||JAI SRI GURUDEV||
Sri Adichunchanagiri Shikshana Trust ®
SJB INSTITUTE OF TECHNOLOGY
No. 67, BGS Health & Education City, Dr. Vishnuvardhan Road, Kengeri, Bengaluru - 560 060
(Affiliated to Visvesvaraya Technological University, Belagavi& Approved by AICTE, New Delhi, accredited by NAAC, Accredited by NBA)
Department of Computer Science & Engineering

Module 5: Hypothesis Testing and Regression

Hypothesis testing

Hypothesis testing is often used to facilitate statistical decisions using experimental datasets. The testing is used to validate assumptions about a population parameter.

Examples:

- The average score of students taking the Machine Learning course at the University of Nepal is 78.
- The average height of boys is higher than that of girls among the students taking the Machine Learning course.

“A hypothesis test evaluates two competing statements about a population and determines which one is supported by the sample data.”

Key terms:

- **Population:** All elements in a data set.
- **Sample:** A subset of observations taken from the population.

Hypothesis Testing Principles

Normalization

Normalization is the process of adjusting values measured on different scales to a common scale. This step ensures that data is comparable before performing descriptive statistics.

$$X_{changed} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Standard Normalization

Standard normalization is a specific form of normalization where the data is adjusted to have a mean of 0 and a standard deviation of 1.

$$X_{changed} = \frac{X - \mu}{\sigma}$$

Important Concepts in Hypothesis Testing

- **Null Hypothesis (H_0)**
  The null hypothesis is a basic assumption about a population. For example, the average typing speed of a person is 38-40 words per minute.

- **Alternative Hypothesis (H_1)**
  The alternative hypothesis contradicts the null hypothesis. For example, the average typing speed is less than 38-40 words per minute. The goal is to test whether we should accept or reject this hypothesis based on the data.

Errors in Hypothesis Testing

- **Type I Error (False Positive):**
  Rejecting the null hypothesis when it is actually true.

- **Type II Error (False Negative):**
  Failing to reject the null hypothesis when it is false.

Key Parameters in Hypothesis Testing

- **P-value**
  The p-value is the probability of obtaining a result given that the null hypothesis is true. If the p-value is lower than a predetermined threshold, we reject the null hypothesis.

- **Level of Significance (α)**
  The level of significance is the threshold used to determine whether to reject the null hypothesis. A common level of significance is 0.05 (5%), meaning there is a 95% confidence in the result supporting the null hypothesis.

  To summarize, see the condition before either selecting or rejecting the null hypothesis:

  - Reject H0 if p <= α
  - Accept the null hypothesis if p > α

statsmodels library

Imagine you're a detective. You suspect that more than 48% of parents believe social media causes stress in their teens. But before jumping to conclusions, you need solid proof — that's what hypothesis testing is for!

The Setup

- **Population**: All parents with teens aged 18 or older.
- **Question**: Do *more than 48%* of them believe social media causes stress?

We collected data from **4,500 parents**, and **65%** of them said yes.

What Are We Trying to Find Out?
We want to check:
Is this new 65% result so high that it can't just be due to chance?
Or could it still happen naturally if the real percentage is still just 48%?
This is **what hypothesis testing answers**.

Hypotheses

- **Null Hypothesis (p)**: Only 48% of parents believe social media is the cause.
  → p = 0.48

- **Alternative Hypothesis (phat)**: More than 48% believe so.
  → phat > 0.48

This is a right-tailed test, because we're checking if the percentage is greater than 48%.

The Data

- **Sample size (n)** = 4500
- **Sample proportion (phat)** = 0.65
- **Null proportion (p)** = 0.48

We want to know: Is 65% high enough to say it’s not just random chance?

Code:

```
import statsmodels.api as sm
n = 4500
pnull = 0.48
phat = 0.65
sm.stats.proportions_ztest(phat * n, n, pnull, alternative='larger')
```

What it does:

- Counts how many parents said yes: 0.65 * 4500 = 2925
- Runs a **Z-test** to compare 65% vs 48%.

Output:

```
(23.91, 1.22e-126)
```

Z-statistic = 23.91: Huge! Means the difference is far from expected.
P-value = 1.22e-126: Almost 0.

There is **strong evidence** that **more than 48%** of parents believe social media is causing stress in their teens.

0.65 is clearly bigger than 0.48, why do we need to run a hypothesis test?

Because not every difference is meaningful!!!

Imagine This:
Suppose you flip a coin 10 times and it lands heads 7 times.

That’s 70% heads. Is the coin unfair?

Maybe.
But maybe not — it’s a small sample. With just 10 flips, 7 heads could easily happen by chance.

Now flip the coin 1,000 times, and you get 700 heads.
Same proportion — 70%.
Now that’s suspicious!

Big differences need **context** — especially sample size — to know if they are *statistically significant* or just *random noise*.

Back to Our Case: 0.65 vs 0.48
Yes, **0.65 is bigger than 0.48**.
But:

- Was it just due to **random sampling**?
- Or is this difference **real and significant**?

Even a small difference can be significant if the sample is **large** (which it is here — 4,500 parents).
Likewise, a large-looking difference might be **insignificant** if the sample is small.
That's why we run the **z-test**:
To ask: *"How likely is it to get 0.65 if the true value is still 0.48?"*

And the Answer from the Test:

- The **z-score was 23.91** → a HUGE number.
- The **p-value was tiny** → almost 0.

👉 So yes — **0.65 is not just bigger** — it's **significantly bigger**, and **not due to random chance**.

