Linear regression is a statistical modeling technique used to establish a linear relationship between a dependent variable and one or more independent variables. The primary goal of linear regression is to find the best-fitting straight line through the data points, which can be used to predict the value of the dependent variable based on the values of the independent variables.

The general form of a linear regression model is:

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon$

Where:
- y is the dependent variable
- $x_1, x_2, \ldots, x_p$ are the independent variables
- β_0 is the y-intercept (the value of y when all independent variables are zero)
- $\beta_1, \beta_2, \ldots, \beta_p$ are the regression coefficients (slopes) for each independent variable
- ε is the error term (the difference between the predicted and actual values of y)

The regression coefficients ($\beta_1, \beta_2, \ldots, \beta_p$) represent the change in the dependent variable for a one-unit change in the corresponding independent variable, holding all other variables constant. These coefficients are estimated using the least squares method, which minimizes the sum of the squared differences between the observed and predicted values of the dependent variable.

Assumptions of linear regression:
1. Linearity: The relationship between the dependent and independent variables is linear.
2. Independence: The observations are independent of each other.
3. Homoscedasticity: The variance of the errors is constant across all levels of the independent variables.
4. Normality: The errors are normally distributed with a mean of zero.
5. No multicollinearity: The independent variables are not highly correlated with each other.

Assessing the model:
1. R-squared (coefficient of determination): Measures the proportion of variance in the dependent variable that is predictable from the independent variable(s). It ranges from 0 to 1, with higher values indicating a better fit.
2. Adjusted R-squared: A modified version of R-squared that accounts for the number of predictors in the model, penalizing the addition of unnecessary variables.
3. F-statistic: Tests the overall significance of the regression model, determining whether the independent variables collectively have a significant effect on the dependent variable.
4. t-statistic: Tests the significance of individual regression coefficients, determining whether each independent variable has a significant effect on the dependent variable.

5. Residual analysis: Examining the differences between the observed and predicted values to assess the model's assumptions and identify outliers or influential observations.

