# Machine Learning

Samatrix Consulting Pvt Ltd

# Project – Advertisement Budget

# Project - Introduction

**Project - Marketing**

- Infer relationship between sales and the three media budgets: TV, Radio and Newspaper.

**Project Steps Followed**

- Define Project Goals/Objective
- Data Retrieval
- Data Cleansing
- Exploratory Data Analysis
- Data Modeling
- Result Analysis

# Project - Introduction

- Suppose you have been assigned as Data Scientist to advice on how to improve the sales of a particular product of a company.
- The company provided you with sales data from 200 different markets.
- The data also contains the advertising budgets for the product in each of those markets for three different media: TV, radio, and newspaper.
- The client cannot directly increase the sales of the product.
- But they can adjust the advertisement budget for each of the three media.
- As a data scientist, if you can establish the relationship between advertisement expenditure and sales, you can provide your feedback on how to adjust the budgets so that sales can increase.
- So, the objective is to develop a model that you can use to predict the sales on the basis of the three media budgets.

# Project - Introduction

- Define Research Goals
  - Infer relationship between sales and three media budgets: TV, Radio and Newspaper
- Data Set
  - The Data set can be downloaded
  - The dataset contains sales data from 200 markets
  - By the end of the project, the learners will be able to learn the approaches required for Multiple Linear Regression

# Import Libraries and Load the Data

## Import the Libraries

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

In [2]: import numpy as np

In [3]: import matplotlib.pyplot as plt

In [4]: import seaborn as sns
```

## Load the data

```
In [5]: advertising =
pd.read_csv(‘Data/Advertising.csv',usecols=[1,2,3,4])
```

Samatrix.io

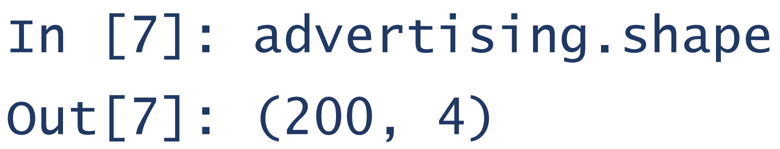
# Understanding Data

- The most important step of model development is understanding the dataset. Generally, we follow the following steps to understand the data:
  - View the raw data
  - Dimensions of the dataset
  - Data Types of the attributes
  - Presence of Null Values in the dataset
  - Statistical Analysis
  - Data Errors (zero values)

# View the raw data

```
In [6]: advertising.head()
Out[6]:
      TV  Radio  Newspaper  Sales
0  230.1   37.8       69.2   22.1
1   44.5   39.3       45.1   10.4
2   17.2   45.9       69.3    9.3
3  151.5   41.3       58.5   18.5
4  180.8   10.8       58.4   12.9
```

# Dimension of the Data



```
In [7]: advertising.shape
Out[7]: (200, 4)
```

We get the dimension of the dataset. The dataset has 200 rows and 4 columns.

# Data Type

```
In [8]: advertising.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

# Data Type

- Our observations are as follows
  - NaN values do not present in the data set. Because of the Non-Null Count and number of rows in the dataset match.
  - There are 3 Input Variables and 1 Output Variable (Sales)
  - The data type of all the input variables is float64. The data type of out variable (Sales) is float64
  - Shows that all the input as well as output variables are continuous (quantitative) data types.
  - None of the columns contain the Null Values

# Null Values

```
In [9]: advertising.isnull().sum()

Out[9]:
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

The dataset does not contain any null values

# Exploratory Data Analysis

# Data Analysis

```
In [11]:
advertising.describe()
Out[11]:
              TV    Radio  Newspaper    Sales
count   200.00   200.00     200.00   200.00
mean    147.04    23.26      30.55    14.02
std      85.85    14.85      21.78     5.22
min       0.70     0.00       0.30     1.60
25%      74.38     9.97      12.75    10.38
50%     149.75    22.90      25.75    12.90
75%     218.82    36.52      45.10    17.40
max     296.40    49.60     114.00    27.00
```

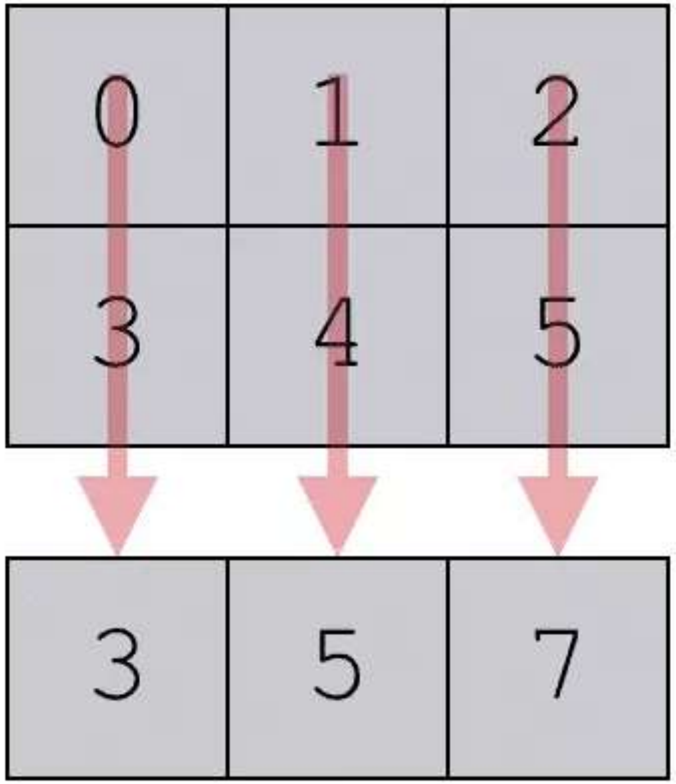
We can see that the min value of Radio is zero. We need to confirm how many zero values existing in the dataset. For all other columns, the data cleaning is not required. However the data scaling is required.

# Analysis of Zero Values in Predictors

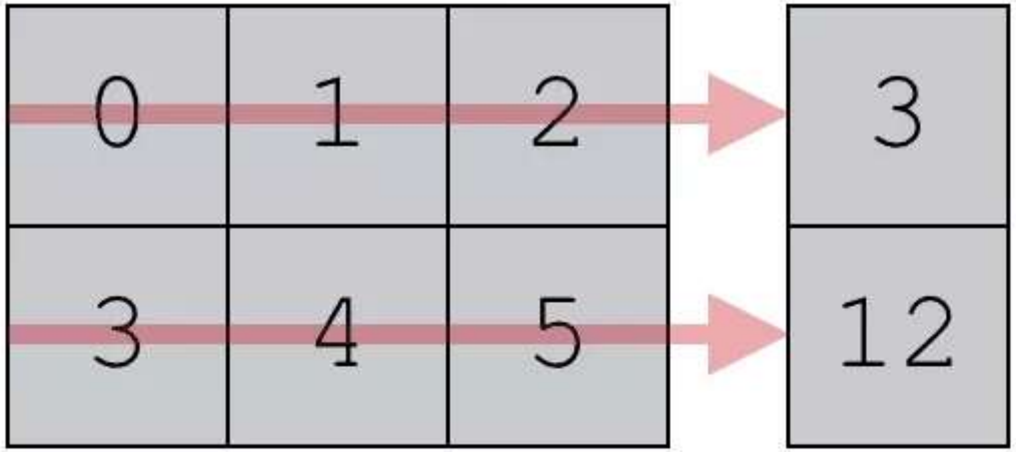
```
In [12]: (advertising == 0).sum(axis=0)
Out[12]:
TV           0
Radio        1
Newspaper    0
Sales        0
dtype: int64
```

Only one row of Radio variable contain the zero value, which is possible. Hence we conclude the data cleaning steps are not required.

# WHEN WE SET `axis = 0,np.sum()` COLLAPSES THE ROWS AND CALCULATES THE SUM



# WHEN WE SET `axis = 1,np.sum()` COLLAPSES THE COLUMNS AND CALCULATES THE SUM



| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 3 | 4 | 5 | 12 |

# Response Variable Analysis

