Statistical analysis: 

Numpy: Numerical computation of a large set of numbers

Scipy: Algorithms to solve problems.

Pandas: Manipulate data

→ Analysis of data →

Scipy: Statistics module
Statsmodels
Pymc3
Stan
Scikit-learn
TensorFlow

Load data

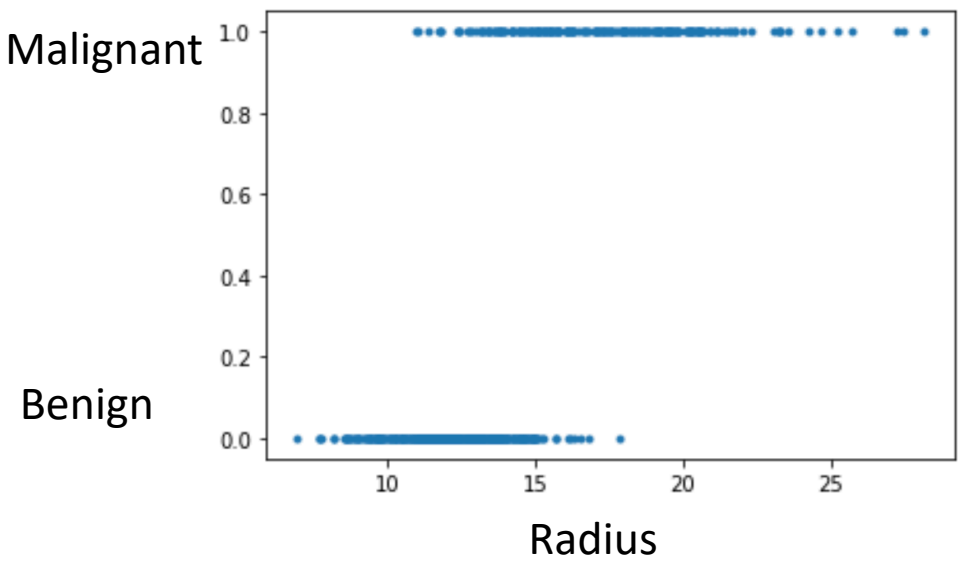
statsmodels uses Pandas dataframe

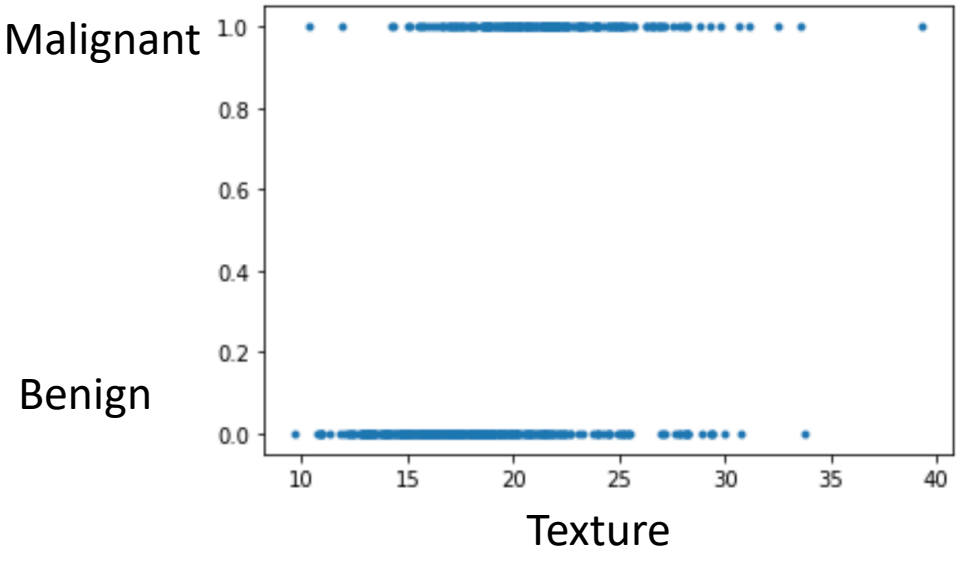
```
import numpy as np
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf

bcd = pd.read_csv('data/wdbc.data', header = None, usecols=np.arange(12), index_col=0)
bcd.columns = [ 'Diagnosis', 'Radius','Texture', 'Perimeter', 'Area', 'Smoothness',
                'Compactness','Concavity','Concave points','Symmetry','Fractal dimension']
bcd.index.name = 'ID'
bcd.head()
```

ID	Diagnosis	Radius	Texture	Perimeter	Area	Smoothness	Compactn
842302	M	17.99	10.38	122.80	1001.0	0.11840	0.277
842517	M	20.57	17.77	132.90	1326.0	0.08474	0.078
84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.159
84348301	M	11.42	20.38	77.58	386.1	0.14250	0.283
84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.132

Plotting the data

```
bcd["Bool_Diagnosis"] = (bcd["Diagnosis"]=='M').astype('int')

import matplotlib.pyplot as plt
plt.plot(bcd.Radius, bcd.Bool_Diagnosis, '.')
plt.plot(bcd.Texture, bcd.Bool_Diagnosis, '.')
```





Regression analysis

$$Diagnosis = a \bullet Radius + b \bullet Texture + const + noise$$

```
import statsmodels.formula.api as smf
mdl = smf.ols('Bool_Diagnosis ~ np.log(Radius) + Texture + 1', data=bcd)
results = mdl.fit()
```

formula

From stats v3.6.
by R-core R-core

Model Formulae

The generic function `formula` and its specific methods provide a way of extracting formulae which have been included in other objects.

