Statistical Analysis

ELEC2103 Assignment Semester 2 2023

Contents

Students Details

```
%NAME1 = "Alicia Pan";
%SID1 = "520452677";
%Contribution_Percentage_1= equal;

%NAME2 = "Yancheng Lin";
%SID2 = "510022956";
%Contribution_Percentage_2= equal;

%NAME3 = "Zihan Ding";
%SID3 = "530658748";
%Contribution_Percentage_3= equal;

%DATE = "13-10-2023";
```

Part 1

The purpose of this report is to investigate the relationship between 3 years CPI (Consumer price index) and unemployment of 45 Walmart stores in different regions. The report analyzes the data from different years, the goodness-of-fit of the model and regression line. We found out that there is only a slight relationship between CPI and unemployment, as is proved by the linear regression. In Sections 3, we undertook a further exploration of predictive modeling for the relationship between Consumer Price Index (CPI) and unemployment rates. Section 3a employed K-fold cross-validation to compare linear and polynomial regression models using all data in 2010, revealing the superiority of the fifth-order polynomial in accuracy. Finally, we delved into 2010-2012 data, shifting from linear models to neural networks in section 3b. The Mean Squared Error analysis favored the neural network, particularly for higher CPI values. These findings collectively underscore the nuanced nature of CPI and unemployment predictions, guiding the adoption of more sophisticated models for enhanced accuracy.

The data is sourced from Kaggle data site (www.kaggle.com), authored by Rutu Patel, posted two years ago. All the following analyses are based on the data given without any change. The data is valid as it is properly formatted, with a complete data set and a large volume. Although it is valid, we could not prove it is reliable. Since Kaggle is a net site where anyone can freely upload data, the accuracy of the data cannot be completely guaranteed. Besides, there is missing data on CPI, which can cause some trouble to the accuracy of the data. The data is much more like a simple datasheet provided to test the author%s students as there are requirements and some comments in the data card. Due to the uncertainty of the original use of the data, we cannot reach an agreement that the data is of high efficiency.

```
data = readtable('WALMART_SALES_DATA.csv'); % Read the data
whos CPI Date Fuel_Price Holiday_Flag Store Temperature Unemployment Weekly_Sales
```

```
  Name                Size             Bytes  Class     Attributes

  CPI              6435x1              51480  double
  Date             6435x1             797940  cell
  Fuel_Price       6435x1              51480  double
  Holiday_Flag     6435x1              51480  double
  Store            6435x1              51480  double
  Temperature      6435x1              51480  double
```

```
Unemployment      6435x1              51480  double
Weekly_Sales      6435x1              51480  double
```