```
In [13]:
advertising.Sales.value_counts()
```

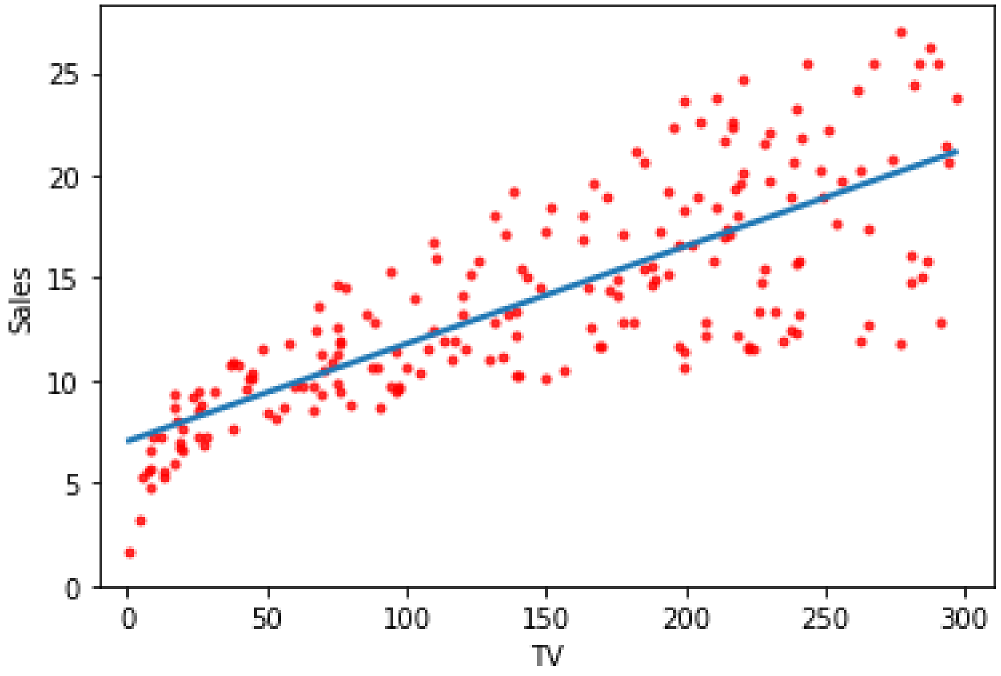
```
Out[13]:
9.7     5
12.9    4
11.7    4
15.9    4
25.4    3
       ..
15.7    1
14.2    1
11.2    1
19.4    1
18.5    1
Name: Sales, Length: 121, dtype: int64
```

- This confirms that the response variable is continuous. The unique values are 121 out of 200.

Samatrix.io

# Relation between Sales and TV

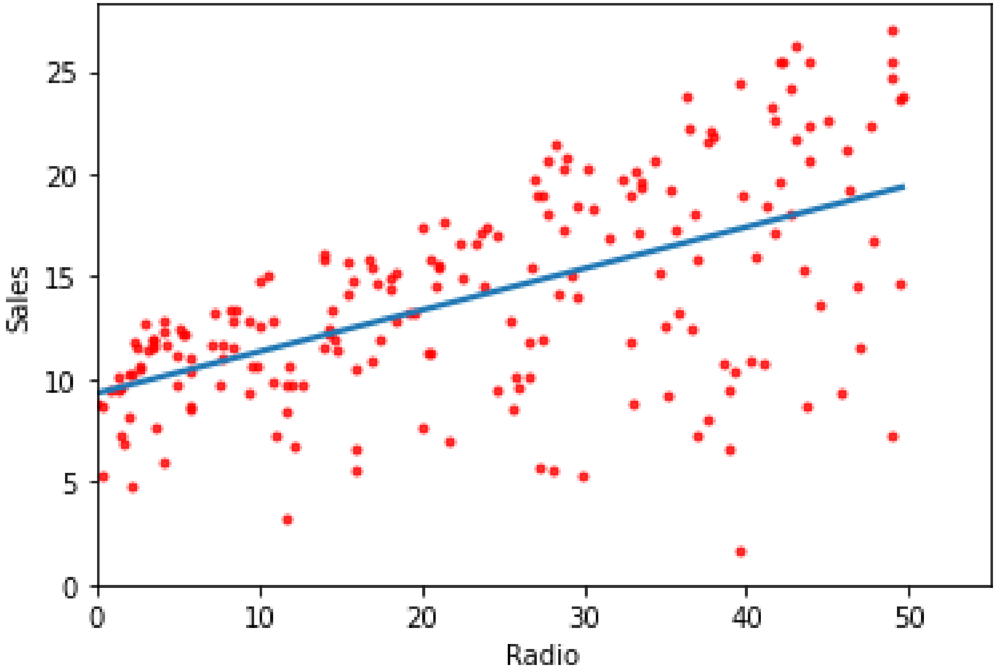
```
In [14]: sns.regplot(x=advertising.TV,
y=advertising.Sales, order=1, ci=None,
scatter_kws={'color':'r',
    plt.xlim(-10,310)
    plt.ylim(bottom=0)
    plt.show()
```



