

# Regression Plots

## Linear Regression

- linear regression computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation to observed data.
- evaluates whether a set of predictor variables do a good job in predicting an outcome (dependent) variable

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
sns.set(palette='deep',rc={"figure.figsize": (5, 4)})
```

### LM plot vs Reg plot

regplot is used for simple scatter plots with a regression line, ideal for quick analysis. Implot offers more flexibility with faceting, allowing you to create complex visualizations by splitting data into subplots.

### LM Plot

```
In [ ]: sns.set_style('whitegrid')
```

```
In [ ]: # Loading dataset
tips=sns.load_dataset('tips')
sns.lmplot(x='total_bill',y='tip',data=tips,palette='Blues')
```

```
In [ ]: sns.lmplot(x='total_bill',y='tip',hue='smoker',data=tips,palette='deep')
```

changing markers

```
In [ ]: sns.lmplot(x="total_bill", y="tip", hue="smoker", data=tips, markers=["o", "x"])
```

seperate plots

```
In [ ]: sns.lmplot(x="total_bill", y="tip", row='sex', col="smoker", data=tips)
```

## Reg Plot

```
In [ ]: sns.regplot(x="total_bill", y="tip", data=tips, color="g")
```

## Correlation

- show how much one variable is affected by presence of another variable.
- statistically represents relationship between two variables
- range is -1 to 1

## Scatter plot

(lm plot) used for

- correlation
- interpolation - try to find a value inside our set of available data points
- extrapolation- we find a value outside our set of data points

## Joint plot

```
In [ ]: # Loading Built-in Dataset:  
tips = sns.load_dataset("tips")  
sns.jointplot(x="total_bill", y="tip", data=tips, color='b')
```

```
In [ ]: sns.jointplot(x="total_bill", y="tip", data=tips, kind="reg")
```

adding a specif point

```
In [ ]: a=sns.jointplot(x="total_bill", y="tip", data=tips, color='b')  
a.ax_joint.plot([0.8],[0.8], 'bx', color='g')  
a.ax_joint.text(0.8,0.8, "Random point")
```

## Letter Value (LV) Plot

non-parametric estimates of the distribution of a dataset

```
In [ ]: titanic=sns.load_dataset('titanic')
```

```
In [ ]: sns.boxenplot(x="day", y="total_bill", hue="smoker", data=tips)
```

```
In [ ]:
```