Case Study: Average Reading Time

Problem:

A reading competition was conducted with some adults. The data collected represents their reading times in words per minute:

Data:
[236, 239, 209, 246, 246, 245, 215, 212, 242, 241, 219, 242, 236, 211, 216, 214, 203, 223, 200, 238, 215, 227, 222, 204, 200, 208, 204, 230, 216, 204, 201, 202, 240, 209, 246, 224, 243, 247, 215, 249, 239, 211, 227, 211, 247, 235, 200, 240, 213, 213, 209, 219, 209, 222, 244, 226, 205, 230, 238, 218, 242, 238, 243, 248, 228, 243, 211, 217, 200, 237, 234, 207, 217, 211, 224, 217, 205, 233, 222, 218, 202, 205, 216, 233, 220, 218, 249, 237, 223]

Hypothesis Question:

Is the average reading speed of random students (adults) more than 212 words per minute?

Step-by-Step Solution:

1. **Population**:
   All adults participating in the competition.

2. **Parameter of Interest**:
   μ (the average reading speed of adults).

3. **Hypotheses**:

   - **Null Hypothesis (H_0):** $\mu = 212$ (The average reading speed is 212 words per minute.)

   - **Alternative Hypothesis (H_1):** $\mu > 212$ (The average reading speed is greater than 212 words per minute.)

4. **Confidence Level**:
   $\alpha = 0.05$ (This means we are 95% confident in our decision to either accept or reject the null hypothesis.)

Z-Test Calculation:

We can perform a Z-test to test our hypothesis. Using Python and the `statsmodels` package, the code is as follows:

```
import numpy as np
import statsmodels.api as sm

# Sample data
sdata = np.random.randint(200, 250, 89)

# Z-test for hypothesis: μ > 212
sm.stats.ztest(sdata, value=212, alternative="larger")
```

Output:

```
(91.63511530225408, 0.0)
```

- **Z-Statistic**: 91.64

- **P-value**: 0.0

Conclusion:

Since the **P-value (0.0)** is much lower than the **significance level (α = 0.05)**, we **reject the null hypothesis**. This means that the average reading speed of adults is significantly greater than 212 words per minute.

Types of Hypothesis Testing

There are different types of hypothesis testing:

- Z-test: Used to compare sample and population means when the population variance is known.
  - A factory knows the average weight of chips in a packet should be 100g. They test a sample to see if the machine is still packing correctly.
  - They are comparing a sample mean to a known population mean.
- T-test: Used to compare sample means when the population variance is unknown.
  - A teacher wants to know if boys and girls in her class scored differently on a test.
  - The teacher is comparing the means of *two* independent groups (boys and girls) to see if there's a significant difference.
- ANOVA test: Used to compare means across three or more groups.
  - A doctor wants to compare the effect of three different diets on weight loss.
  - ANOVA (Analysis of Variance) is specifically designed for comparing multiple group means.
- Chi-squared test: Used to test relationships between categorical variables.
  - A store wants to know if people’s shoe preferences (sneakers, sandals, boots) depend on their age group.
  - The chi-squared test is used to analyze categorical data

T-test

The T-test is a statistical method used in *inferential statistics* to determine if there is a significant difference between the means (averages) of **two groups**.

Examples:

- Comparing average marks of students in two different classes.
- Checking if a new teaching method is more effective than the traditional one.
- Testing if two brands of painkillers provide the same level of relief.

When to use:

- When the sample size is small (typically less than 30).
- When the population standard deviation is unknown.
- When data is approximately normally distributed.

Case Study:

Context - We have a dataset of students from certain classes. The dataset contains the height of each student. We are checking whether the average height is 175 cm or not.

- **Population**: All students in that class
- **Parameter of interest**: μ, the population of a classroom (actual avg height of the class)
- **Null hypothesis**: The average height is $\mu = 175$
- **Alternative hypothesis**: $\mu > 175$
- **Confidence level:** $\alpha = 0.05$

Step 1: Let's first set up the dataset

```
import numpy as np
from scipy.stats import ttest_1samp

height = np.array([172, 184, 174, 168, 174, 183, 173, 173, 184, 179, 171, 173, 181,
183, 172, 178, 170, 182, 181, 172, 175, 170, 168, 178, 170, 181, 180, 173, 183, 180,
177, 181, 171, 173, 171, 182, 180, 170, 172, 175, 178, 174, 184, 177, 181, 180, 178,
179, 175, 170, 182, 176, 183, 179, 177])
```

Step 2: Calculate average

```
height_avg = np.mean(height)
print(f"Average height = {height_avg:.2f}")
```

Step 3: Run T-test

```
t_stat, p_val = ttest_1samp(height, 175)
print("P-value =", p_val)
```

Step 4: Decision rule

```
if p_val < 0.05:
   print("Reject the null hypothesis")
else:
   print("Accept the null hypothesis")
```

Output:

```
Average height = 176.55
P-value = 0.019403337788027563
Reject the null hypothesis
```

Explanation:

`ttest_1samp(data, popmean)`

- Tests whether the average (mean) of your sample data is significantly different from a known/hypothesized population mean.

What is P-Hacking?

P-hacking is the misuse of statistical analysis to artificially produce statistically significant results (typically $p < 0.05$).

Also known as:

- Data fishing
- Data dredging
- Data butchery

Why Does It Happen?

- **Hypothesis testing** relies on the **p-value** to decide whether a result is significant.
- A **low p-value (< 0.05)** suggests a real effect, attracting attention from journals and readers.
- **Problem:** Sometimes, researchers test many hypotheses and **only report the ones that show significance**, even if they're due to **random chance**.

Consequences

- **False positives**: Claiming there's an effect when there isn’t.
- **Misleading conclusions** in research.
- Undermines the **credibility** of scientific findings.
- Leads to **publication bias** — only results with "positive" findings are shared.

Especially common in fields where:

- There’s pressure to publish
- Only significant results are accepted
- Large datasets are explored without a clear hypothesis

How to Avoid P-Hacking

- Pre-register your hypothesis and analysis plan.
- Report all results, not just the significant ones.
- Use corrections for multiple comparisons (e.g., Bonferroni correction).
- Be transparent about your methods and decisions.