```
# order 1 for linear model
# ci - confidence interval
#scatter_kws Color - red size - 9
```

# Relation between Sales and Radio

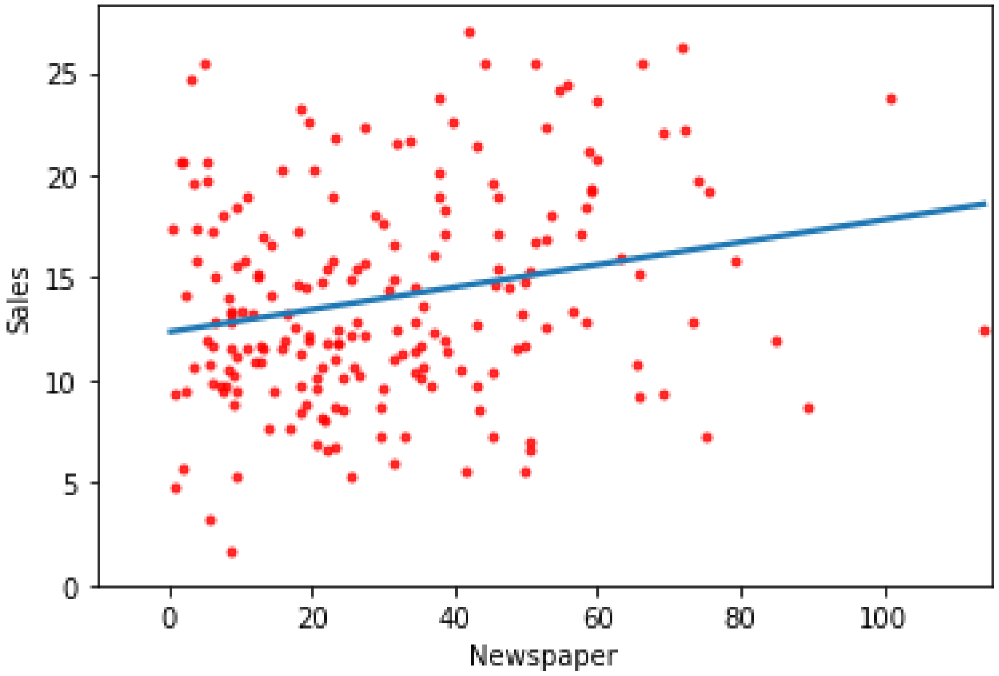
```
In [15]: sns.regplot(advertising.Radio, advertising.Sales, order=1, ci=None, sca
    ...: tter_kws={'color':'r', 's':9})
    ...: plt.xlim(0,55)
    ...: plt.ylim(bottom=0)
    ...: plt.show()
```



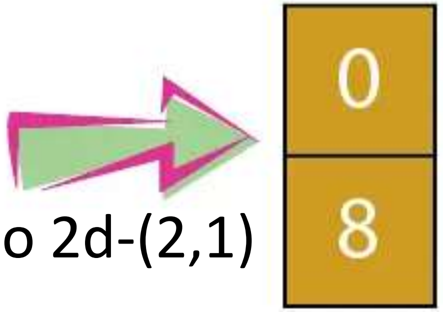
Samatrix.io

# Relation between Sales and Newspaper

```
In [16]: sns.regplot(advertising.Newspaper, advertising.Sales, order=1, ci=None,
    ...:  scatter_kws={'color':'r', 's':9})
    ...: plt.xlim(-10,115)
    ...: plt.ylim(bottom=0)
    ...: plt.show()
```



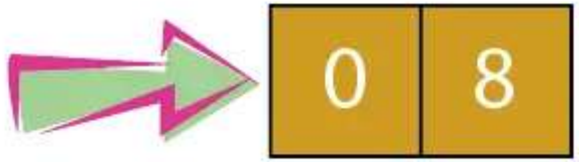
```
np.array([0,8]).reshape(-1,1)
```



| |
|---|
| 0 |
| 8 |

The dimensions of the arrays also change from 1d-(2,) to 2d-(2,1)

