck column?
- How do we visualize if the rich and old survived?

Thanks!

Any questions ?

You can find me at

- kcw115@ic.ac.uk

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash and Vinsionaire