Understanding Regression

What is Regression?

- Regression is a statistical method used to model the relationship between variables.
- Helps predict the value of one quantitative variable (dependent) based on another (independent).

Key Concepts

- **Y**: Dependent variable (Outcome or what you want to predict)
- **X**: Independent variable (Predictor or feature or covariate)
- **a**: Intercept (Value of Y when X = 0)
- **b**: Slope (Change in Y for every unit change in X)
- **u**: Regression Residual (Error term; difference between actual and predicted Y)

Regression Equation:
Y = a + bX + u

Why Use Regression?

- Prediction – e.g., predicting sales based on advertising spend.
- Understanding relationships – e.g., how salary changes with years of experience.
- Forecasting trends – e.g., estimating future growth or demand.

Types of Regression

Two main regression types:

1. Simple linear regression
2. Multiple linear regression

Simple Linear Regression

- Defines the relationship between two variables using a straight line.
- One independent variable (X) is used to predict one dependent variable (Y).

Equation form:
Y = a + bX + u

  - X: Predictor (independent variable)
  - Y: Target (dependent variable)
  - a: Intercept
  - b: Slope
  - u: Error (residual term)

Goal of Linear Regression:

  - Find the best-fitting line (called the regression line) that minimizes the error.
  - The error is the vertical distance between actual data points and the line.

Error Function:

  - Also called loss or cost function.
  - Defined as the sum of squared differences between observed values and predicted values (to avoid positive/negative cancellations).

This method of minimizing the squared errors is called **Ordinary Least Squares (OLS).**

Multiple Linear Regression

- An extension of simple linear regression that models the relationship between one dependent variable (Y) and two or more independent (explanatory) variables (X_1, X_2, ..., Xt).
- Used to capture **more complex linear relationships** in real-world data.

- The dependent variable (Y) is influenced by multiple factors (independent variables).
- Each independent variable contributes linearly to the prediction of Y.

Example: Predicting House Prices

The **price of a house** (Y) depends on multiple factors:

1. Size of the house in square feet (X_1)
2. Number of bedrooms (X_2)
3. Distance to the city center in kilometers (X_3)

This relationship can be modeled as:

$$\mathbf{Y = a + b_1X_1 + b_2X_2 + b_3X_3 + u}$$

Where:

- **Y** = Price of wine (dependent variable)
- **X_1** = Quantity bought
- **X_2** = Time of year
- **X_3** = Inventory level
- **a** = Intercept
- **b_1, b_2, b_3** = Coefficients (slopes) for each independent variable
- **u** = Error term (residual)

General Form:

$$\mathbf{Y = a + b_1X_1 + b_2X_2 + ... + btXt + u}$$

- **Y**: Dependent variable
- **X_i**: Independent variables
- **a**: Intercept
- **b_i**: Coefficients representing effect of each X_i
- **u**: Error term

Nonlinear Regression

Nonlinear regression is a type of regression analysis where the relationship between independent variables (X) and the dependent variable (Y) does not follow a straight line. Instead, it follows a mathematical function that creates a curve.

Equation for nonlinear regression:

Y = f(X, β) + ε

- **X** = A vector of *p* predictors (independent variables)
- **β** = A vector of *k* parameters (model coefficients)
- **f(·)** = A known nonlinear regression function
- **ε** = Error term (residual)

Characteristics:

- Unlike simple linear regression which fits a **straight line**, nonlinear regression fits a **curve**.
- It allows for **complex relationships** between variables.

Common Functions Used:

- Logarithmic functions
- Trigonometric functions
- Exponential functions
- Other mathematical fitting functions

Similarity to Linear Regression:

- Both try to predict Y from a set of X variables
- Both aim to minimize error using Ordinary Least Squares (OLS)

Development Complexity: Nonlinear models are harder to develop because they require a series of iterations, often involving trial and error.

Common solving methods:

- Gauss-Newton method
- Levenberg-Marquardt algorithm

Model Development and Evaluation

We will use the scikit-learn library to implement linear regression and evaluate the model.To do this, we will use the famous Boston housing prices dataset, which is widely used in regression analysis.

We will build a linear regression model in Python using a sample dataset available in scikit-learn, the Boston housing prices dataset.

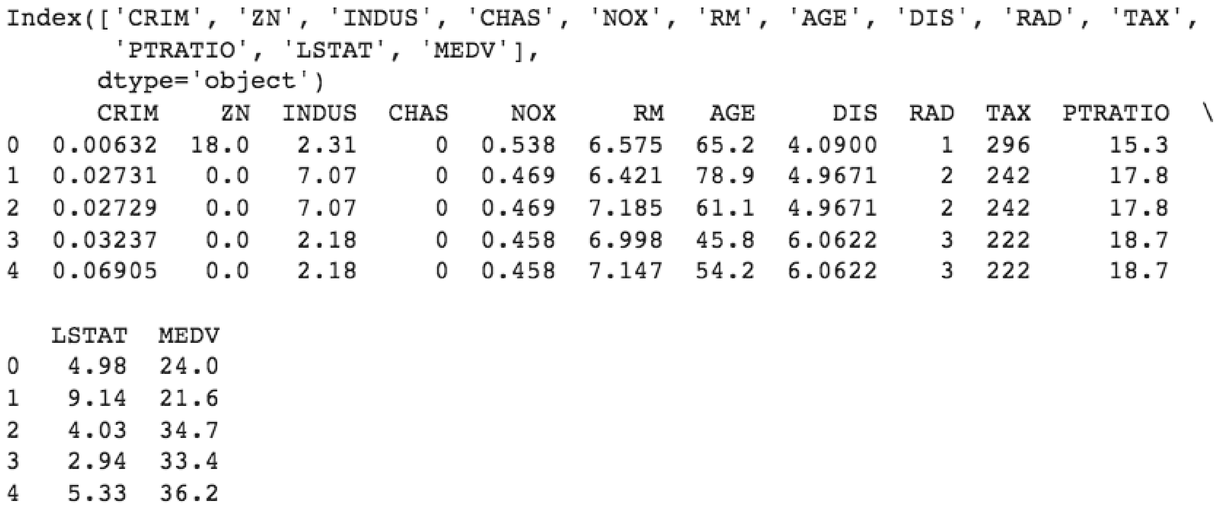
Step 1: Import Necessary Libraries and Load the Dataset

```
# Importing the necessary libraries
# Importing the necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="ticks", color_codes=True)
plt.rcParams['figure.figsize'] = (8, 5)
plt.rcParams['figure.dpi'] = 150

# Loading the dataset from URL
df = pd.read_csv("https://raw.githubusercontent.com/PacktPublishing/hands-on-exploratory-data-analysis-with-python/master/Chapter%209/Boston.csv")
```