```
np.array ([0,8]).reshape(1,-1)
```

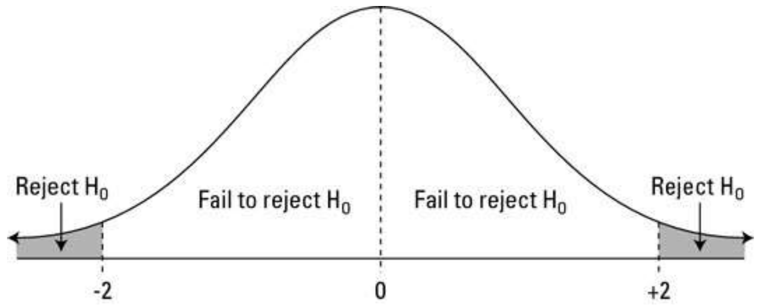


| | |
|---|---|
| 0 | 8 |

The dimensions of the arrays also change from 1d-(2,) to 2d-(1,2)

# Data Modeling

# Hypothesis Testing



- We can also use the standard errors to perform hypothesis tests on the coefficients. The null hypothesis is

$$H_0 : There\ is\ no\ relationship\ between\ X\ and\ Y$$

- The alternative hypothesis

$$H_a : There\ is\ some\ relationship\ between\ X\ and\ Y$$

- This corresponds to testing

$$H_0 :\ \beta_1 = 0$$

- Versus

$$H_a : \beta_1 \neq 0$$

Samatrix.io

# What is p-value?

- It is an alternative method for rejecting or Accepting the null hypothesis.
- It overcomes the problem that if researcher wants to apply the results of the study using different values of alpha(α).
- The steps followed are

a. Probability of the event of interest

b. Probability of events having same probability as above

c. Probability of rarer events.

**P-value is computed assuming the null hypothesis is true.**

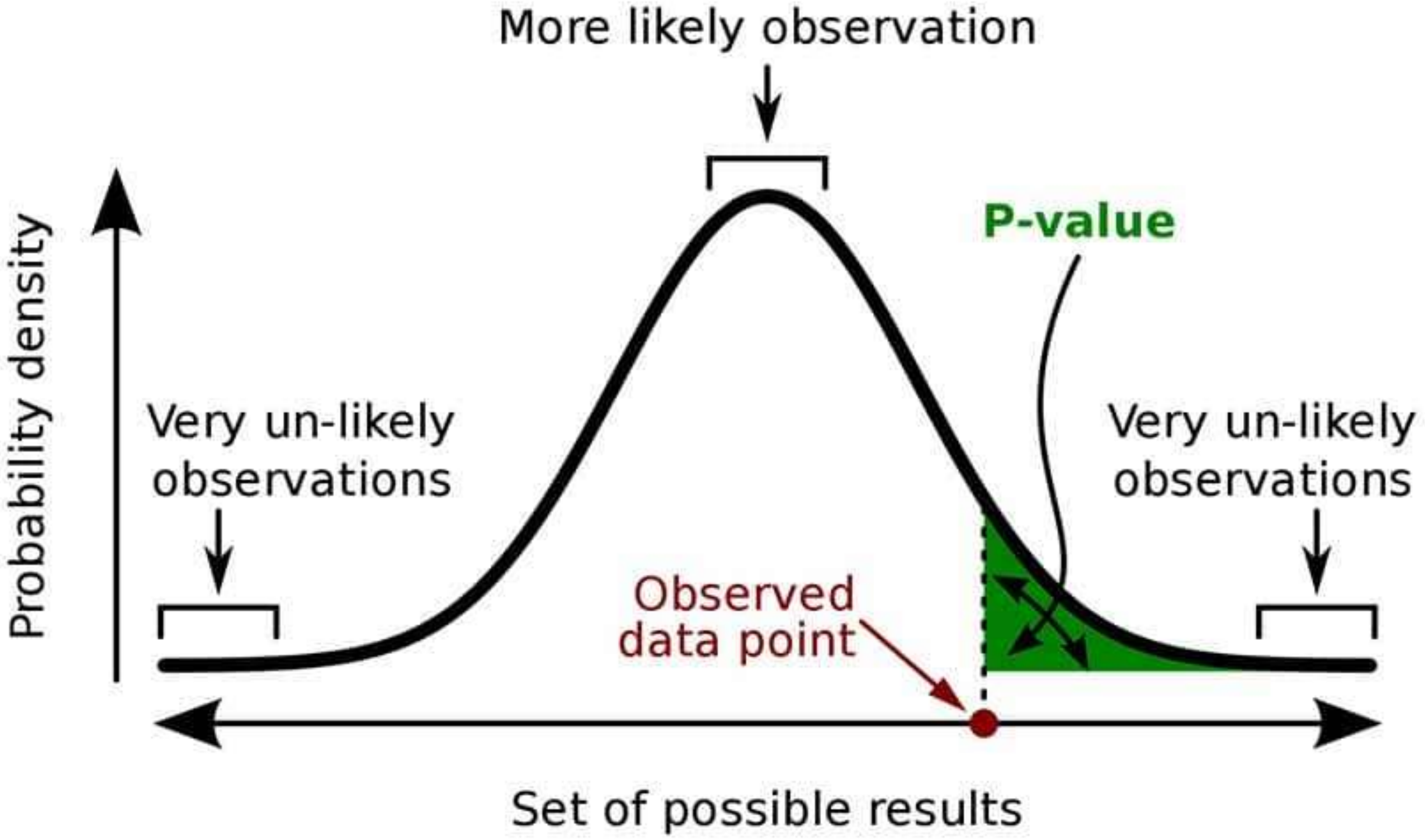
- P values are expressed as decimals although it may be easier to understand what they are if you convert them to a percentage.
- For example, a p value of 0.0254 is 2.54%. This means there is a 2.54% chance your results could be random (i.e. happened by chance). That’s pretty tiny.
- On the other hand, a large p-value of .9(90%) means your results have a 90% probability of being completely random and not due to anything in your experiment.
- **Therefore, the smaller the p-value, the more important (“significant“) your results**.

# P Value vs Alpha level

- Alpha levels are controlled by the researcher and are related to confidence levels. You get an alpha level by subtracting your confidence level from 100%.
- For example, if you want to be 98 percent confident in your research, the alpha level would be 2% (100% – 98%).
- When you run the hypothesis test, the test will give you a value for p. Compare that value to your chosen alpha level. For example, let’s say you chose an alpha level of 5% (0.05). If the results from the test give you:
- **A small p** (≤ 0.05), reject the null hypothesis. This is strong evidence that the null hypothesis is invalid.
- **A large p** (> 0.05) means the alternate hypothesis is weak, so you do not reject the null.

# What if I Don’t Have an Alpha Level?

- In an ideal world, you’ll have an alpha level. But if you do not, you can still use the following rough guidelines in deciding whether to support or reject the null hypothesis:
- If p > .10 → “not significant”
- If p ≤ .10 → “marginally significant”
- If p ≤ .05 → “significant”
- If p ≤ .01 → “highly significant.”
- **When you perform a statistical test a *p*-value helps you determine the significance of your results in relation to the null hypothesis.**



A **p-value** (shaded green area) is the probability of an observed (or more extreme) result assuming that the null hypothesis is true.

Samatrix.io

- To check whether an independent variable (predictor) is significant or not for the prediction of the target variable. Two common methods for this are:

**By the use of p-values:**

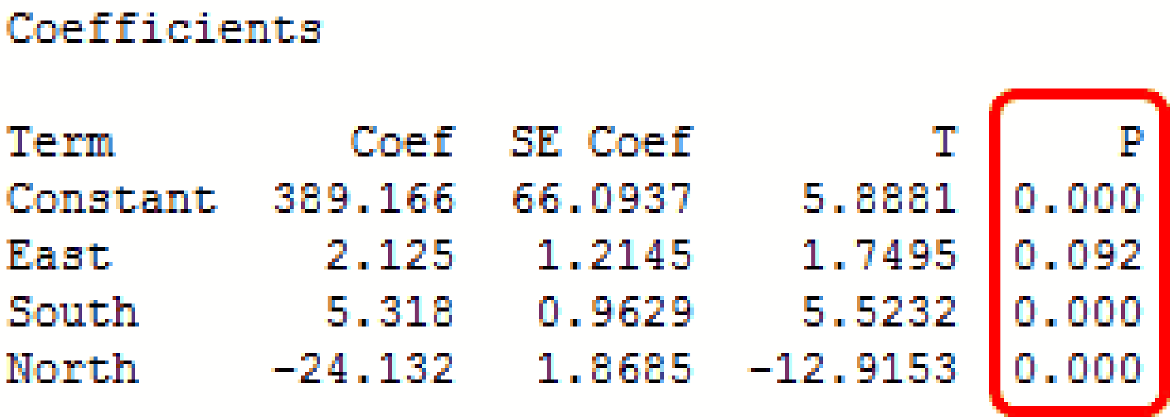
- If the p-value of a particular independent variable is greater than a certain threshold (usually 0.05), then that independent variable is insignificant for the prediction of the target variable.

**By checking the values of the regression coefficient:**

- If the value of the regression coefficient corresponding to a particular independent variable is zero, then that variable is insignificant for the predictions of the dependent variable and has no linear relationship with it.
- To verify whether the calculated regression coefficients i.e, with the help of linear regression algorithm, are good estimators or not of the actual coefficients.

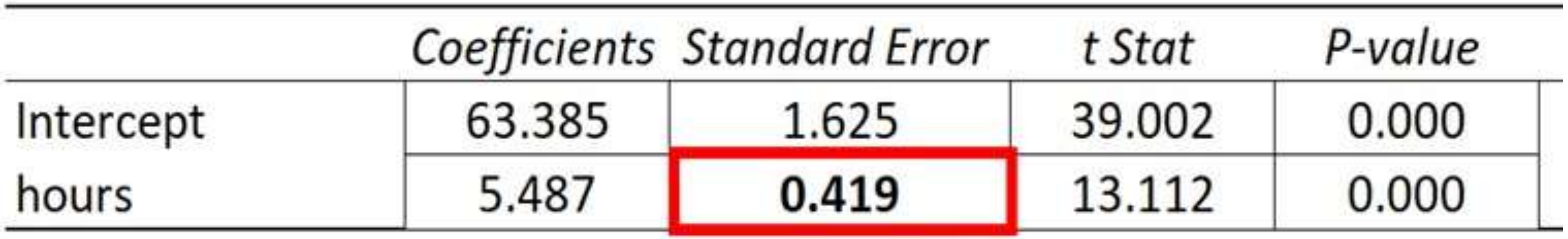
# How do I Interpret the  p-values in Linear Regression?

- The p-value for each term tests the null hypothesis that the coefficient is equal to zero (no effect).
- A low p-value (< 0.05) indicates that you can reject the null hypothesis.
- Conversely, a larger (insignificant) p-value suggests that changes in the predictor are not associated with changes in the response.
- In the output below, we can see that the predictor variables of South and North are significant because both of their p-values are 0.000. However, the p-value for East (0.092) is greater than the common alpha level of 0.05, which indicates that it is not statistically significant.