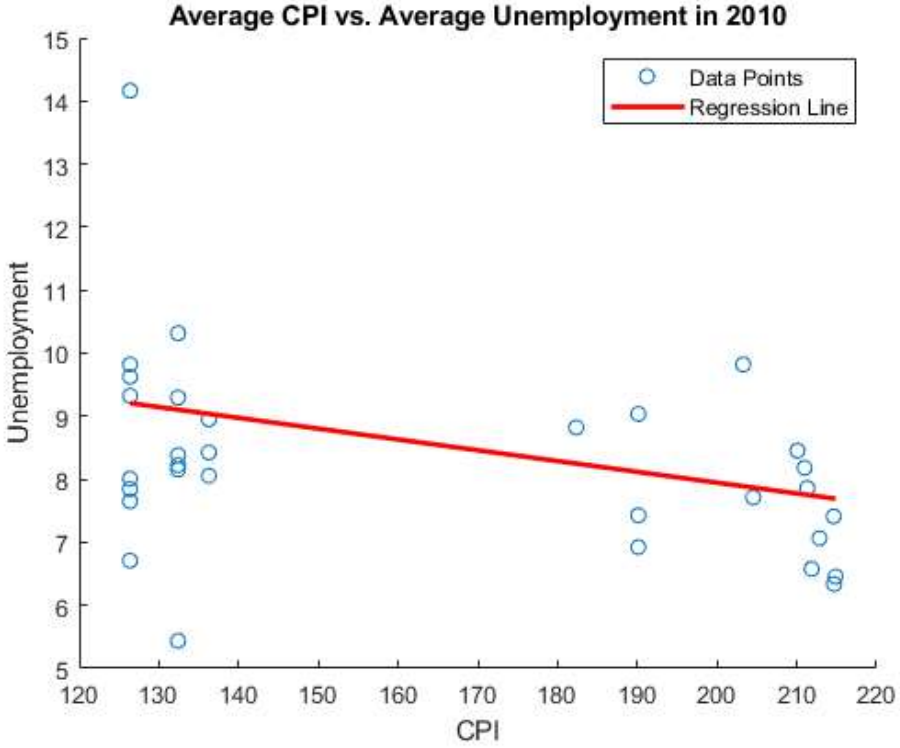
The dataset contains 6,435 rows, with each row representing weekly data for a particular store. This dataset is organized into eight columns: Store, Date, Weekly_Sales, Holiday_Flag, Temperature, Fuel_Price, CPI, and Unemployment. Store: indicates the specific store number. There are a total of 45 stores, numbered from 1 to 45. Date: represents the date when the data was collected. Data points are spaced one week apart, spanning from 05-02-2010 to 26-10-2012 . CPI: knowns as Consumer Price Index, CPI is a statistic that measures the average change in consumer prices for goods and services over time. Variations in the CPI can aid in the monitoring of inflation over time and the comparison of inflation rates between countries. Unemployment: refers to the unemployment rate on a specific date for each store. In our report, we opted not to utilize the columns Holiday_Flag, Temperature, and Fuel_Price. Our primary focus was on understanding the relationship between CPI and unemployment.