Step 2: Explore the Dataset

```
# Check the column names
print(df.columns)
```

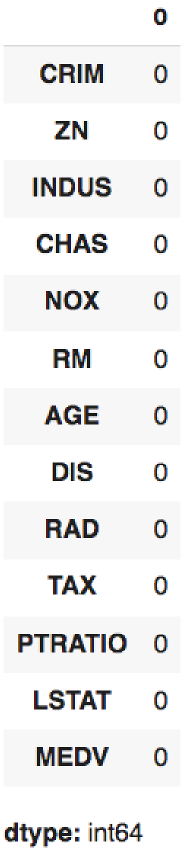
# Display the top 5 rows
print(df.head())



```
Index(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
       'PTRATIO', 'LSTAT', 'MEDV'],
      dtype='object')
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  PTRATIO  \
0  0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1  296     15.3
1  0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2  242     17.8
2  0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2  242     17.8
3  0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3  222     18.7
4  0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3  222     18.7

   LSTAT  MEDV
0   4.98  24.0
1   9.14  21.6
2   4.03  34.7
3   2.94  33.4
4   5.33  36.2
```

Step 3: Check for Missing Values

df.isna().sum()



	0
CRIM	0
ZN	0
INDUS	0
CHAS	0
NOX	0
RM	0
AGE	0
DIS	0
RAD	0
TAX	0
PTRATIO	0
LSTAT	0
MEDV	0

dtype: int64

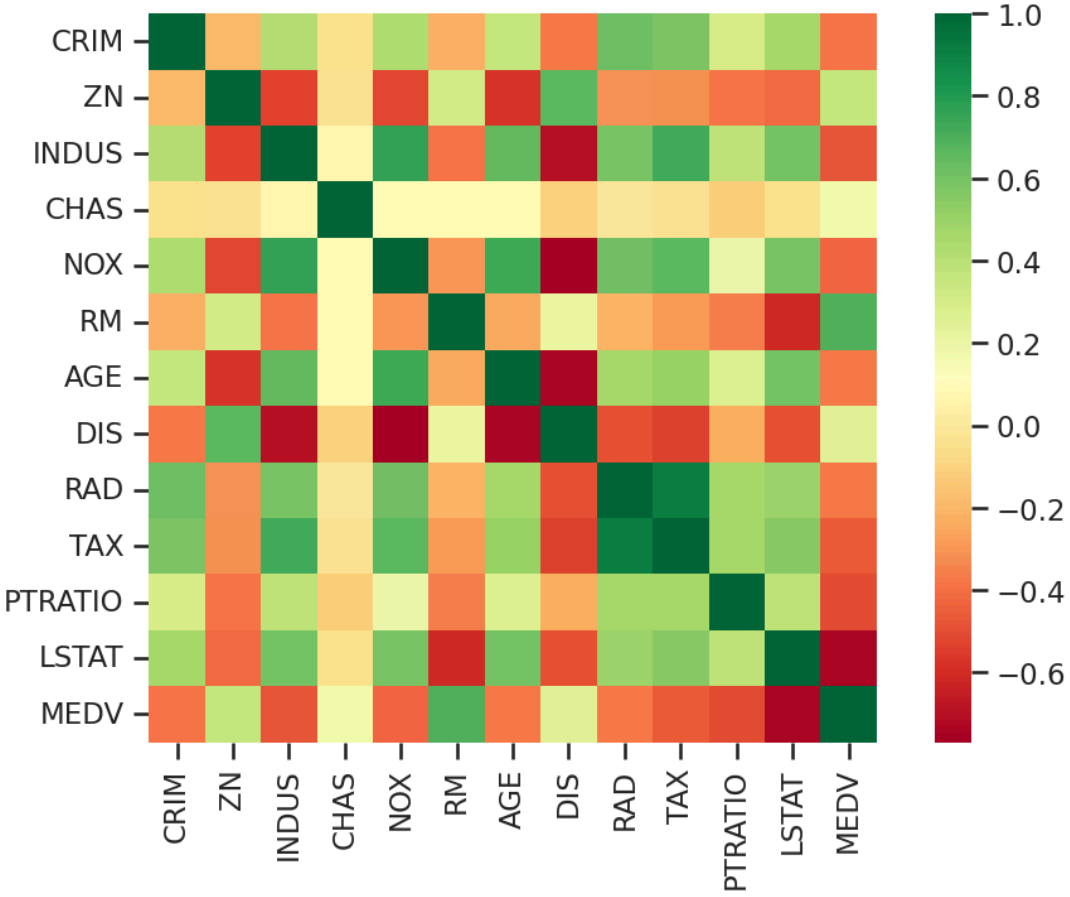
In case of regression, it is important to make sure that our data does not have any missing values because regression won't work if the data has missing values.

The column **MEDV** is used as the target variable while building the model. The target variable(y) is separate from the feature variable(x).