```
Coefficients

Term          Coef  SE Coef        T      P
Constant   389.166  66.0937   5.8881  0.000
East         2.125   1.2145   1.7495  0.092
South        5.318   0.9629   5.5232  0.000
North      -24.132   1.8685 -12.9153  0.000
```



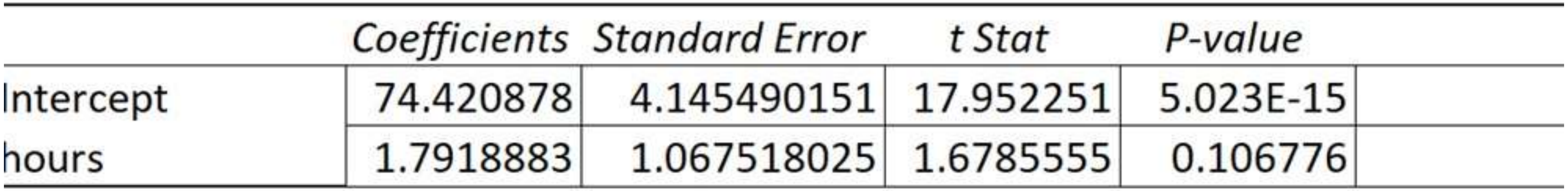
| | *Coefficients* | *Standard Error* | *t Stat* | *P-value* |
|---|---|---|---|---|
| Intercept | 63.385 | 1.625 | 39.002 | 0.000 |
| hours | 5.487 | **0.419** | 13.112 | 0.000 |

- The coefficient for the predictor variable ‘hours studied’ is 5.487. This tells us that each additional hour studied is associated with an average increase of 5.487 in exam score.

- The standard error is 0.419, which is a measure of the variability around this estimate for the regression slope.

- We can use this value to calculate the t-statistic for the predictor variable ‘hours studied’:
- t-statistic = coefficient estimate / standard error
- t-statistic = 5.487 / .419
- t-statistic = 13.112

- The p-value that corresponds to this test statistic is 0.000, which indicates that ‘hours studied’ has a statistically significant relationship with final exam score.
- Since the standard error of the regression slope was small relative to the coefficient estimate of the regression slope, the predictor variable was statistically significant.



| | Coefficients | Standard Error | t Stat | P-value |
|---|---|---|---|---|
| Intercept | 74.420878 | 4.145490151 | 17.952251 | 5.023E-15 |
| hours | 1.7918883 | 1.067518025 | 1.6785555 | 0.106776 |

- The coefficient for the predictor variable ‘hours studied’ is 1.7919. This tells us that each additional hour studied is associated with an average increase of **1.7919** in exam score.
- The standard error is **1.0675**, which is a measure of the variability around this estimate for the regression slope.
- We can use this value to calculate the t-statistic for the predictor variable ‘hours studied’:
- t-statistic = coefficient estimate / standard error
- t-statistic = 1.7919 / 1.0675
- t-statistic = 1.678
- The p-value that corresponds to this test statistic is 0.107. Since this p-value is not less than .05, this indicates that ‘hours studied’ does not have a statistically significant relationship with final exam score.
- Since the standard error of the regression slope was large relative to the coefficient estimate of the regression slope, the predictor variable was *not* statistically significant.

# Regression Using sklearn