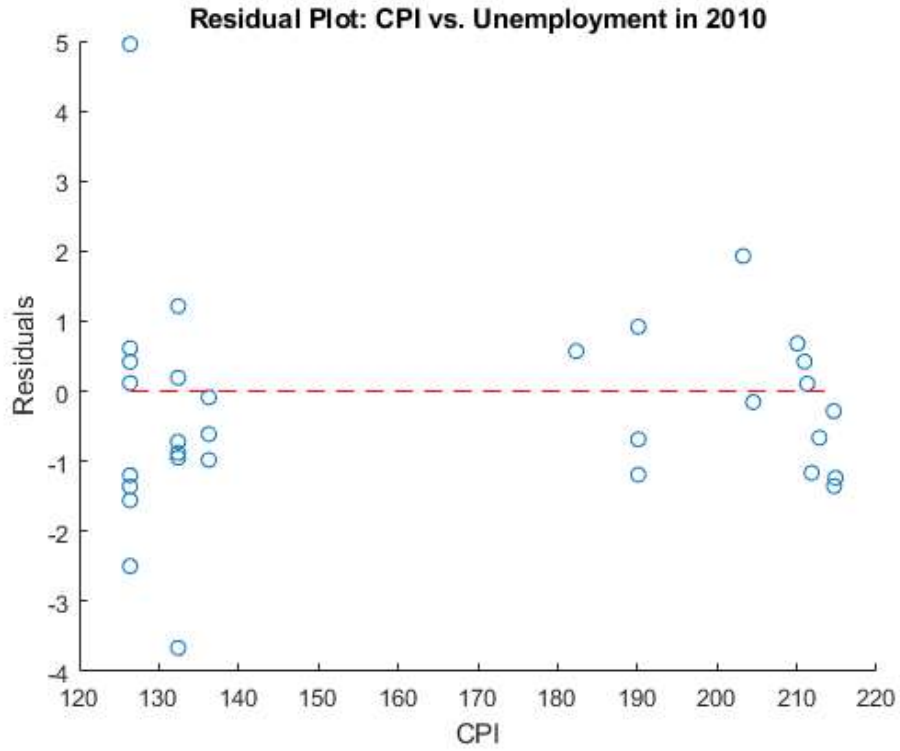
Part 2

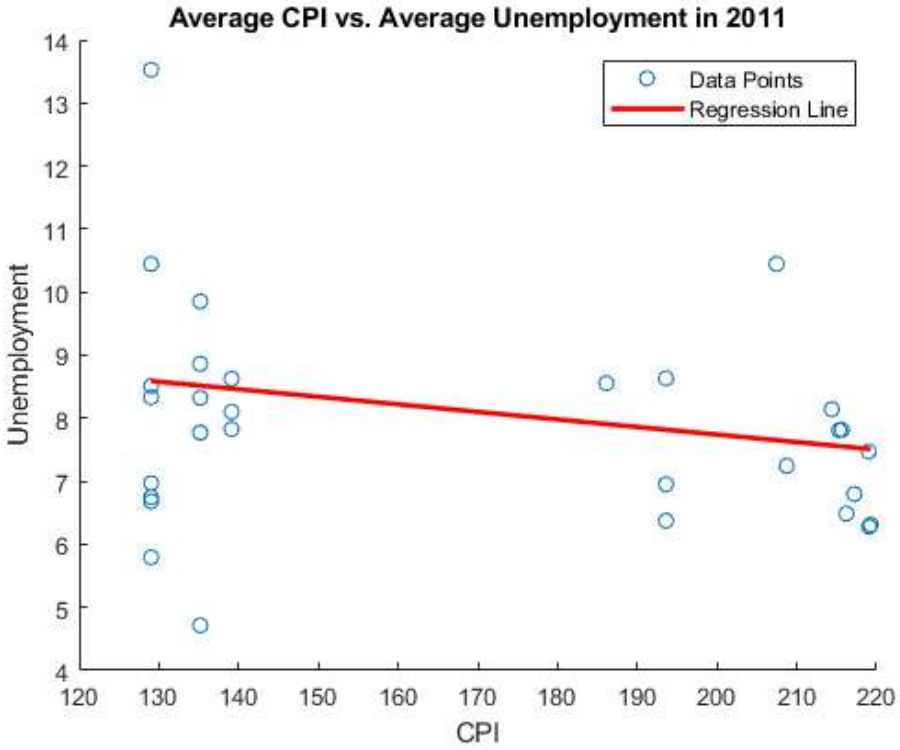
To begin, we classified the data by years to pre-process them, which can have a better view of the model. Then we used scatter plots with regression line and correlation coefficients from CPI to unemployment divided by years to determine whether there is a linear relationship between them.

