



Data Exploration in Python USING

NumPy

NumPy stands for Numerical Python. This library contains basic linear algebra functions, Fourier transforms, advanced random number capabilities.

Pandas

Pandas for structured data operations and manipulations. It is extensively used for data munging and preparation.

Matplotlib

Python based plotting library offers matplotlib with a complete 2D support along with limited 3D graphic support.

CHEATSHEET

Contents

Data Exploration

1. How to load data file(s)?
2. How to convert a variable to different data type?
3. How to transpose a table?
4. How to sort Data?
5. How to create plots (Histogram, Scatter, Box Plot)?
6. How to generate frequency tables?
7. How to do sampling of Data set?
8. How to remove duplicate values of a variable?
9. How to group variables to calculate count, average, sum?
10. How to recognize and treat missing values and outliers?
11. How to merge / join data set effectively?

How to load data file(s)?

loading...
Here are some common functions used to read data

Function	Description
read_csv	Read delimited data from a file. Use Comma as default delimiter
read_table	Read delimited data from a file. Use tab ('\t') as default delimiter
read_excel	Read data from excel file
read_fwf	Read data in fixed width column format
read_clipboard	Read data from clipboard. Useful for converting tables from web pages

Loading data from CSV file(s):

CODE

```
import pandas as pd
#Import Library Pandas
df = pd.read_csv("E:/train.csv") #I am working in Windows environment
#Reading the dataset in a dataframe using Pandas
print df.head(3) #Prints first three observations
```

Output

	datetime	season	holiday	workingday	weather	temp	atemp	\
0	01-01-2011 00:00	1	0	0	1	9.84	14.395	
1	01-01-2011 01:00	1	0	0	1	9.02	13.635	
2	01-01-2011 02:00	1	0	0	1	9.02	13.635	

	humidity	windspeed	casual	registered	count
0	81	0	3	13	16
1	80	0	8	32	40
2	80	0	5	27	32

Loading data from excel file(s):

CODE

```
df=pd.read_excel("E:/EMP.xlsx", "Data") # Load Data sheet of excel file EMP
```

Loading data from txt file(s):

CODE

```
# Load Data from text file having tab '\t' delimiter print df
df=pd.read_csv("E:/Test.txt", sep='\t')
```

How to convert a variable to different data type?

- Convert numeric variables to string variables and vice versa

```
string_outcome = str(numeric_input) #Converts numeric_input to string_outcome
integer_outcome = int(string_input) #Converts string_input to integer_outcome
float_outcome = float(string_input) #Converts string_input to float_outcome
```

- Convert character date to Date

```
from datetime import datetime
char_date = 'Apr 1 2015 120 PM' #creating example character date
date_obj = datetime.strptime(char_date, '%b %d %Y %I:%M %p')
print date_obj
```

How to transpose a Data set?

- Data set used

Table A		
ID	Product	Sales
1	AAA	50
1	BBB	45
2	AAA	52
2	BBB	46

Table B		
ID	AAA	BBB
1	50	45
2	52	46

Code

```
#Transposing dataframe by a variable
```

```
df=pd.read_excel("E:/transpose.xlsx", "Sheet1") # Load Data sheet of excel file EMP
print df
result= df.pivot(index='ID', columns='Product', values='Sales')
result
```

Output

```
Out[35]:
```

	ID	Product	Sales
0	1	AAA	50
1	1	BBB	45
2	2	AAA	52
3	2	BBB	46

	Product	AAA	BBB
ID			
1		50	45
2		52	46

How to sort DataFrame?

CODE

```
#Sorting DataFrame
df=pd.read_excel("E:/transpose.xlsx", "Sheet1")
#Add by variable name(s) to sort
print df.sort(['Product','Sales'], ascending=[True, False])
```

or more subplots using
t, because you can't
x figure
subplot(1,1,1)

[Age],df['Sales'])

Scatter plot titled "Sales and Age distribution". The X-axis represents Age (1 to 4) and the Y-axis represents Sales (120 to 145). The data points are as follows:

Age	Sales
1	140
2	139
3	133
4	135
1	123
2	121

Original Table

Sorted Table

How to create plots (Histogram, Scatter, Box Plot)?