```
import sklearn.linear_model as
skl_lm
```

```
In [30]: regr =
skl_lm.LinearRegression()


In [31]: X = advertising.TV.values.reshape(-
1,1)


In [32]: y =
advertising.Sales


In [33]:
regr.fit(X,y)


Out[33]: LinearRegression()


In [34]:
regr.intercept_
```

# RSS & MSE

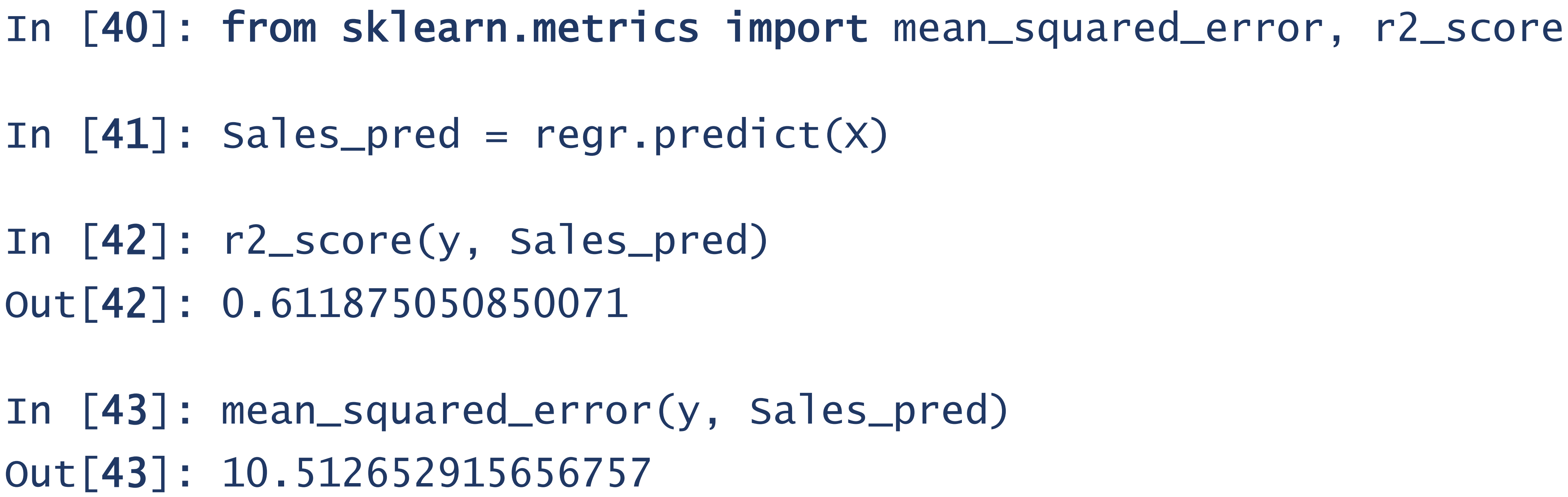
```
In [36]: min_rss = np.sum((regr.intercept_+regr.coef_*X
- y.values.reshape(-1,1))**2)

In [37]: min_rss
Out[37]: 2102.5305831313514


In [38]: mse = min_rss/len(y)


In [39]: mse
Out[39]: 10.512652915656757
```

# MSE, R-Sq Using Sklearn



```
In [40]: from sklearn.metrics import mean_squared_error, r2_score

In [41]: Sales_pred = regr.predict(X)

In [42]: r2_score(y, Sales_pred)
Out[42]: 0.611875050850071

In [43]: mean_squared_error(y, Sales_pred)
Out[43]: 10.512652915656757
```

# Regression Summary using statsmodels

```
In [44]: import statsmodels.formula.api as smf
In [45]: est = smf.ols('Sales ~ TV', advertising).fit()
In [46]:
est.summary()

Out[46]:
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.612
Model:                            OLS   Adj. R-squared:                  0.610
Method:                 Least Squares   F-statistic:                     312.1
Date:                Mon, XX Apr XXXX   Prob (F-statistic):           1.47e-42
Time:                        11:49:06   Log-Likelihood:                -519.05
No. Observations:                 200   AIC:                             1042.
Df Residuals:                     198   BIC:                             1049.
Df Model:                           1
Covariance Type:            nonrobust
```

# Regression Summary using statsmodels

```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      7.0326      0.458     15.360      0.000       6.130       7.935
TV             0.0475      0.003     17.668      0.000       0.042       0.053
==============================================================================
Omnibus:                        0.531   Durbin-Watson:                   1.935
Prob(Omnibus):                  0.767   Jarque-Bera (JB):                0.669
Skew:                          -0.089   Prob(JB):                        0.716
Kurtosis:                       2.779   Cond. No.                         338.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