Step 4: Perform Correlation Analysis

Correlation analysis is an important part of building any model. We have to understand the distribution of the data and how the independent variables correlate with the dependent variable.

```
# Plotting a correlation heatmap
sns.heatmap(df.corr(), square=True, cmap='RdYlGn')
```

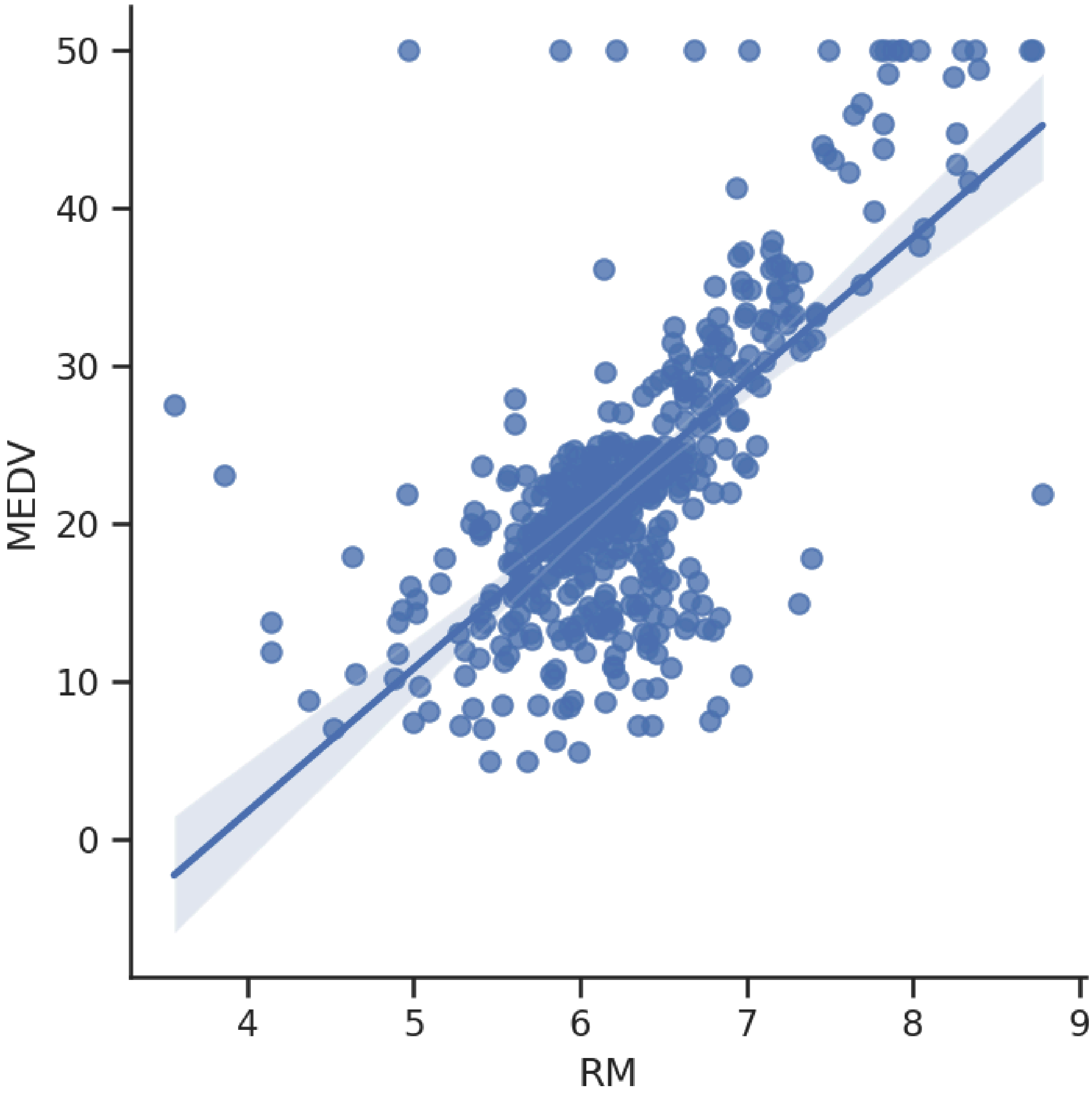


MEDV represents the **median value of owner-occupied homes** and serves as the target variable we want to predict.

RM is the **average number of rooms per dwelling.** The correlation heatmap shows a strong positive correlation between RM and MEDV, indicating that homes with more rooms generally have higher median prices.
Because of this strong relationship, we select RM as the input feature (X) and MEDV as the output target (y) for our regression model.

Step 5: Visualize the Relationship

```
# Plotting the linear relationship
sns.lmplot(x='RM', y='MEDV', data=df)
```



The above screenshot shows a strong correlation between these two variables. However, there are some outliers that can easily be spotted from the graph.

Step 6: Prepare Data for Training

```
# Preparing the features and target
X = df[['RM']]
y = df[['MEDV']]
```

The double square brackets ([['RM']]) ensure that the result is a DataFrame (not just a Series), which is compatible with scikit-learn models.

Step 7: Split the Dataset

```
from sklearn.model_selection import train_test_split

# Splitting data into training and testing sets (70% train, 30% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=10)
```

We split the dataset into training and testing subsets:

- 70% for training (`X_train`, `y_train`) – this portion is used to train the model.
- 30% for testing (`X_test`, `y_test`) – this is used to evaluate how well the model performs on unseen data.

The parameter `random_state=10` ensures that the split is reproducible — you'll get the same split every time you run it.

