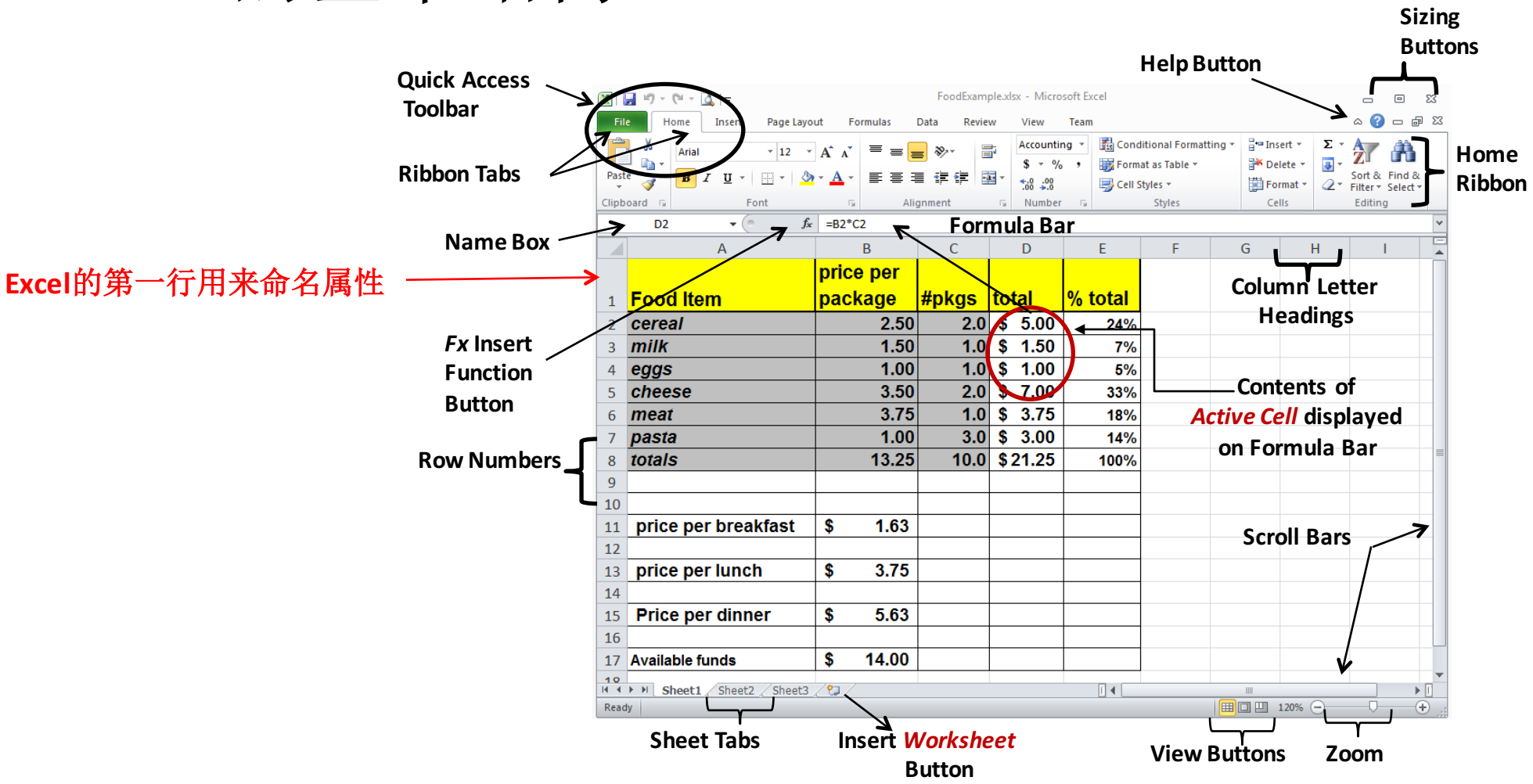
实用python编程 第5讲

Excel数据分析

2017-10-30

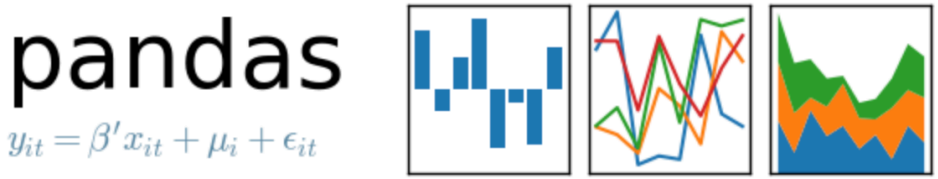
Excel的基本结构



http://web.cse.ohio-state.edu/cse1111/AUGUST%202013/Slides/

Pandas

- 一个强大的python数据分析库



home // about // get pandas // documentation // community // talks // donate

Python Data Analysis Library

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

pandas is a NUMFocus sponsored project. This will help ensure the success of development of *pandas* as a world-class open-source project, and makes it possible to donate to the project.

