LAB 1: Simple Linear Regression

There are five basic steps when you're implementing linear regression:

1. Import the packages and classes you need.
2. Provide data to work with and eventually do appropriate transformations.
3. Create a regression model and fit it with existing data.
4. Check the results of model fitting to know whether the model is satisfactory.
5. Apply the model for predictions.

```
# Import packages and classes

   import numpy as np
   from sklearn.linear_model import LinearRegression


# Provide Data

x = np.array([5, 15, 25, 35, 45, 55]).reshape((-1, 1))
y = np.array([5, 20, 14, 32, 22, 38])

print(x)
print(y)

# Create a model and fit it

model = LinearRegression()
model.fit(x,y)


# Get result

r_sq = model.score(x, y)
print('coefficient of determination:', r_sq)

print('intercept:', model.intercept_)

print('slope:', model.coef_)


# Predict responses

y_pred = model.predict(x)
print('predicted response:', y_pred, sep='\n')
```

LAB 2: Simple Linear Regression

```
# Import packages and classes
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
%matplotlib inline


# Read the IceCreamData.csv file
IceCream=pd.read_csv('IceCreamData.csv')
print(IceCream)

# Print first 5 data
IceCream.head()

# Print last 5 data
IceCream.tail()

# Print mathematical description
IceCream.describe()

# Print information of Dataset
IceCream.info()

# Divide the data into “Attributes” and “labels”
X = IceCream[['Temperature']]
y = IceCream['Revenue']

# Split 80% of the data to the training set while 20% of the data to test set

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)

# Create a Linear Regression model and fit it
regressor =LinearRegression(fit_intercept=True)
regressor.fit(X_train,y_train)

# Getting Results
print('Linear Model Coeff (m) =' , regressor.coef_)
print('Linear Model Coeff (b) =' , regressor.intercept_)

# Predicting the data
y_predict=regressor.predict(X_test)
print(y_predict)
```

```
# Scatter plot on Training Data
plt.scatter(X_train,y_train,color='blue')
plt.plot(X_train,regressor.predict(X_train),color='red')
plt.ylabel('Revenue [$]')
plt.xlabel('Temperatur [degC]')
plt.title('Revenue Generated vs. Temperature @Ice Cream Stand (Training)')

# Scatter plot on Testing Data
plt.scatter(X_test,y_test,color='blue')
plt.plot(X_test,regressor.predict(X_test),color='red')
plt.ylabel('Revenue [$]')
plt.xlabel('Temperatur [degC]')
plt.title('Revenue Generated vs. Temperature @Ice Cream Stand (Training)')


# Prediction the revenve using Temperature Value directly
print('---------0---------')
Temp = -0
Revenue = regressor.predict([[Temp]])
print(Revenue)
print('--------35----------')
Temp = 35
Revenue = regressor.predict([[Temp]])
print(Revenue)
print('--------55----------')
Temp = 55
Revenue = regressor.predict([[Temp]])
print(Revenue)
```