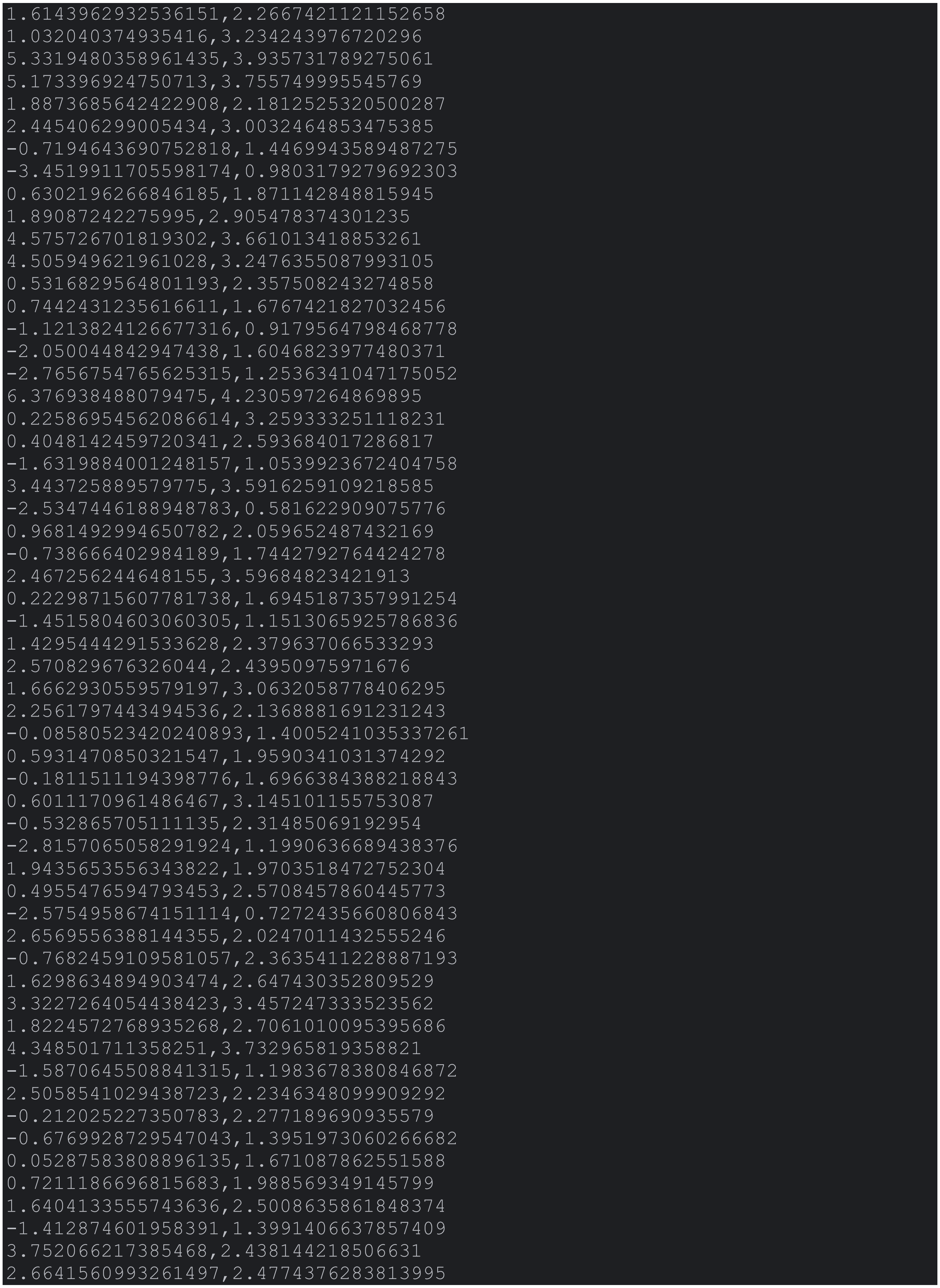
Applications of linear regression:
1. Predictive modeling: Using the model to predict the value of the dependent variable based on new values of the independent variables.
2. Explanatory modeling: Understanding the relationships between variables and identifying which factors have the most significant impact on the outcome.
3. Trend analysis: Examining how the dependent variable changes over time or in relation to other variables.

Linear regression is widely used in various fields, such as economics, finance, social sciences, and engineering, to understand and predict relationships between variables. However, it is essential to recognize its limitations, especially when dealing with non-linear relationships or when the model's assumptions are violated. In such cases, more advanced techniques like polynomial regression, generalized linear models, or machine learning algorithms may be more appropriate.