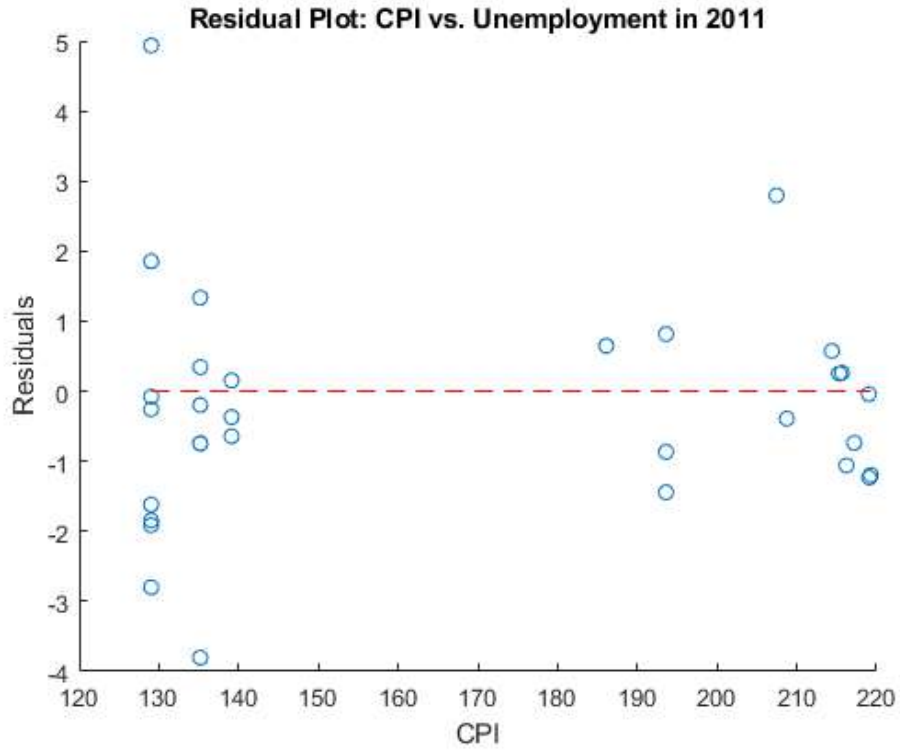
```
data = readtable('WALMART_SALES_DATA.csv'); % Read the data

% Analyze data for the year 2010 with 'dd-MM-yyyy' date format
results2010 = analyzeYear(2010, 'dd-MM-yyyy');
% Extract relevant data
avgCPI = results2010.avgCPI;
avgUnemployment = results2010.avgUnemployment;
linearModel = results2010.linearModel;
% Create a scatter plot
figure(1);
scatter(avgCPI, avgUnemployment);
xlabel('CPI');
ylabel('Unemployment');
title('Average CPI vs. Average Unemployment in 2010');
% Display regression results
coefficients = polyfit(avgCPI, avgUnemployment, 1);
fit_line = polyval(coefficients, avgCPI);
hold on;
plot(avgCPI, fit_line, 'r-', 'LineWidth', 2);
legend('Data Points', 'Regression Line');

% Create a residual plot
figure(2);
residuals = avgUnemployment - predict(linearModel, avgCPI);
scatter(avgCPI, residuals);
xlabel('CPI');
ylabel('Residuals');
title('Residual Plot: CPI vs. Unemployment in 2010');
hold on;
plot([min(avgCPI), max(avgCPI)], [0, 0], 'r--');
hold off;

% Data Analysis for the year 2011
results2011 = analyzeYear(2011, 'dd-MM-yyyy');
% Extract relevant data
avgCPI = results2011.avgCPI;
avgUnemployment = results2011.avgUnemployment;
linearModel = results2011.linearModel;
% Create a scatter plot
figure(3);
scatter(avgCPI, avgUnemployment);
xlabel('CPI');
ylabel('Unemployment');
title('Average CPI vs. Average Unemployment in 2011');
% Display regression results
coefficients = polyfit(avgCPI, avgUnemployment, 1);
fit_line = polyval(coefficients, avgCPI);
```