VERSIONS

Release
0.21.0 - October 2017
download // docs // pdf

Development
0.21.0 - 2017
github // docs

安装 pandas

```
pip install pandas

# excel相关
pip install xlrd
pip install openpyxl
```

案例分析

- 泰坦尼克号沉船事件
  - 1952-04-15撞冰山沉没
  - 2224名乘客，死亡1502人

https://www.kaggle.com/c/titanic

乘客数据

- 每名乘客有12个属性

https://github.com/ruc-python/2017fall/tree/master/data/titanic

	A	B	C	D	E	F	G	H	I	J	K	L
1	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, Mr.	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, Mr	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen,	female	26	0	0	STON/02. 31	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, Mr.	male		0	0	330877	8.4583		Q
8	7	0	1	McCarthy, M	male	54	0	0	17463	51.8625	E46	S
9	8	0	3	Palsson, Ma	male	2	3	1	349909	21.075		S
10	9	1	3	Johnson, Mr	female	27	0	2	347742	11.1333		S
11	10	1	2	Nasser, Mrs	female	14	1	0	237736	30.0708		C
12	11	1	3	Sandstrom,	female	4	1	1	PP 9549	16.7	G6	S
13	12	1	1	Bonnell, Mi	female	58	0	0	113783	26.55	C103	S
14	13	0	3	Saundercock	male	20	0	0	A/5. 2151	8.05		S
15	14	0	3	Andersson,	male	39	1	5	347082	31.275		S
16	15	0	3	Vestrom, Mi	female	14	0	0	350406	7.8542		S

关于乘客数据的简单分析

- 男性 / 女性有多少人？
- 有多少人活下来了？
- 那些人活下来了？
  - 性别
  - 年龄
  - 社会阶层
  - 票价

pandas.read_excel()读取数据

读取一个excel表格
存入一个DataFrame

默认sheetname=0

```
In [1]: import pandas as pd

In [2]: df = pd.read_excel('data/titanic/train.xlsx', sheetname="train")

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

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

DataFrame用于数据统计的属性和方法

```
In [4]: df.shape
```
属性

```
Out[4]: (891, 12)
```

```
In [5]: df.Survived.value_counts()
```
方法

```
Out[5]: 0    549
        1    342
        Name: Survived, dtype: int64
```

```
In [6]: df.Fare.mean()
```
方法

```
Out[6]: 32.204207968574636
```

```
In [7]: mean_age, max_age, min_age = df.Age.mean(), df.Age.max(), df.Age.min()
        print mean_age, max_age, min_age
```

```
29.6991176471 80.0 0.42
```

给出数据的一个总体性的描述

- pandas.DataFrame.describe()

```
In [9]: df.describe()
```

Out[9]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

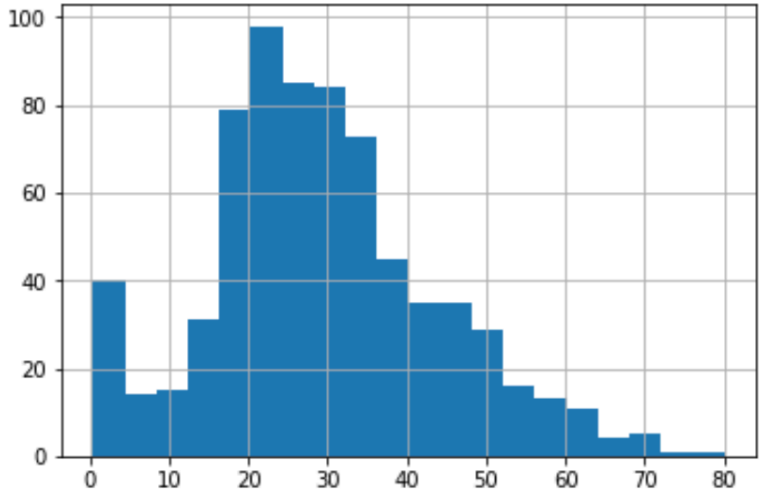
四分位数
Quartile

查看乘客年龄分布

- pandas.DataFrame.hist()

```
In [18]: import matplotlib.pyplot as plt
         %matplotlib inline
         df.Age.hist(bins=20)
```

```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x110eca590>
```



数据选择

- 获得表格中特定列的数据

In [24]:
```
df_1 = df[['Name', 'Sex', 'Age']]
df_1.head()
```

Out[24]:

	Name	Sex	Age
0	Braund, Mr. Owen Harris	male	22.0
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0
2	Heikkinen, Miss. Laina	female	26.0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0
4	Allen, Mr. William Henry	male	35.0

数据选择

- 按行取

取前6条记录