EmpID	Gender	Age	Sales
E001	M	34	123
E002	F	40	114
E003	F	37	135
E004	M	30	139
E005	F	44	117
E006	M	36	121
E007	M	32	133
E008	F	26	140
E009	M	32	133
E010	M	36	133

Histogram

Code

```
#Plot Histogram
```

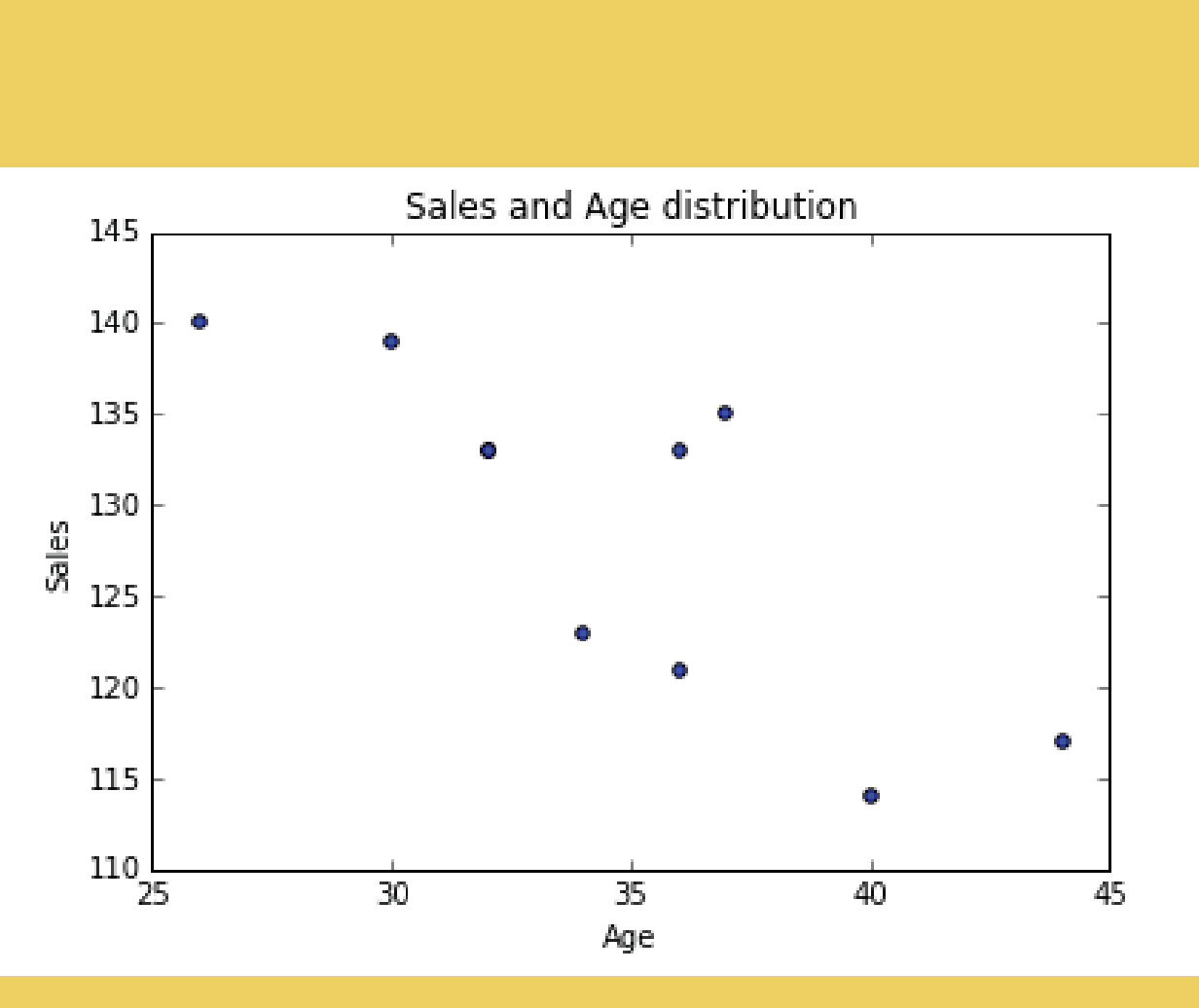
```
import matplotlib.pyplot as plt
import pandas as pd
```

```
df=pd.read_excel("E:/First.xlsx", "Sheet1")
#Plots in matplotlib reside within a figure object, use plt.figure to create new figure
fig=plt.figure()
```

```
#Create one or more subplots using add_subplot, because you can't create blank figure
ax = fig.add_subplot(1,1)
```

```
#Variable
ax.hist(df[Age], bins = 5)
```

```
#Labels and Tit
plt.title('Age distribution')
plt.xlabel('Age')
plt.ylabel('#Employee')
plt.show()
```



