

The screenshot displays a Jupyter Notebook titled "Understanding Simple Linear Regression" with the last checkpoint at 08/16/2022. The notebook is running on Python 3 (ipykernel). The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with various icons for file operations and execution.

The first code cell contains the following text:

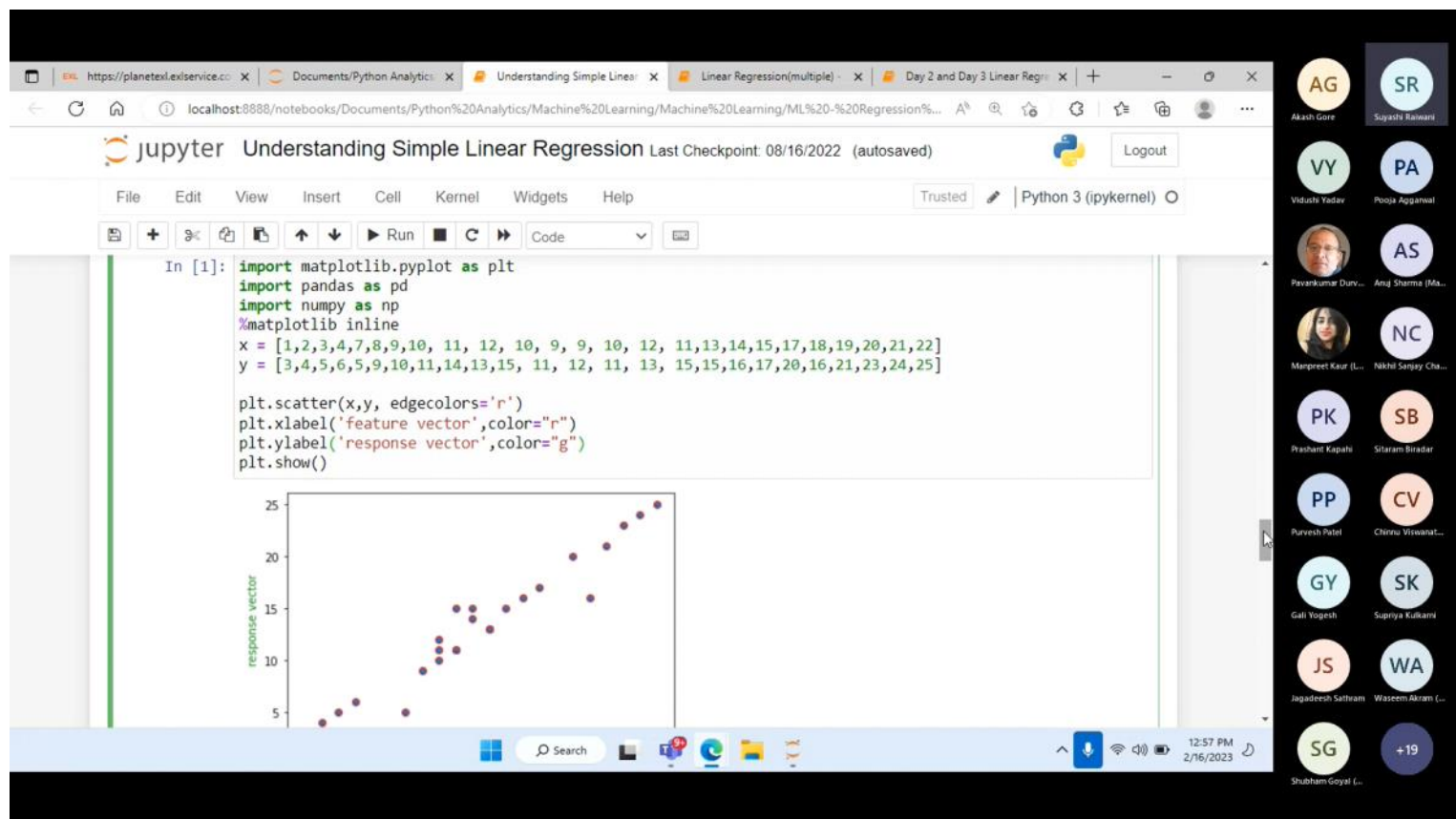
```
In [ ]: Predict outcome - single outcome (Y)- numerical outcome
Independent variables : x1,x2
Dependendt on independent

1. Popultaion = Year
2. Salary APPRAISAL = WP+ BEHAVE+ TENURE IN COM+cc
LINEAR RELATION -
pROFIT = C(Sales)+ C1
profit = c(sales)+b(dis)+c2
```