Step 8: Train the Linear Regression Model

```
from sklearn.linear_model import LinearRegression

# Creating and training the model
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

We import the LinearRegression model from scikit-learn.

regressor = LinearRegression() creates an instance of the linear regression model.

.fit(X_train, y_train) trains the model by finding the best-fitting line for the training data. This involves calculating the slope and intercept that minimize the difference between actual and predicted values (minimizing the mean squared error).

After this step, the model is ready to make predictions on new data.

Model Evaluation

After training a model, we need to check how well it performs using test data (data the model hasn't seen before). This helps us know if the model can generalize to new data. We use R^2 - statistics, which is a common method of measuring the accuracy of regression models:

Step 1: Evaluate Using R^2 Score

R^2 Score (Coefficient of Determination) measures how well the model explains the variation in the target variable.
Range: 0 to 1
0 → No predictive power
1 → Perfect predictions

```
#check prediction score/accuracy
regressor.score(X_test, y_test)
```

- regressor is the trained Linear Regression model.
- X_test contains the input features from the test set.
- y_pred contains the predicted home prices (MEDV values).

Output: 0.5383003344910231

- **$R^2 \approx 0.53$** → Model explains **53% of the variance** in home prices using RM.
- Can be improved by using **more independent variables**.

Step 2:

```
# predict the y values
y_pred=regressor.predict(X_test)
# a data frame with actual and predicted values of y
evaluate = pd.DataFrame({'Actual': y_test.values.flatten(),
'Predicted': y_pred.flatten()})
```

- y_test contains actual MEDV values.

- flatten() is used to convert from column shape to 1D array (for clean tabular display).
- evaluate is a DataFrame with two columns:
    - 'Actual' → True values from the dataset
    - 'Predicted' → Model's estimated values

Step 3: Display the first 10 comparisons

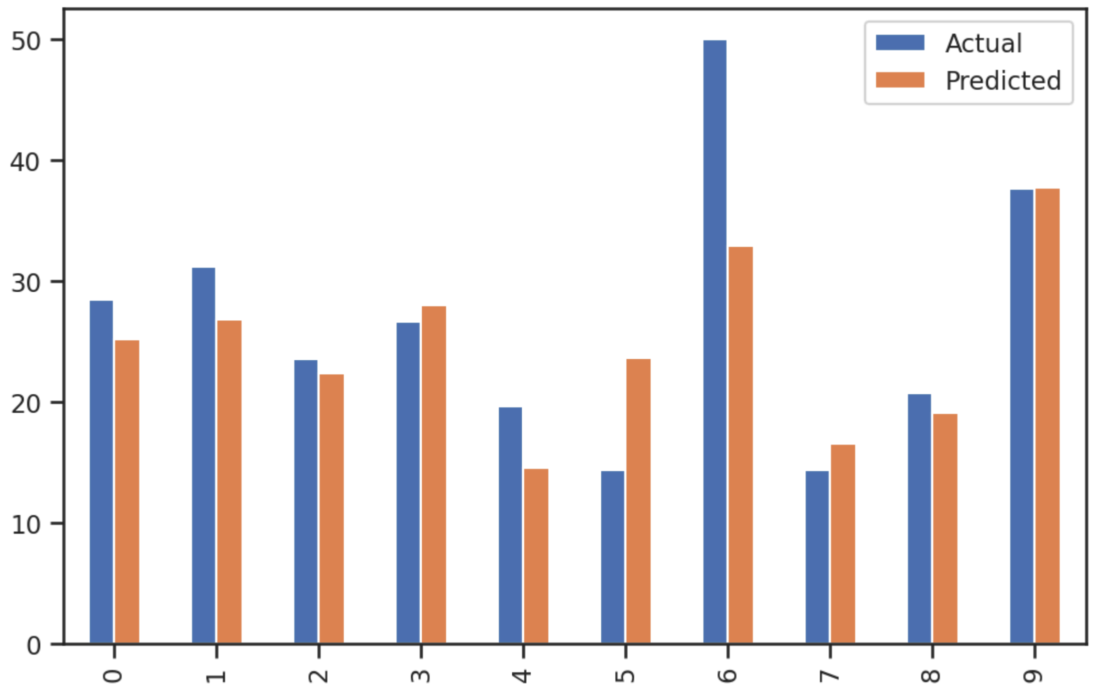
```
evaluate.head(10)
```

Output:

```
        Actual      Predicted
0       28.4        25.153909
1       31.1        26.773693
2       23.5        22.284072
3       26.6        27.997335
4       19.6        14.484456
5       14.3        23.569336
6       50.0        32.839084
7       14.3        16.535597
8       20.7        19.026896
9       37.6        37.689635
```

Step 4: Visualize with a Bar Plot

```
evaluate.head(10).plot(kind = 'bar')
```



Note: Predicted values are lower than the actual values.

Computing Accuracy

Sklearn provides metrics that help us evaluate our models with multiple formulas. Three main metrics used to evaluate models are

- Mean absolute error
- Mean squared error
- R² score

Let's try these methods:

```
# Scoring the model
from sklearn.metrics import r2_score, mean_squared_error,mean_absolute_error