OutPut

Scatter plot

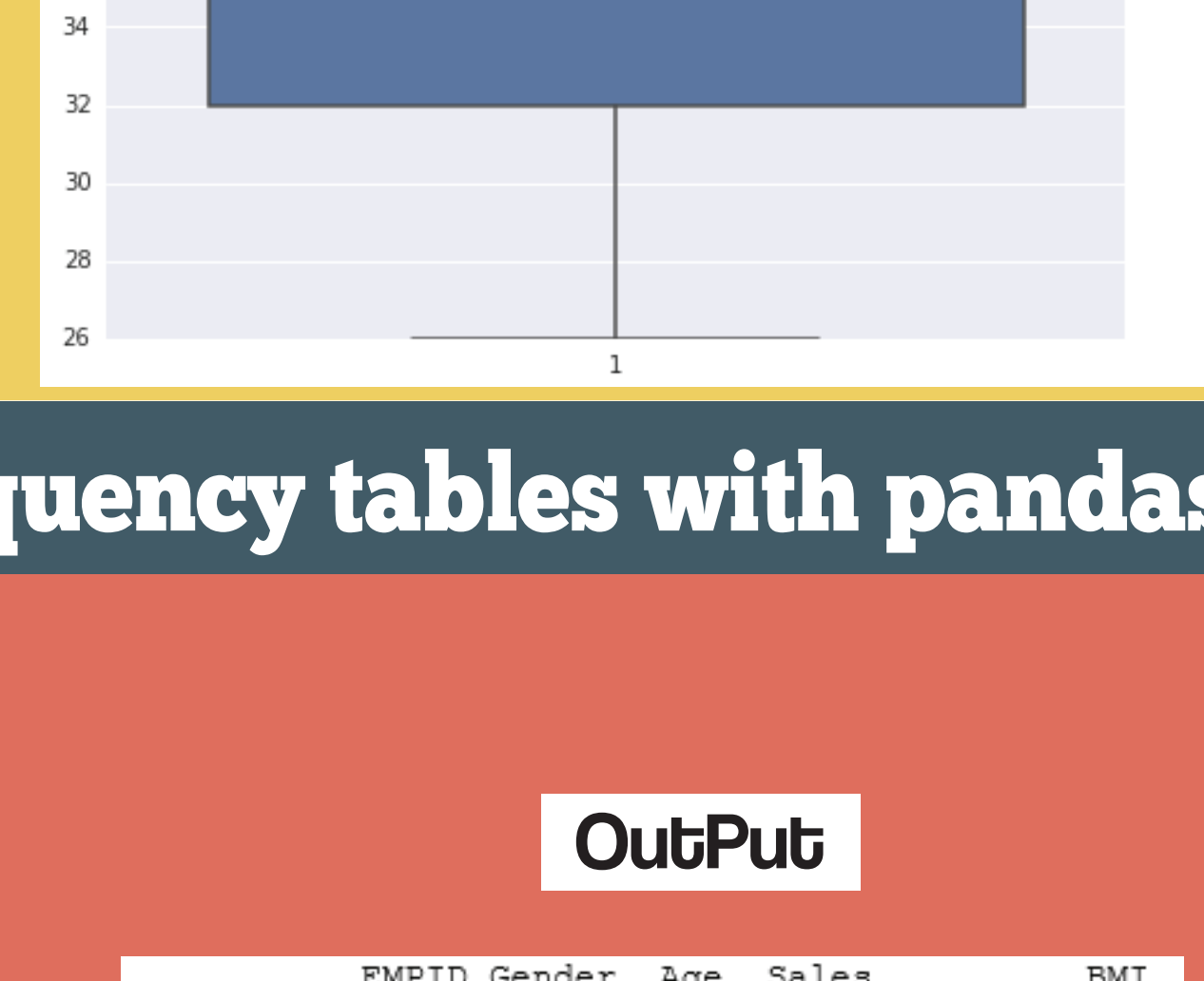
Code

```
#Plots in matplotlib reside within a figure object, use plt.figure to create new figure
fig=plt.figure()
```

```
#Create one or more subplots using add_subplot, because you can't create blank figure
ax = fig.add_subplot(1,1)
```

```
#Variable
ax.scatter(df[Age], df[Sales])
```

```
#Labels and Tit
plt.title('Sales and Age distribution')
plt.xlabel('Age')
plt.ylabel('Sales')
plt.show()
```