The second code cell contains the following Python code:

```
In [ ]: import numpy as np
a=np.array([1,2,3])
a+a
a*a
a/a
sum(a)
np.mean(a)
np.size(a)
```

The bottom of the screen shows a Windows taskbar with the time 5:52 / 1:30:01 and a system tray with various icons.



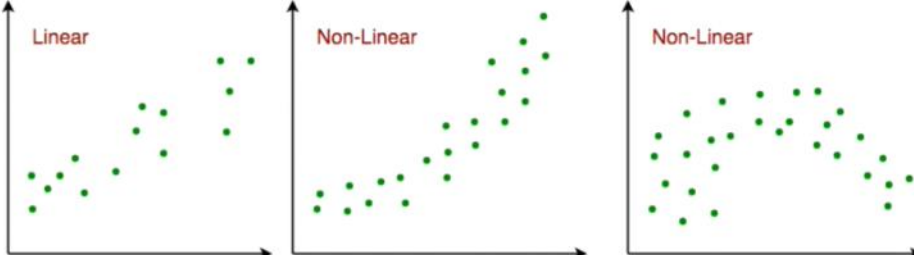
Draw a scatter plot and check the curve to check if linear reg can be applied

Linear Regression(multiple) Last Checkpoint: 06/09/2022 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

• Simple Linear Regression

Simple linear regression is an approach for predicting a response using a single feature.



The figure shows three scatter plots side-by-side. The first plot is labeled 'Linear' and shows a positive linear trend. The second plot is labeled 'Non-Linear' and shows a non-linear, upward-curving trend. The third plot is also labeled 'Non-Linear' and shows a non-linear, downward-curving trend. All plots have green data points.

WHY Linear Regression?

- To find the parameters so that the model **best fits** the data.
- Forecasting an effect

1:03 PM 2/16/2023

Linear Regression(multiple) Last Checkpoint: 06/09/2022 (autosaved)

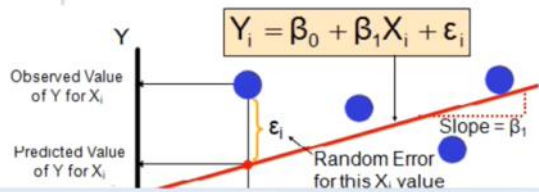
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

WHY Linear Regression?

- To find the parameters so that the model **best fits** the data.
- Forecasting an effect
- Determining a Trend

How do we determine the best fit line?

- The line for which the error between the **predicted** values and the **observed** values is minimum is called the best fit line or the regression line. These errors are also called as **residuals**.
- The residuals can be visualized by the vertical lines from the observed data value to the **regression line**.



The diagram shows a scatter plot with a red regression line. A blue data point is shown above the line. A vertical line connects the point to the regression line, labeled ϵ_i . The regression line is labeled with the equation $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ and the slope $\text{Slope} = \beta_1$. The y-axis is labeled 'Y' and the x-axis is labeled 'X'. The observed value of Y for X_i is shown as a blue dot, and the predicted value of Y for X_i is shown as a red dot on the regression line.

1:04 PM 2/16/2023

<https://planetex.lxservice.co>
[Documents/Python Analytics](#)
[Understanding Simple Linear](#)
[Linear Regression\(multiple\)](#)
[Day 2 and Day 3 Linear Regr](#)

localhost:8888/notebooks/Documents/Python%20Analytics/Machine%20Learning/Machine%20Learning/Machine%20Learnin...

jupyter

Linear Regression(multiple) Last Checkpoint: 06/09/2022 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3 (ipykernel)

random_state=1)]

Feature Importance for selection

One of the assumptions of linear regression is that the independent variables need to be uncorrelated with each other. If these variables are correlated with each other, then we need to keep only one of them and drop the rest.

