Scipy_feb14

August 19, 2022

[ ]:

```
# Scipy------Scientific python

# Scientific calculation

# Statistical Analysis

# Advanced mathematical Analysis(integration,linear algebra...)
```

[1]:

```
import scipy.constants

from scipy import constants
```

[2]:

```
print(constants.pi)
```

```
3.141592653589793
```

[4]:

```
print(dir(constants))
```

```
['Avogadro', 'Boltzmann', 'Btu', 'Btu_IT', 'Btu_th', 'ConstantWarning', 'G',
'Julian_year', 'N_A', 'Planck', 'R', 'Rydberg', 'Stefan_Boltzmann', 'Wien',
'__all__', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__',
'__name__', '__package__', '__path__', '__spec__', '_obsolete_constants',
'absolute_import', 'acre', 'alpha', 'angstrom', 'arcmin', 'arcminute', 'arcsec',
'arcsecond', 'astronomical_unit', 'atm', 'atmosphere', 'atomic_mass', 'atto',
'au', 'bar', 'barrel', 'bbl', 'blob', 'c', 'calorie', 'calorie_IT',
'calorie_th', 'carat', 'centi', 'codata', 'constants', 'convert_temperature',
'day', 'deci', 'degree', 'degree_Fahrenheit', 'deka', 'division', 'dyn', 'dyne',
'e', 'eV', 'electron_mass', 'electron_volt', 'elementary_charge', 'epsilon_0',
'erg', 'exa', 'exbi', 'femto', 'fermi', 'find', 'fine_structure', 'fluid_ounce',
'fluid_ounce_US', 'fluid_ounce_imp', 'foot', 'g', 'gallon', 'gallon_US',
'gallon_imp', 'gas_constant', 'gibi', 'giga', 'golden', 'golden_ratio', 'grain',
'gram', 'gravitational_constant', 'h', 'hbar', 'hectare', 'hecto', 'horsepower',
'hour', 'hp', 'inch', 'k', 'kgf', 'kibi', 'kilo', 'kilogram_force', 'kmh',
'knot', 'lambda2nu', 'lb', 'lbf', 'light_year', 'liter', 'litre', 'long_ton',
'm_e', 'm_n', 'm_p', 'm_u', 'mach', 'mebi', 'mega', 'metric_ton', 'micro',
'micron', 'mil', 'mile', 'milli', 'minute', 'mmHg', 'mph', 'mu_0', 'nano',
'nautical_mile', 'neutron_mass', 'nu2lambda', 'ounce', 'oz', 'parsec', 'pebi',
'peta', 'physical_constants', 'pi', 'pico', 'point', 'pound', 'pound_force',
'precision', 'print_function', 'proton_mass', 'psi', 'pt', 'short_ton', 'sigma',
'slinch', 'slug', 'speed_of_light', 'speed_of_sound', 'stone', 'survey_foot',
'survey_mile', 'tebi', 'tera', 'test', 'ton_TNT', 'torr', 'troy_ounce',
'troy_pound', 'u', 'unit', 'value', 'week', 'yard', 'year', 'yobi', 'yotta',
'zebi', 'zepto', 'zero_Celsius', 'zetta']
```

[5]:

```
print(constants.kilo)
```

```
1000.0
```

[ ]:

```
#mathematical operation
# integration

# 1 integral ----------Quad

# double integral---------dblquad

# triple integral---------->tplquad
```

[9]:

```
#integration

# 3x+2

from scipy.integrate import quad

def inte(x):
    return 3*x+7


quad(inte,0,1)
```

[9]:

```
(8.5, 9.43689570931383e-14)
```

[10]:

```
#double integral
from scipy.integrate import dblquad

def inte1(x,y):
    return 3*x+4*y

dblquad(inte1,0,1,4,7)# (0,1)-->xlimit  (4,7) ylimit
```

[10]:

```
(55.50000000000001, 6.824978690884149e-13)
```

[12]:

```
# Statistics package is stats

# oulier analyis

import seaborn as sns
```

```python
data=sns.load_dataset("tips")
data
```

[12]:

```
     total_bill   tip     sex smoker   day    time  size
0         16.99  1.01  Female     No   Sun  Dinner     2
1         10.34  1.66    Male     No   Sun  Dinner     3
2         21.01  3.50    Male     No   Sun  Dinner     3
3         23.68  3.31    Male     No   Sun  Dinner     2
4         24.59  3.61  Female     No   Sun  Dinner     4
..          …     …       …     …     …       …     …
239       29.03  5.92    Male     No   Sat  Dinner     3
240       27.18  2.00  Female    Yes   Sat  Dinner     2
241       22.67  2.00    Male    Yes   Sat  Dinner     2
242       17.82  1.75    Male     No   Sat  Dinner     2
243       18.78  3.00  Female     No  Thur  Dinner     2

[244 rows x 7 columns]
```