# Regression RSS, MSE Using statsmodels

```
In [47]: est.params
Out[47]:
Intercept    7.03
TV           0.05
dtype: float64
```

**RSS**

```
In [48]: ((advertising.Sales - (est.params[0] + est.params[1]
* advertising.TV))** 2).sum()

Out[48]: 2102.5305831313512
```

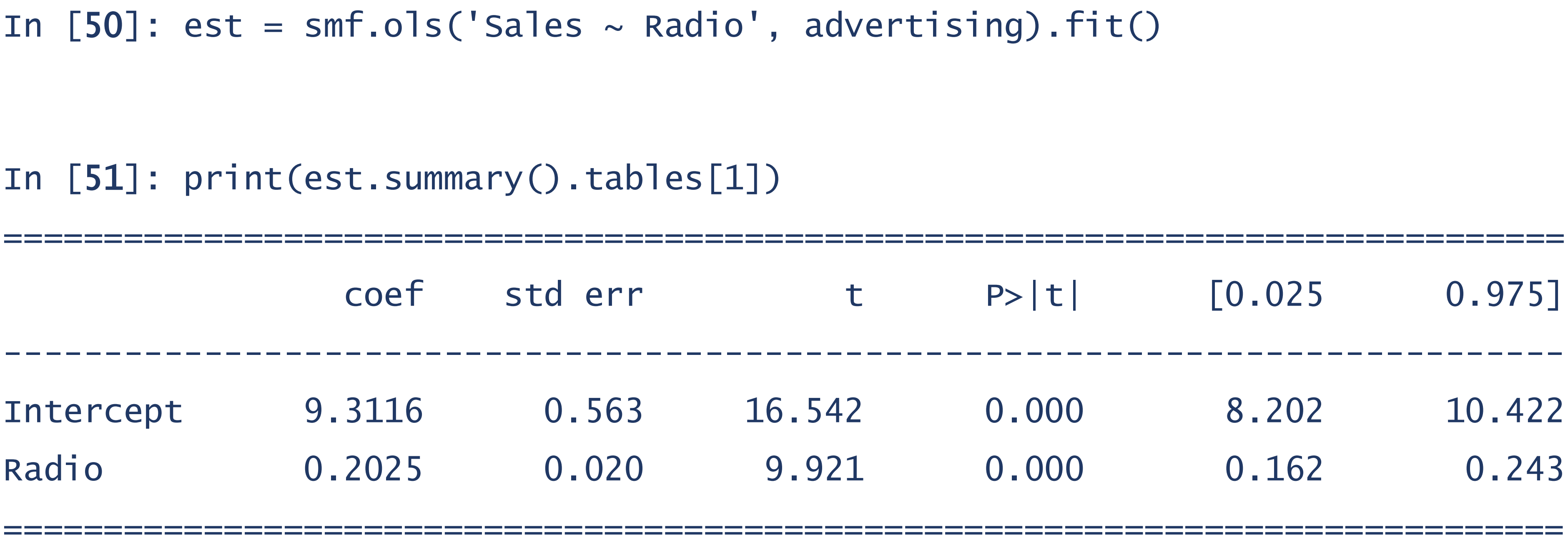
# Regression RSS, MSE Using statsmodels

## MSE

```
In [49]: ((advertising.Sales - (est.params[0] +
est.params[1]*advertising.TV))** 2).sum()/len(advertising.Sales)


Out[49]: 10.512652915656757
```

# Linear Regression for Radio



```
In [50]: est = smf.ols('Sales ~ Radio', advertising).fit()


In [51]: print(est.summary().tables[1])
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      9.3116      0.563     16.542      0.000       8.202      10.422
Radio          0.2025      0.020      9.921      0.000       0.162       0.243
==============================================================================
```

Check the p value of Intercept and Radio.

It shows that there is a relationship between Sales and Radio

# Linear Regression for Newspaper

```
In [52]: est = smf.ols('Sales ~ Newspaper', advertising).fit()

In [53]: print(est.summary().tables[1])
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     12.3514      0.621     19.876      0.000      11.126      13.577
Newspaper      0.0547      0.017      3.300      0.001       0.022       0.087
==============================================================================
```

Check the p value of Intercept and Newspaper.

It shows that there is a relationship between Sales and Newspaper

# Multiple Linear Regression

# Multiple Linear Regression

In a three-dimensional setting, with two predictors and one response, the least squares regression line becomes a plane.