The correlation coefficient has values between -1 to 1

- A value closer to 0 implies weaker correlation (exact 0 implying no correlation)
- A value closer to 1 implies stronger positive correlation
- A value closer to -1 implies stronger negative correlation

In [46]:

```
#Using Pearson Correlation
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
sns.set()
plt.figure(figsize=(10,10))
cor = c_data.corr()
sns.heatmap(cor, annot=True, cmap=plt.cm.Red)
```

1:33 PM 2/16/2023

SG Shubham Goyal

SR Supriya Raiwar

AG Akash Gore

VY Vidushi Yadav

P Purnesh

PA Pooja Appanwal

NC Nikhil Sanjay Ch...

SK Supriya Kulkarni

SM Sushrut Marathe

Raakesh Sharma

Manpreet Kaur (L...

RV Ravi Verma (Man...

AD Arushi Dudeja

MV Manjunatha V

SS Sheetal Sahu

PK Prashant Kapahi

+15

<https://planetex.lxservice.co>
[Documents/Python Analytics](#)
[Understanding Simple Linear](#)
[Linear Regression\(multiple\)](#)
[Day 2 and Day 3 Linear Regr](#)

localhost:8888/notebooks/Documents/Python%20Analytics/Machine%20Learning/Machine%20Learning/Machine%20Learnin...

jupyter

Linear Regression(multiple) Last Checkpoint: 06/09/2022 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3 (ipykernel)

- A value closer to -1 implies stronger negative correlation

In [46]:

```
#Using Pearson Correlation
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
sns.set()
plt.figure(figsize=(10,10))
cor = c_data.corr()
sns.heatmap(cor, annot=True, cmap=plt.cm.Red)
plt.show()
```

	ID	Price	vy
ID	1	-0.00084	0.1
Price	-0.00084	1	-0.0037
vy	0.1	-0.0037	1

1:35 PM 2/16/2023

SG Shubham Goyal

SR Supriya Raiwar

AG Akash Gore

VY Vidushi Yadav

P Purnesh

PA Pooja Appanwal

NC Nikhil Sanjay Ch...

SK Supriya Kulkarni

SM Sushrut Marathe

Raakesh Sharma

Manpreet Kaur (L...

RV Ravi Verma (Man...

AD Arushi Dudeja

MV Manjunatha V

SS Sheetal Sahu

PK Prashant Kapahi

+15

PLOT HEATMAP TO CHECK THE CORRELATION--NOT A GOOD ENOUGH METHOD

regression Page 3

https://planetexl.exiservice.co...Documents/Python Analytics...Understanding Simple Linea...Linear Regression(multiple)...Day 2 and Day 3 Linear Regr...+--

localhost:8888/notebooks/Documents/Python%20Analytics/Machine%20Learning/Machine%20Learning/Machine%20Learning%20...A...Logout

Jupyter Linear Regression(multiple) Last Checkpoint: 06/09/2022 (autosaved)

FileEditViewInsertCellKernelWidgetsHelpTrustedPython 3 (ipykernel)

relevant_features

Out[47]: Price 1.0
Name: Price, dtype: float64

RFE (Recursive Feature Elimination)

The Recursive Feature Elimination (RFE) method works by recursively removing attributes and building a model on those attributes that remain. It uses accuracy metric to rank the feature according to their importance. The RFE method takes the model to be used and the number of required features as input. It then gives the ranking of all the variables, 1 being most important. It also gives its support, True being relevant feature and False being irrelevant feature.

In [49]: X.head()

Out[49]:

	Levy	Leather interior	Mileage	Cylinders	Airbags
0	0	0	200000	4	2
1	0	1	128500	6	12
2	0	0	0	8	0
3	0	0	350000	4	4
4	0	0	230000	4	0

The goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features

Search

1:38 PM
2/16/2023

SGSR
Shubham Goyal (...)
Sugata Rayward

AGVY
Akash Gore
Vidushi Yadav

PPA
Punesh
Pooja Aggarwal

NC
Pavankumar Duv...
Nikhil Sanjay Chu...

SSWA
Sheetal Sahu
Wassem Akram (...)

ADM
Anushi Dadeja
Manpreet Kaur (...)

RVSK
Ravi Verma (Man...
Supriya Kulkarni

SMPK
Sushrut Marathe
Prashant Kapale

+14
Rakesh Sharma

regression Page 4