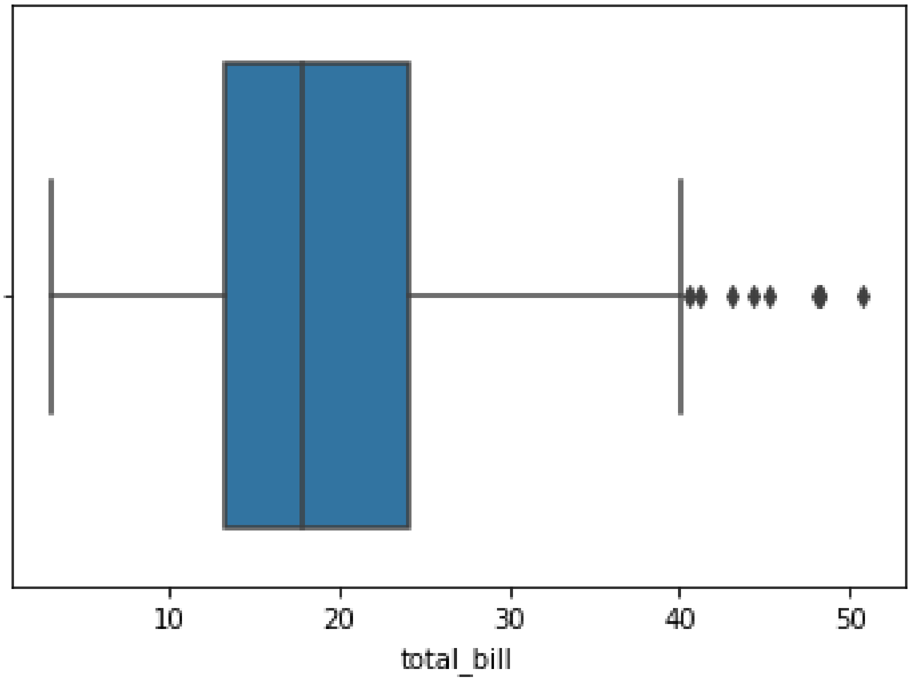
[14]:

```python
sns.boxplot(data["total_bill"])
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

[14]:

```
<AxesSubplot:xlabel='total_bill'>
```



[ ]:

```
# outlier

# Q1--->25th percentile value

# Q2--------->median( 50th percentile)

# Q3------->75th percentile value

# IQR= Q3-Q1

# LB=Q1-1.5*IQR

# UB= Q3+1.5*IQR
```

[16]:

```
from scipy import stats # statistical package

q1=stats.scoreatpercentile(data["total_bill"],25)

q3=stats.scoreatpercentile(data["total_bill"],75)

iqr=q3-q1

LB=q1-1.5*iqr
```

```
UB=q3+1.5*iqr

print("Q1---",q1)
print("Q3---",q3)
print("IQR---",iqr)

print("lower bound-------->",LB)
print("upper bound-------->",UB)
```

```
Q1--- 13.3475
Q3--- 24.127499999999998
IQR--- 10.779999999999998
lower bound--------> -2.8224999999999945
upper bound--------> 40.29749999999999
```

[18]:
```
outliervalues=    data["total_bill"] [((data["total_bill"]<LB) |␣
 ↪(data["total_bill"]>UB))]
outliervalues
```

[18]:
```
59     48.27
102    44.30
142    41.19
156    48.17
170    50.81
182    45.35
184    40.55
197    43.11
212    48.33
Name: total_bill, dtype: float64
```

[30]:
```
# Hypothesis testing

# correlation

import numpy as np

salary=np.array([-100,200,-300,-400,500])
exp=np.array([3,5,7,9,10])
```

[31]:
```
#pearsan correaltion

#help(stats.pearsonr)
```

[32]:
```
coeff,pval=stats.pearsonr(salary,exp)
print("coefffficent",coeff)
print("Pvalue",pval)
```

```
coeffficent 0.1603916339241686
Pvalue 0.796662035722664
```

[33]:
```python
if pval<0.05:
    print("Alter Hypo----->relation exist")
else:
    print("Null Hypo---->No relation")
```

```
Null Hypo---->No relation
```

[34]:
```python
#chisquare
#2 categorical value
data
```

[34]:
```
     total_bill   tip     sex smoker   day    time  size
0         16.99  1.01  Female     No   Sun  Dinner     2
1         10.34  1.66    Male     No   Sun  Dinner     3
2         21.01  3.50    Male     No   Sun  Dinner     3
3         23.68  3.31    Male     No   Sun  Dinner     2
4         24.59  3.61  Female     No   Sun  Dinner     4
..          …     …       …      …     …       …   …
239       29.03  5.92    Male     No   Sat  Dinner     3
240       27.18  2.00  Female    Yes   Sat  Dinner     2
241       22.67  2.00    Male    Yes   Sat  Dinner     2
242       17.82  1.75    Male     No   Sat  Dinner     2
243       18.78  3.00  Female     No  Thur  Dinner     2

[244 rows x 7 columns]
```

[50]:
```python
import pandas as pd
data1=pd.crosstab(data["sex"],data["smoker"])
data1
```

[50]:
```
smoker  Yes  No
sex
Male     60  97
Female   33  54
```

[56]:
```python
#help(stats.chi2_contingency)
```

[52]:
```python
data1.values
```

[52]:
```
array([[60, 97],
       [33, 54]])
```

[53]:
```python
coff,pval,dof,expec=stats.chi2_contingency(data1.values)
print("chisquare",coff)
print("Pvalue",pval)
print("DOF",dof)
print("Expected",expec)
```

```
chisquare 0.008763290531773594
Pvalue 0.925417020494423
DOF 1
Expected [[59.84016393 97.15983607]
 [33.15983607 53.84016393]]
```

[45]:

```
if pval<0.05:
    print("Alter Hypo----->relation exist")
else:
    print("Null Hypo---->No relation")
```

```
Null Hypo---->No relation
```

[ ]:

```

```