# R2 Score
print(f"R2 score: {r2_score(y_test, y_pred)}")

# Mean Absolute Error (MAE)
print(f"MAE score: {mean_absolute_error(y_test, y_pred)}")

# Mean Squared Error (MSE)
print(f"MSE score: {mean_squared_error(y_test, y_pred)}")
```

Output:

R2 score: 0.5383003344910231

MAE score: 4.750294229575126

MSE score: 45.07733942471831

Note - Model Evaluation Metrics in Regression:

- R² Score: Measures goodness of fit. Closer to 1 = better.
- MAE: Average absolute error. Lower = better.
- MSE: Squared average error. Lower = better.

Understanding Accuracy

After training a regression model and making predictions, we need to assess how accurate those predictions are. Accuracy in regression refers to how close the predicted values are to the actual observed values. Since regression deals with continuous outcomes, accuracy is measured by how small the prediction errors are, using various error metrics.

Example:

Let's say:
Actual Value (xi) = 28.4
Predicted Value (ŷi) = 25.15
Error (ei) = xi - ŷi = 28.4 - 25.15 = 3.25

The error is given by the following formula:

$$\epsilon_i = x_i - \hat{x_i}$$

If we just average the errors, positive and negative values cancel each other.

Solution: We square the errors!

$$\text{Squared Error (SE)} = \sum_{i}^{n} \epsilon_i^2$$

Once we know how to compute the squared error, we can compute the mean squared error.

$$\text{Mean Squared Error (MSE)} = \frac{1}{n} \times SE$$

Now, if we take the root of the mean squared error, we get another accuracy measure called the root mean squared error (RMSE). The equation now becomes this:

$$\text{Root Mean Squared Error (RMSE)} = \sqrt{MSE}$$

Another type of accuracy measure that is widely used is called the relative mean squared error (rMSE)

$$rMSE = \frac{n-1}{n} \frac{\sum_i^n \epsilon_i^2}{\sum_i^n (x_i - E(x)^2)} = \frac{MSE}{Var(x)}$$

In addition to rMSE, we have used the R2 method. The formula for computing R2 is as follows:

$$R^2 = 1 - rMSE$$

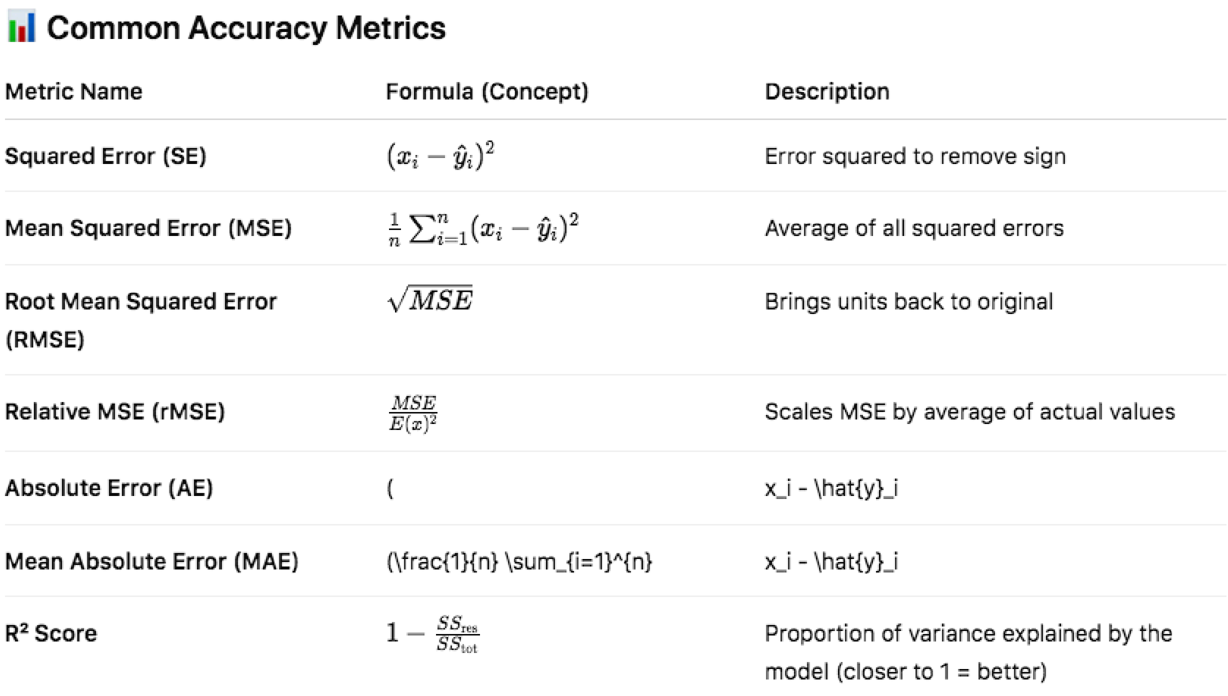
One more type of accuracy measure that is often seen in data science is the absolute error. As the name suggests, it takes the absolute value and computes the sum.

$$\text{Absolute error (AE)} = \sum_{i}^{n} \sqrt{\epsilon_i^2} = \sum_{i}^{n} |\epsilon_i|$$

Finally, one more type of error that can be used in addition to absolute error is the mean absolute error. The formula for computing mean absolute error is as follows:

$$\text{Mean Absolute Error(MAE)} = \frac{1}{n} \times AE$$

Summary



📊 Common Accuracy Metrics

Metric Name	Formula (Concept)	Description
Squared Error (SE)	$(x_i - \hat{y}_i)^2$	Error squared to remove sign
Mean Squared Error (MSE)	$\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{y}_i)^2$	Average of all squared errors
Root Mean Squared Error (RMSE)	$\sqrt{MSE}$	Brings units back to original
Relative MSE (rMSE)	$\frac{MSE}{E(x)^2}$	Scales MSE by average of actual values
Absolute Error (AE)	(	x_i - \hat{y}_i
Mean Absolute Error (MAE)	(\frac{1}{n} \sum_{i=1}^{n}	x_i - \hat{y}_i
R² Score	$1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$	Proportion of variance explained by the model (closer to 1 = better)

Tip to Remember

The name of the metric tells you what it does:

- **Mean = average**
- **Absolute = ignore minus sign**
- **Squared = magnify large errors**
- **Root = reverse squaring**
- **Relative = scaled for context**

Implementing a Multiple Linear Regression Model

When a dependent variable relies on several independent variables, the relationship can be captured using multiple linear regression.

In simple linear regression, we use one independent variable (X) to predict the dependent variable (y) whereas in multiple linear regression, we use multiple independent variables to predict y.

Predicting house price (MEDV) based on:

- % lower status of population (LSTAT)
- Crime rate (CRIM)
- Nitric oxide concentration (NOX)
- Tax rate (TAX)
- Student-teacher ratio (PTRATIO)
- Charles River dummy variable (CHAS)
- Distance to employment centers (DIS)

# Visualizing the Concept
Simple Linear Regression
y = b0 + b1·x
Geometry: A line in 2D space.

Multiple Linear Regression with 2 features
y = b0 + b1·x1 + b2·x2
Geometry: A plane in 3D space.
More than 2 features (like 7 variables in our case)
y = b0 + b1·x1 + b2·x2 + ... + b7·x7
Geometry: A hyperplane in n-dimensional space.