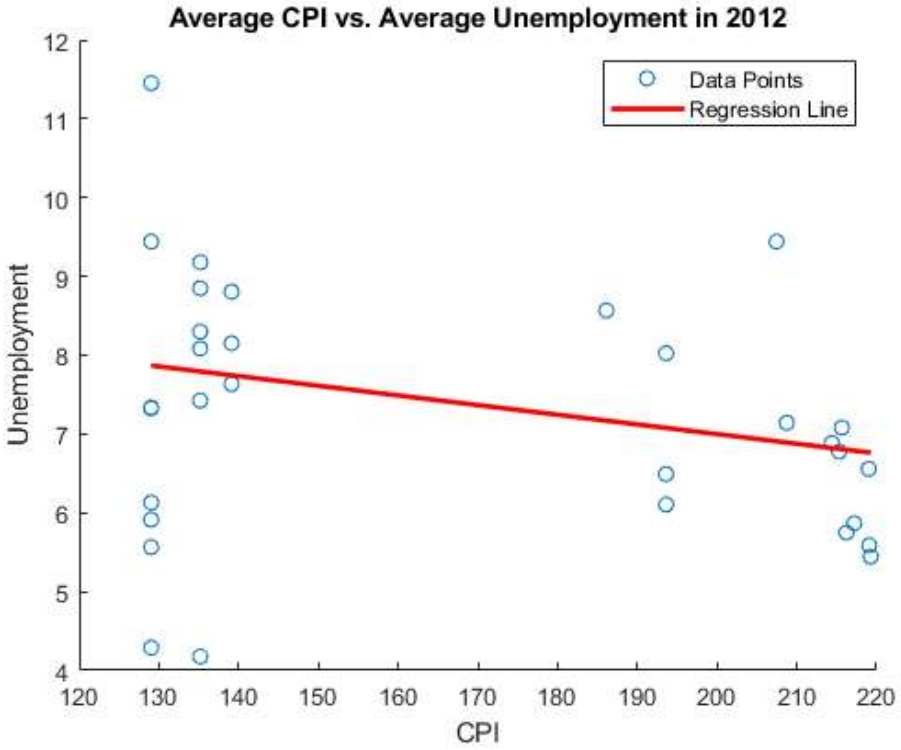
```
hold on;
plot(avgCPI, fit_line, 'r-', 'LineWidth', 2);
legend('Data Points', 'Regression Line');
% Create a residual plot
figure(4);
residuals = avgUnemployment - predict(linearModel, avgCPI);
scatter(avgCPI, residuals);
xlabel('CPI');
ylabel('Residuals');
title('Residual Plot: CPI vs. Unemployment in 2011');
hold on;
plot([min(avgCPI), max(avgCPI)], [0, 0], 'r--');
hold off;

% Data Analysis for the year 2012
results2012 = analyzeYear(2012, 'dd-MM-yyyy');
% Extract relevant data
avgCPI = results2011.avgCPI;
avgUnemployment = results2012.avgUnemployment;
linearModel = results2012.linearModel;
% Create a scatter plot
figure(5);
scatter(avgCPI, avgUnemployment);
xlabel('CPI');
ylabel('Unemployment');
title('Average CPI vs. Average Unemployment in 2012');
% Display regression results
coefficients = polyfit(avgCPI, avgUnemployment, 1);
fit_line = polyval(coefficients, avgCPI);
hold on;
plot(avgCPI, fit_line, 'r-', 'LineWidth', 2);
legend('Data Points', 'Regression Line');
% Create a residual plot
figure(6);
residuals = avgUnemployment - predict(linearModel, avgCPI);
scatter(avgCPI, residuals);
xlabel('CPI');
ylabel('Residuals');
title('Residual Plot: CPI vs. Unemployment in 2012');
hold on;
plot([min(avgCPI), max(avgCPI)], [0, 0], 'r--');
hold off;
```