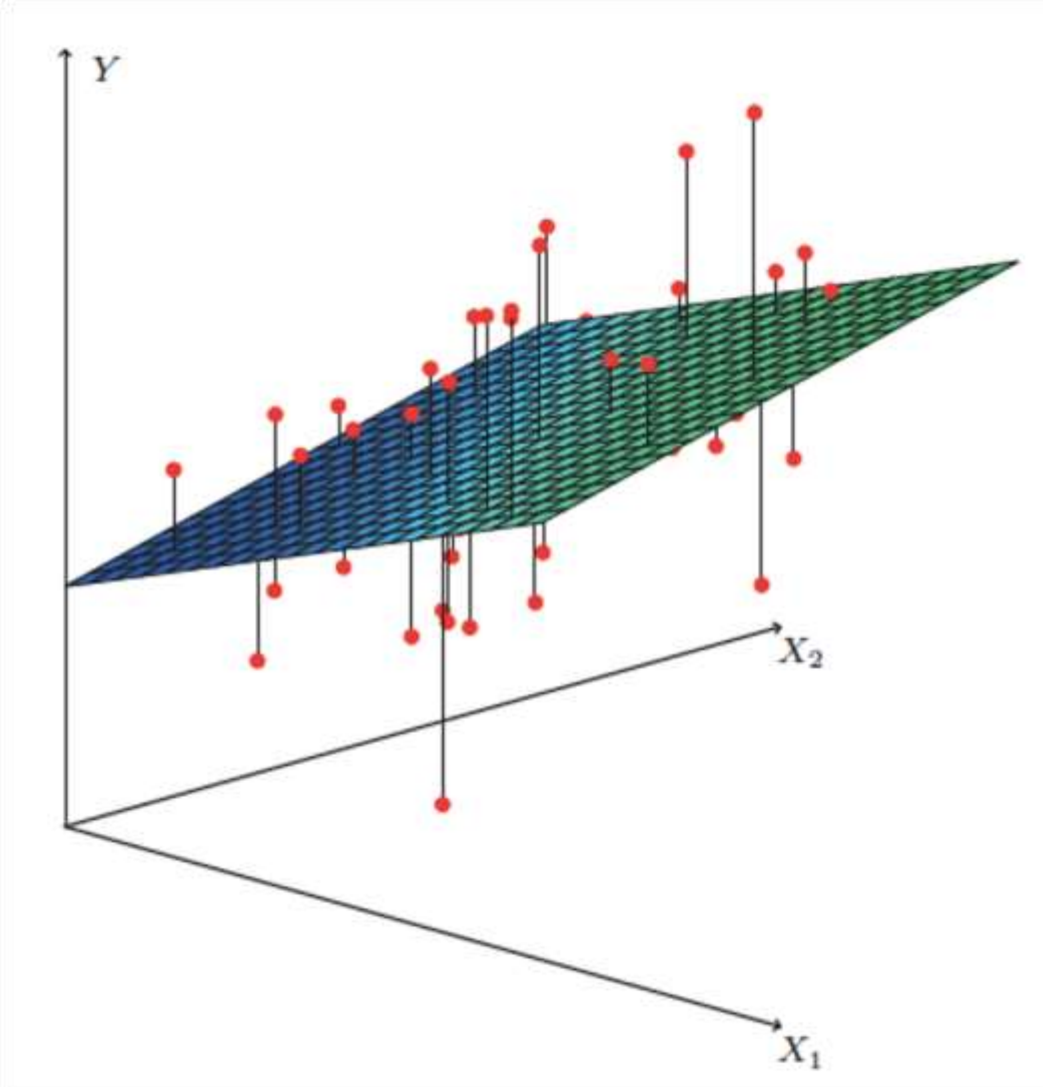
The plane is chosen to minimize the sum of the squared vertical distances between each observation (shown in red) and the plane.

The multiple regression model is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

The model for this project is

$$sales = \beta_0 + \beta_1 \times \ TV + \beta_2 \times \ Radio + \beta_3 \times \ Newspaper + \epsilon$$



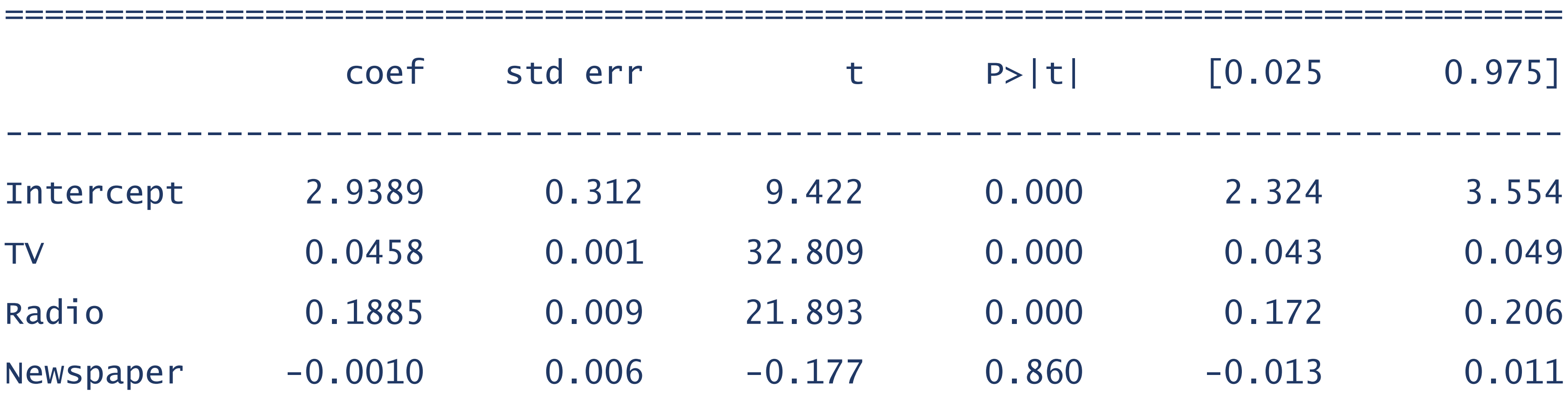
# Multiple Linear Regression

```
In [54]: est = smf.ols('Sales ~ TV + Radio + Newspaper',
advertising).fit()
```

```
In [55]:
est.summary()
```

```
Out[55]:
<class 'statsmodels.iolib.summary.Summary'
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.897
Model:                            OLS   Adj. R-squared:                  0.896
Method:                 Least Squares   F-statistic:                     570.3
Date:                Tue, xx Apr xxxx   Prob (F-statistic):           1.58e-96
Time:                        17:30:45   Log-Likelihood:                -386.18
No. Observations:                 200   AIC:                             780.4
Df Residuals:                     196   BIC:                             793.6
Df Model:                           3
Covariance Type:            nonrobust
```

# Multiple Linear Regression



```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.9389      0.312      9.422      0.000       2.324       3.554
TV             0.0458      0.001     32.809      0.000       0.043       0.049
Radio          0.1885      0.009     21.893      0.000       0.172       0.206
Newspaper     -0.0010      0.006     -0.177      0.860      -0.013       0.011
```

Check the p value of TV, Radio and Newspaper.

# Correlation

```
In [56]: advertising.corr()
Out[56]:
                 TV  Radio  Newspaper  Sales
TV             1.00   0.05       0.06   0.78
Radio          0.05   1.00       0.35   0.58
Newspaper      0.06   0.35       1.00   0.23
Sales          0.78   0.58       0.23   1.00
```

# Thanks

Samatrix Consulting Pvt Ltd

Samatrix.io