`as.formula` is almost identical, additionally preserving attributes when `object` already inherits from `"formula"`.

Keywords models

Usage

```
formula(x, …)
DF2formula(x, env = parent.frame())
as.formula(object, env = parent.frame())

# S3 method for formula
print(x, showEnv = !identical(e, .GlobalEnv), …)
```

Arguments

x, object	R object, for `DF2formula()` a `data.frame`.
...	further arguments passed to or from other methods.
env	the environment to associate with the result, if not already a formula.
showEnv	logical indicating if the environment should be printed as well.

https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/formula
https://patsy.readthedocs.io/en/latest/formulas.html#the-formula-language

Result of regression analysis

```
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         Bool_Diagnosis   R-squared:                       0.566
Model:                            OLS   Adj. R-squared:                  0.564
Method:                 Least Squares   F-statistic:                     368.9
Date:                Sat, 06 Mar 2021   Prob (F-statistic):          2.82e-103
Time:                        07:44:36   Log-Likelihood:                -156.49
No. Observations:                 569   AIC:                             319.0
Df Residuals:                     566   BIC:                             332.0
Df Model:                           2
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept         -3.5927      0.149    -24.170      0.000      -3.885      -3.301
np.log(Radius)     1.3439      0.059     22.649      0.000       1.227       1.460
Texture            0.0231      0.003      7.026      0.000       0.017       0.030
==============================================================================
Omnibus:                       22.521   Durbin-Watson:                   1.559
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               22.690
Skew:                           0.454   Prob(JB):                     1.18e-05
Kurtosis:                       2.637   Cond. No.                         236.
==============================================================================
```

Regression analysis using Numpy array

```
y = bcd.Bool_Diagnosis.to_numpy()
x = np.asarray(bcd.loc[:,["Radius","Texture"]])
x[:,0] = np.log(x[:,0])
```

```
import statsmodels.api as sm
mdl = sm.OLS(y, sm.add_constant(x))
results = mdl.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.566
Model:                            OLS   Adj. R-squared:                  0.564
Method:                 Least Squares   F-statistic:                     368.9
Date:                Sat, 06 Mar 2021   Prob (F-statistic):          2.82e-103
Time:                        06:40:29   Log-Likelihood:                -156.49
No. Observations:                 569   AIC:                             319.0
Df Residuals:                     566   BIC:                             332.0
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -3.5927      0.149    -24.170      0.000      -3.885      -3.301
x1             1.3439      0.059     22.649      0.000       1.227       1.460
x2             0.0231      0.003      7.026      0.000       0.017       0.030
==============================================================================
Omnibus:                       22.521   Durbin-Watson:                   1.559
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               22.690
Skew:                           0.454   Prob(JB):                     1.18e-05
Kurtosis:                       2.637   Cond. No.                         236.
==============================================================================
```

Generalized Linear Models (GLMs)

https://www.statsmodels.org/stable/glm.html

```
y = bcd.Bool_Diagnosis.to_numpy()
x = bcd.loc[:,["Radius","Texture", 'Smoothness', 'Compactness','Concavity','Symmetry']].to_numpy()

gaussian_model = sm.GLM(y, sm.add_constant(x), family=sm.families.Gaussian())
gaussian_results = gaussian_model.fit()
print(gaussian_results.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  569
Model:                            GLM   Df Residuals:                      562
Model Family:                Gaussian   Df Model:                            6
Link Function:               identity   Scale:                         0.081946
Method:                          IRLS   Log-Likelihood:                -92.122
Date:                Sat, 06 Mar 2021   Deviance:                       46.054
Time:                        06:40:30   Pearson chi2:                     46.1
No. Iterations:                     3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -1.8349      0.158    -11.596      0.000      -2.145      -1.525
x1             0.0693      0.005     13.844      0.000       0.060       0.079
x2             0.0218      0.003      7.191      0.000       0.016       0.028
x3             5.0654      1.227      4.127      0.000       2.660       7.471
x4            -0.3085      0.580     -0.532      0.595      -1.445       0.828
x5             1.2855      0.392      3.275      0.001       0.516       2.055
x6             1.3100      0.580      2.259      0.024       0.174       2.446
==============================================================================
```

T-test

https://www.statsmodels.org/stable/generated/statsmodels.stats.weightstats.ttest_ind.html

```
import statsmodels.stats.weightstats as ws

a = bcd.loc[ bcd.Diagnosis == 'M' , 'Radius' ]
b = bcd.loc[ bcd.Diagnosis == 'B' , 'Radius' ]

tstat, pvalue, df = ws.ttest_ind(a,b)
print('test statistics:', tstat)
print('degree of freedom:', df)
print('p-value:', pvalue)
```

```
test statistics: 25.435821610057015
degree of freedom: 567.0
p-value: 8.46594057226676e-96
```

Summary

Analysis of data 

Scipy: Statistics module
Statsmodels
Pymc3
Stan
Scikit-learn
TensorFlow