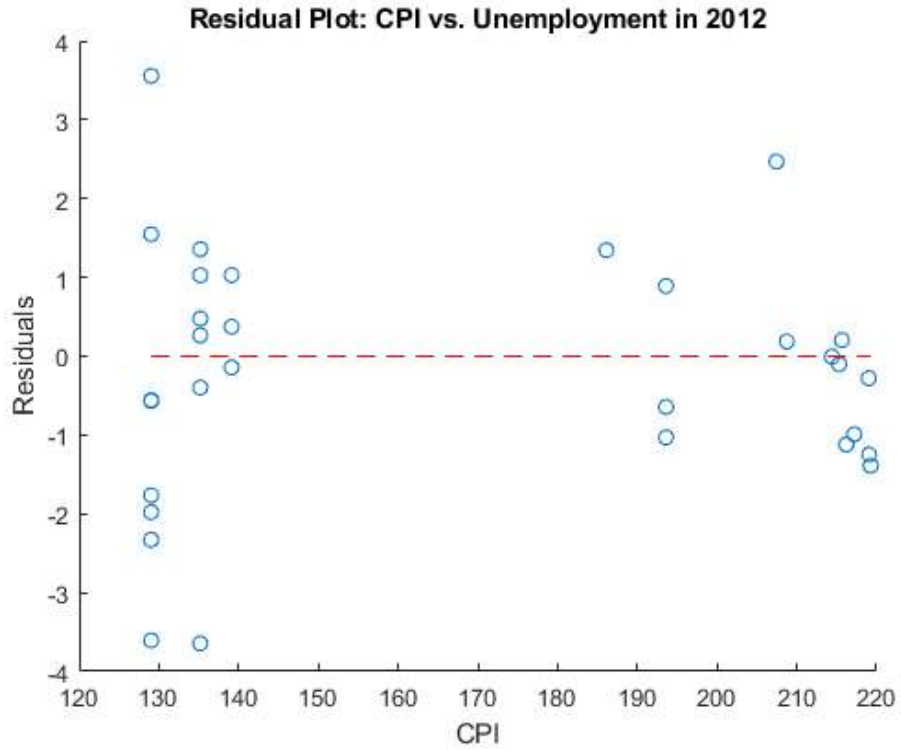

Average CPI vs. Average Unemployment in 2010
Data Points
Regression Line
Unemployment
CPI
15
14
13
12
11
10
9
8
7
6
5
120
130
140
150
160
170
180
190
200
210
220


Residual Plot: CPI vs. Unemployment in 2010
Residuals
CPI
5
4
3
2
1
0
-1
-2
-3
-4
120
130
140
150
160
170
180
190
200
210
220


Average CPI vs. Average Unemployment in 2011
Data Points
Regression Line
Unemployment
CPI
14
13
12
11
10
9
8
7
6
5
4
120
130
140
150
160
170
180
190
200
210
220


Residual Plot: CPI vs. Unemployment in 2011
Residuals
CPI
5
4
3
2
1
0
-1
-2
-3
-4
120
130
140
150
160
170
180
190
200
210
220





For the year 2010, the regression line was given by y=−0.0171x+11.3722 with an R2 value of 0.125. In 2011, the regression equation was y=−0.0119x+10.137 with an R2 of 0.0628, and for 2012, it was y=−0.0117x+9.4122 with an R2 of 0.0811. The closer the R2 value is to 1, the stronger the relationship between the variables. Based on our results, there appears to be only a marginal correlation between CPI and unemployment. To delve deeper into this relationship and to identify potential outliers or patterns, it would be beneficial to plot residual graphs.

The plot diagram shown illustrates the relationship between CPI and residuals. It can be used to determine if the errors are normally distributed, which is an important assumption in some regression models. On a residual plot, the x-axis typically represents the predicted values, and the y-axis represents the residuals. The better the linear fit for the data, the more scatter dots in the residual plot are randomly and uniformly distributed around the center (horizontal axis). In contrast, if the scatter points exhibit patterns or are not evenly distributed, a linear model may not be the best match. In the above figure, the scatter points are arranged irregularly, this shows that the values of R-square we obtained are reliable. However, there is a large gap in the center of the figure, which may be the result of missing data as mentioned before or the unreliability of the data provider. In the broadest sense, the CPI and unemployment rates are often inversely related. In fact, although the results we obtained indicate that the relationship between CPI and unemployment rate is small, it does not deny the relationship between them. As for the reason why this result is not obvious, it may be related to other factors (demand, economic

policy, etc.). In the third part, we conduct further analysis on the relationship between CPI and unemployment using different prediction models.

Part 3(a)

Further research in machine learning: Comparing linear and 5 order polynomial regression models for predicting CPI versus unemployment in 2010 (all data) using K-fold cross-validation.