```
# Preparing the data
X = df[['LSTAT','CRIM','NOX','TAX','PTRATIO','CHAS','DIS']]
y = df[['MEDV']]
```

```
#Splitting the dataset into train and test sets
# Splitting the dataset into train and test sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
= 0.3, random_state = 10)
```

```
#Fitting the training data to our model
regressor.fit(X_train, y_train)
#Score of this model
regressor.score(X_test, y_test)
```

Output: 0.644694253426537

Lets predict the y values with our model and evaluate it:

```
# predict the y values
y_pred=regressor.predict(X_test)
# a data frame with actual and predicted values of y
evaluate = pd.DataFrame({'Actual': y_test.values.flatten(),
'Predicted': y_pred.flatten()})
evaluate.head(10)
```

Output:

```
        Actual      Predicted
0       28.4        27.445779
1       31.1        31.364849
2       23.5        30.681874
3       26.6        22.143726
4       19.6        23.063037
5       14.3        16.421246
```

```
6       50.0          36.733894
7       14.3          15.887917
8       20.7          25.718492
9       37.6          32.816198
```

Let’s make another multiple linear regression model with fewer features:

```
# Preparing the data
X = df[['LSTAT','CRIM','NOX','TAX','PTRATIO']]
y = df[['MEDV']]
# Splitting the dataset into train and test sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
= 0.3, random_state = 10)
# Fitting the training data to our model
regressor.fit(X_train, y_train)
#score of this model
regressor.score(X_test, y_test)
```

Output: 0.5798770784084717

```
# predict the y values
y_pred=regressor.predict(X_test)
# a data frame with actual and predicted values of y
evaluate = pd.DataFrame({'Actual': y_test.values.flatten(),
'Predicted': y_pred.flatten()})
evaluate.head(10)
```

Output:

```
        Actual        Predicted
0       28.4          25.512908
1       31.1          31.496427
2       23.5          31.260496
3       26.6          26.553401
4       19.6          25.826407
5       14.3          17.589252
6       50.0          34.913399
7       14.3          15.165121
8       20.7          21.605243
9       37.6          31.078599
```

Questions

1. **Explain the concept of hypothesis testing along with its key principles. Also, describe the following parameters related to hypothesis testing:**
   - **Null hypothesis**
   - **Alternative hypothesis**
   - **Type I and Type II errors**
   - **P-values**
   - **Level of significance**
2. **In a survey of 4,500 parents of teens aged 18+, 65% said social media causes stress. Earlier, it was believed that only 48% felt this way.**

   Answer the following based on the case using the `statsmodels` library:
   a. **Define the null and alternative hypotheses.**
   b. **Perform a one-sample proportion Z-test.**
   c. **Interpret the z-statistic and p-value in the context of the problem.**
   d. **Conclude whether the new data provides significant evidence that more than 48% of parents believe social media causes stress.**
3. **In a reading competition for adults, the reading speeds (words per minute) of 89 randomly selected participants were recorded. The historical average reading speed is assumed to be 212 words per minute. Sample speeds include:**

   236, 239, 209, 246, 246, 245, 215, 212, 242, 241, ..., 237, 223.

   Assume the population standard deviation is known. Based on this data, answer the following:

   a. **Formulate the null and alternative hypotheses to test if the average reading speed has increased from 212 words per minute.**
   b. **Identify the appropriate type of hypothesis test and justify your choice.**
   c. **Briefly outline the steps to conduct this hypothesis test.**
   d. **Given the test results: Z-statistic = 91.64, P-value = 0.0, state your conclusion at a 5% significance level.**
4. **List different types of hypothesis tests used in statistics. For each, mention one context where it is typically applied.**

   A teacher wants to check whether the average height of students in a particular class is more than 175 cm. A random sample of 55 students' heights was collected:

Sample data:
[172, 184, 174, 168, 174, 183, 173, 173, 184, 179, 171, 173, 181, 183, 172, 178, 170, 182, 181, 172, 175, 170, 168, 178, 170, 181, 180, 173, 183, 180,

177, 181, 171, 173, 171, 182, 180, 170, 172, 175, 178, 174, 184, 177, 181, 180, 178, 179, 175, 170, 182, 176, 183, 179, 177]

The population standard deviation is unknown, and the sample size is less than 60. Answer the following questions based on this scenario:

a. **Define the population and the parameter of interest.**
b. **State the null and alternative hypotheses for the test.**
c. **Justify which hypothesis test is appropriate here and why.**
d. **Suppose the calculated average height is 176.55 cm and the P-value is 0.0194, interpret this result at a 5% significance level.**
e. **State your conclusion clearly in the context of the problem.**

5. **Name any four types of hypothesis tests with one example use case each. What is a T-test? Explain with an example, including hypotheses and result interpretation.**
6. **Explain the concept of p-hacking.**
7. **Explain the concept of regression and its types. Illustrate your answer with suitable examples.**
8. **Describe the process of developing and evaluating a simple linear regression model using the Boston housing dataset.**
9. **Illustrate how to evaluate a simple linear regression model using the Boston housing dataset. Explain the R^2 score, comparison of actual vs predicted values, and other important evaluation metrics.**
10. **Explain the concept of accuracy in regression models and describe various error metrics used to measure it.**
11. **Explain how to implement multiple linear regression with the Boston housing dataset. How does it differ from simple linear regression?**