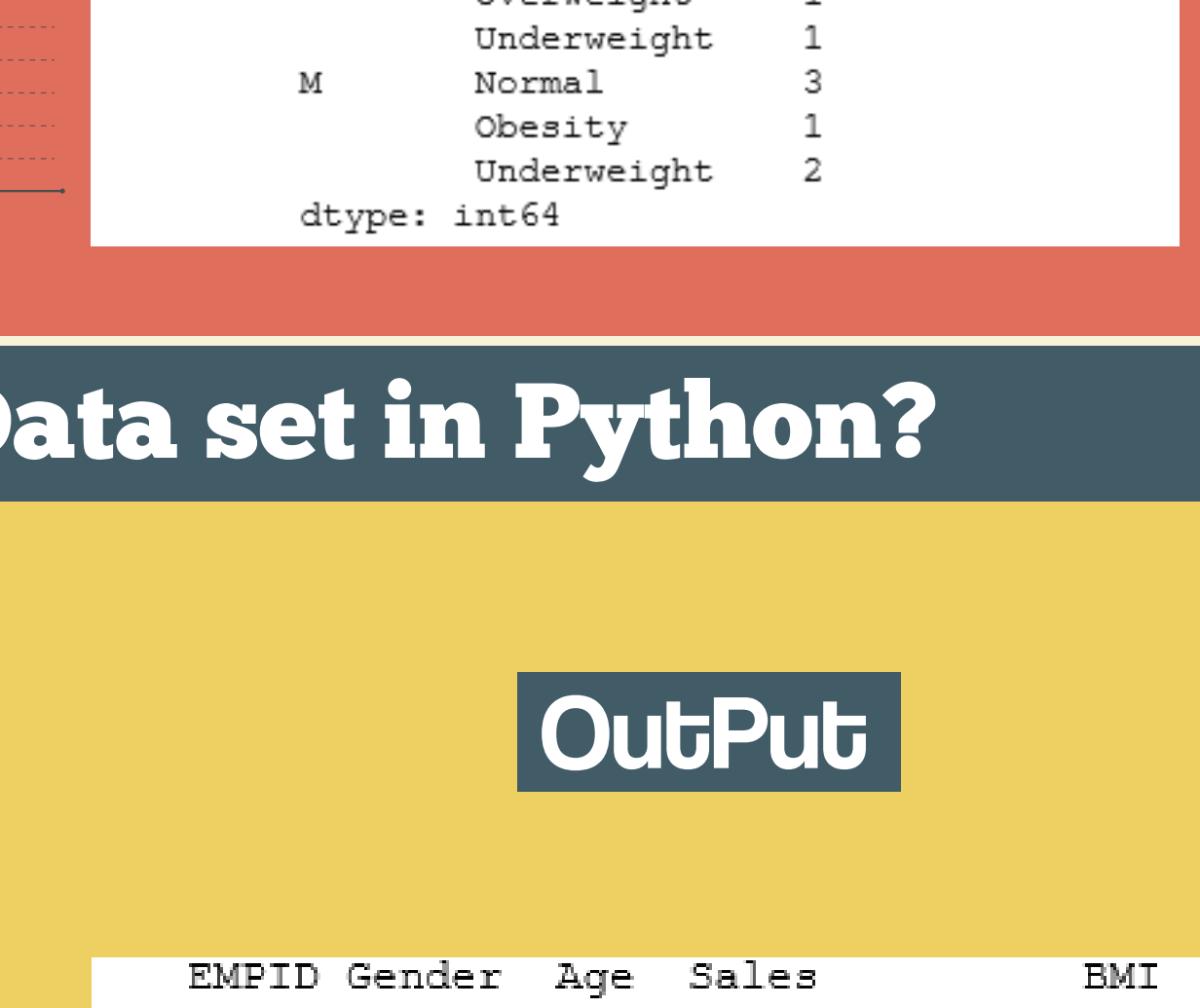


OutPut

Box-plot:

Code

```
import seaborn as sns
sns.boxplot(df[Age])
sns.despine()
```



OutPut

How to generate frequency tables with pandas?

Code

```
import pandas as pd
df=pd.read_excel("E:/First.xlsx", "Sheet1")
print df
test= df.groupby(['Gender','BMI'])
test.size()
```

OutPut

	EMPID	Gender	Age	Sales	BMI
0	E001	M	34	123	Normal
1	E002	F	40	114	Overweight
2	E003	F	37	135	Obesity
3	E004	M	30	139	Underweight
4	E005	F	44	117	Underweight
5	E006	M	36	121	Normal
6	E007	M	32	133	Obesity
7	E008	F	26	140	Normal
8	E009	M	32	133	Normal
9	E010	M	36	133	Underweight

```
Out[84]:
```

	Gender	BMI
F	Normal	1
	Obesity	1
	Overweight	1
M	Underweight	1
	Normal	3
	Obesity	1
	Underweight	2

dtype: int64

How to do sample Data set in Python?

Code

```
#Create Sample dataframe
import numpy as np
import pandas as pd
from random import sample
```

```
# create random index
rindex = np.array(sample(xrange(len(df)), 5))
```

```
# get 5 random rows from df
dfr = df.ix[rindex]
print dfr
```

	EMPID	Gender	Age	Sales	BMI
4	E005	F	44	117	Underweight
2	E003	F	37	135	Obesity
7	E008	F	26	140	Normal
8	E009	M	32	133	Normal
5	E006	M	36	121	Normal

OutPut

How to remove duplicate values of a variable?

Code

```
#Remove Duplicate Values based on values of variables "Gender" and "BMI"
```

```
rem_dup=df.drop_duplicates(['Gender','BMI'])
print rem_dup
```

Output

	EMPID	Gender	Age	Sales	BMI
0	E001	M	34	123	Normal
1	E002	F	40	114	Overweight
2	E003	F	37	135	Obesity
3	E004	M	30	139	Underweight
4	E005	F	44	117	Underweight
6	E007	M	32	133	Obesity
7	E008	F	26	140	Normal

How to group variables in Python to calculate count, average, sum?

Code

```
test= df.groupby(['Gender'])
test.describe()
```

Output

		Age	Sales
F	count	4.000000	4.000000
	mean	36.750000	126.500000
	std	7.719024	12.922848
	min	26.000000	114.000000
	25%	34.250000	116.250000
	50%	38.500000	126.000000
	75%	41.000000	136.250000
	max	44.000000	140.000000
M	count	6.000000	6.000000
	mean	33.333333	130.333333
	std	2.422120	6.889606
	min	30.000000	121.000000
	25%	32.000000	125.000000
	50%	33.000000	133.000000
	75%	35.000000	133.000000
	max	36.000000	139.000000

How to recognize and Treat missing values and outliers?

Code

```
# Identify missing values of dataframe
df.isnull()
```

Output

```
In [116]: # Identify missing values of dataframe
df.isnull()
```

```
Out [116]:
```

	EMPID	Gender	Age	Sales	BMI
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False
9	False	False	False	False	False

```
#Example to impute missing values in Age by the mean
import numpy as np
#Using numpy mean function to calculate the mean value
meanAge = np.mean(df.Age)
#replacing missing values in the DataFrame
df.Age = df.Age.fillna(meanAge)
```

How to merge / join data sets?

Code

```
df_new = pd.merge(df1, df2, how = 'inner', left_index = True, right_index = True)
# merges df1 and df2 on index
# By changing how = 'outer', you can do outer join.
# Similarly how = 'left' will do a left join
# You can also specify the columns to join instead of indexes, which are used by default.
```