```
In [46]: df_1.iloc[:6]
```

Out[46]:

	Name	Sex	Age
0	Braund, Mr. Owen Harris	male	22.0
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0
2	Heikkinen, Miss. Laina	female	26.0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0
4	Allen, Mr. William Henry	male	35.0
5	Moran, Mr. James	male	NaN

取第6条记录

```
In [47]: df_1.iloc[5]
```

```
Out[47]: Name    Moran, Mr. James
         Sex                male
         Age                 NaN
         Name: 5, dtype: object
```

课堂练习1030-1

- 如何取最后6条记录？
- 如何取最后6条女性记录？

数据选择

- 布尔索引（boolean indexing）

```
In [56]: df_1.Sex == 'female'
Out[56]: 0      False
         1       True
         2       True
         3       True
         4      False
         5      False
         6      False
         7      False
         8       True
         9       True
         10      True
         11      True
         12     False
         13     False
         14      True
         15      True
```

```
In [58]: df_1[df_1.Sex == 'female']
Out[58]:
```

	Name	Sex	Age
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0
2	Heikkinen, Miss. Laina	female	26.0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0
8	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0
9	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0
10	Sandstrom, Miss. Marguerite Rut	female	4.0
11	Bonnell, Miss. Elizabeth	female	58.0
14	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0
15	Hewlett, Mrs. (Mary D Kingcome)	female	55.0
18	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31.0

数据选择

- 布尔索引（boolean indexing）

```
In [58]: df_1[df_1.Sex == 'female']
```

Out[58]:

	Name	Sex	Age
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0
2	Heikkinen, Miss. Laina	female	26.0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0
8	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0
9	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0
10	Sandstrom, Miss. Marguerite Rut	female	4.0
11	Bonnell, Miss. Elizabeth	female	58.0
14	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0
15	Hewlett, Mrs. (Mary D Kingcome)	female	55.0
18	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31.0

```
In [60]: df_1[(df_1.Sex == 'female') & (df_1.Age < 20)]
```

Out[60]:

	Name	Sex	Age
9	Nasser, Mrs. Nicholas (Adele Achem)	female	14.00
10	Sandstrom, Miss. Marguerite Rut	female	4.00
14	Vestrom, Miss. Hulda Amanda Adolfina	female	14.00
22	McGowan, Miss. Anna "Annie"	female	15.00
24	Palsson, Miss. Torborg Danira	female	8.00
38	Vander Planke, Miss. Augusta Maria	female	18.00
39	Nicola-Yarred, Miss. Jamila	female	14.00
43	Laroche, Miss. Simonne Marie Anne Andree	female	3.00
44	Devaney, Miss. Margaret Delia	female	19.00

课堂练习1030-2

- 如何取最后6条女性记录？
  - 用布尔索引实现

哪些人活下来了？

```
df[df.Survived == 0].describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	549.000000	549.0	549.000000	424.000000	549.000000	549.000000	549.000000
mean	447.016393	0.0	2.531876	30.626179	0.553734	0.329690	22.117887
std	260.640469	0.0	0.735805	14.172110	1.288399	0.823166	31.388207
min	1.000000	0.0	1.000000	1.000000	0.000000	0.000000	0.000000
25%	211.000000	0.0	2.000000	21.000000	0.000000	0.000000	7.854200
50%	455.000000	0.0	3.000000	28.000000	0.000000	0.000000	10.500000
75%	675.000000	0.0	3.000000	39.000000	1.000000	0.000000	26.000000
max	891.000000	0.0	3.000000	74.000000	8.000000	6.000000	263.000000

```
df[df.Survived == 1].describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	342.000000	342.0	342.000000	290.000000	342.000000	342.000000	342.000000
mean	444.368421	1.0	1.950292	28.343690	0.473684	0.464912	48.395408
std	252.358840	0.0	0.863321	14.950952	0.708688	0.771712	66.596998
min	2.000000	1.0	1.000000	0.420000	0.000000	0.000000	0.000000
25%	250.750000	1.0	1.000000	19.000000	0.000000	0.000000	12.475000
50%	439.500000	1.0	2.000000	28.000000	0.000000	0.000000	26.000000
75%	651.500000	1.0	3.000000	36.000000	1.000000	1.000000	57.000000
max	890.000000	1.0	3.000000	80.000000	4.000000	5.000000	512.329200

将DataFrame保存到excel文件

- DataFrame.to_excel(*excel_writer, sheet_name='Sheet1')*

```
writer = pd.ExcelWriter('new.xlsx')
df_1.to_excel(writer, 'train')
writer.save()
```

Pandas writes Excel files using the Xlwt module for xls files and the Openpyxl or XlsxWriter modules for xlsx files.

参考资料

- https://pandas.pydata.org/pandas-docs/stable/indexing.html
- http://xlsxwriter.readthedocs.io/working_with_pandas.html