Practical Example

AdvertisingSalesData.csv :

```
TV_Ad_Budget,Sales
5.91013086491916,4.714614608003876
2.5003930209180583,2.076238375704194
3.946844960264348,2.5488109888364376
7.102232998003645,4.615368253480099
6.1688949753749185,3.2641067900553953
-0.9431946996910274,2.688852182917338
3.8752210438139736,2.9557568227643185
1.1216069792557553,1.9627546880563478
1.2419528705161054,3.3340568743950243
2.526496254845931,3.4982062721709912
1.860108927902195,3.4918121585839437
5.135683767407437,3.9937274593599237
3.4025943128674836,2.5901654513328936
1.804187541232071,3.496288738919138
2.6096580818635644,2.6488957390833794
2.334185818435667,3.1014839434288977
5.235197682894015,4.044185288755079
0.9871043405854978,2.218626255630232
2.2826692541272533,2.991840461411216
-0.6352393482543119,2.2705315313069696
-4.882474539585196,0.7234704037022559
3.1340464886009016,2.390513551288173
3.6610904971487646,3.2474462362476575
-0.3554125510161048,2.556569183038684
7.174386559969019,3.8050320381250233
-2.135914186496912,1.2844084738870911
```



```
1.6143962932536151,2.2667421121152658
1.032040374935416,3.234243976720296
5.3319480358961435,3.935731789275061
5.173396924750713,3.755749995545769
1.8873685642422908,2.1812525320500287
2.445406299005434,3.0032464853475385
-0.7194643690752818,1.4469943589487275
-3.4519911705598174,0.9803179279692303
0.6302196266846185,1.871142848815945
1.89087242275995,2.905478374301235
4.575726701819302,3.661013418853261
4.505949621961028,3.2476355087993105
0.5316829564801193,2.357508243274858
0.7442431235616611,1.6767421827032456
-1.1213824126677316,0.9179564798468778
-2.050044842947438,1.6046823977480371
-2.7656754765625315,1.2536341047175052
6.376938488079475,4.230597264869895
0.22586954562086614,3.259333251118231
0.4048142459720341,2.593684017286817
-1.6319884001248157,1.0539923672404758
3.443725889579775,3.5916259109218585
-2.5347446188948783,0.581622909075776
0.9681492994650782,2.059652487432169
-0.738666402984189,1.7442792764424278
2.467256244648155,3.59684823421913
0.22298715607781738,1.6945187357991254
-1.4515804603060305,1.1513065925786836
1.4295444291533628,2.379637066533293
2.570829676326044,2.43950975971676
1.6662930559579197,3.0632058778406295
2.2561797443494536,2.1368881691231243
-0.08580523420240893,1.4005241035337261
0.5931470850321547,1.9590341031374292
-0.1811511194398776,1.6966384388218843
0.6011170961486467,3.145101155753087
-0.532865705111135,2.31485069192954
-2.8157065058291924,1.1990636689438376
1.9435653556343822,1.9703518472752304
0.4955476594793453,2.5708457860445773
-2.5754958674151114,0.7272435660806843
2.6569556388144355,2.0247011432555246
-0.7682459109581057,2.3635411228887193
1.6298634894903474,2.647430352809529
3.3227264054438423,3.457247333523562
1.8224572768935268,2.7061010095395686
4.348501711358251,3.732965819358821
-1.5870645508841315,1.1983678380846872
2.5058541029438723,2.2346348099909292
-0.212025227350783,2.277189690935579
-0.6769928729547043,1.3951973060266682
0.05287583808896135,1.671087862551588
0.7211186696815683,1.988569349145799
1.6404133555743636,2.5008635861848374
-1.412874601958391,1.3991406637857409
3.752066217385468,2.438144218506631
2.6641560993261497,2.4774376283813995
```

```
-2.3406092156930596,0.18611565917986872
5.220630484488999,3.8788048708602934
6.239722940076458,3.070888054219563
4.446948927899127,2.781893008655513
1.0501879104691227,2.341138912771224
-1.1768815537763562,1.2771540356714366
4.136129317327842,4.012346092901721
0.49205763256705093,1.501188834908391
4.556112675956069,3.5003592374614123
2.020687445192151,2.5865648244438977
3.9415975912092818,2.598432528492186
2.3909159929360047,2.978913128146678
3.26643292047987,2.894156710532837
1.5262500518020512,2.8437702911474494
5.964676234764588,4.201154947411243
1.8172802317590497,3.62680204416806
2.504973408611754,3.419755997301722
```

Source code:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# Load data from CSV file
data = pd.read_csv('AdvertisingSalesData.csv')

# Splitting data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data['TV_Ad_Budget'],
data['Sales'], test_size=0.2, random_state=1)

# Reshape the data to fit the model
X_train = X_train.values.reshape(-1,1)
X_test = X_test.values.reshape(-1,1)

# Creating a linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Predicting sales from the test set
y_pred = model.predict(X_test)

# Plotting
plt.scatter(X_test, y_test, color='black', label='Actual sales')
plt.plot(X_test, y_pred, color='blue', linewidth=3, label='Predicted sales')
plt.title('TV Advertising Budget vs Sales')
plt.xlabel('TV Advertising Budget (thousands of dollars)')
plt.ylabel('Sales (thousands of units)')
plt.legend()
plt.show()

# Displaying the model coefficients
print("Coefficient:", model.coef_[0])
print("Intercept:", model.intercept_)
```