In Section 3a, the objective is to delve deeper into the predictive modeling of the relationship between Consumer Price Index (CPI) and unemployment rate. We employ K-fold cross-validation to rigorously evaluate the performance of both linear and polynomial regression models. The K-fold approach ensures a robust assessment by partitioning the data set into multiple folds, allowing for comprehensive training and testing iterations. By comparing the average R-squared values across folds, this section aims to discern whether the added complexity of polynomial regression yields a significant improvement in prediction accuracy compared to the simpler linear model.

```
% Read the data
data = readtable('WALMART_SALES_DATA.csv');

% Preprocess the data
data.Date = datetime(data.Date, 'InputFormat', 'dd-MM-yyyy');
startdate = datetime('05-02-2010', 'InputFormat', 'dd-MM-yyyy');
enddate = datetime('31-12-2010', 'InputFormat', 'dd-MM-yyyy');
data2010 = data(data.Date >= startdate & data.Date <= enddate, :);

% Define the predictor and response variables
predictor = data2010.CPI;
response = data2010.Unemployment;

% Set K, the number of folds
K = 5;

% Create K equally sized folds
folds = crossvalind('Kfold', length(response), K);

% Model fitting and evaluation
order = 5; % Polynomial order
num_models = 2; % Number of models (1: Linear Regression, 2: Polynomial Regression)
r_square = zeros(K, num_models);
rng('default');
for i = 1:K
    % Segment data by fold
    testIndexes = (folds == i);
    testData = predictor(testIndexes);
    trainData = predictor(~testIndexes);
    trainResponse = response(~testIndexes);

     % Centering and scaling
    trainData = zscore(trainData);
    testData = zscore(testData);

    % Model fitting and evaluation
    % Model 1: Linear Regression
    fit_linear = fitlm(trainData, trainResponse);
    r_square(i, 1) = fit_linear.Rsquared.Ordinary;

    % Model 2: Polynomial Regression
    r_square_poly = zeros(1, order); % Initialize to store R-squared for each degree

    for j = 1:order
        fit_poly = polyfit(trainData, trainResponse, j);
        fit_test = polyval(fit_poly, testData);
        r_square_poly(j) = corr(fit_test, response(testIndexes))^2;
    end

    % Assign the maximum R-squared for this fold to the overall matrix
    r_square(i, 2) = max(r_square_poly);
end
```

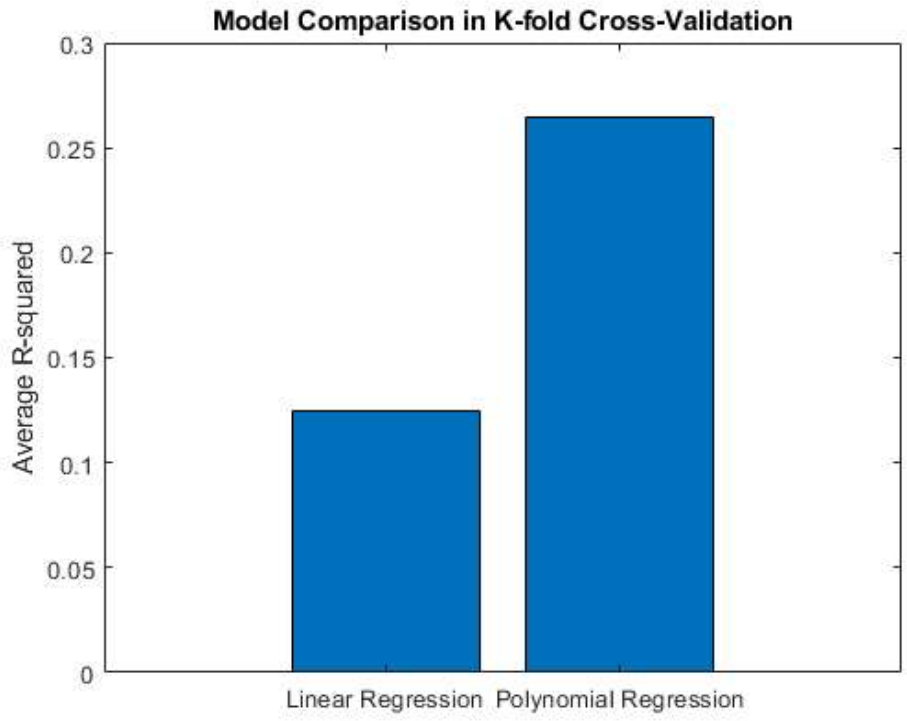
```
% Display R² values for each model in k-fold  cross-validation

r_square
```

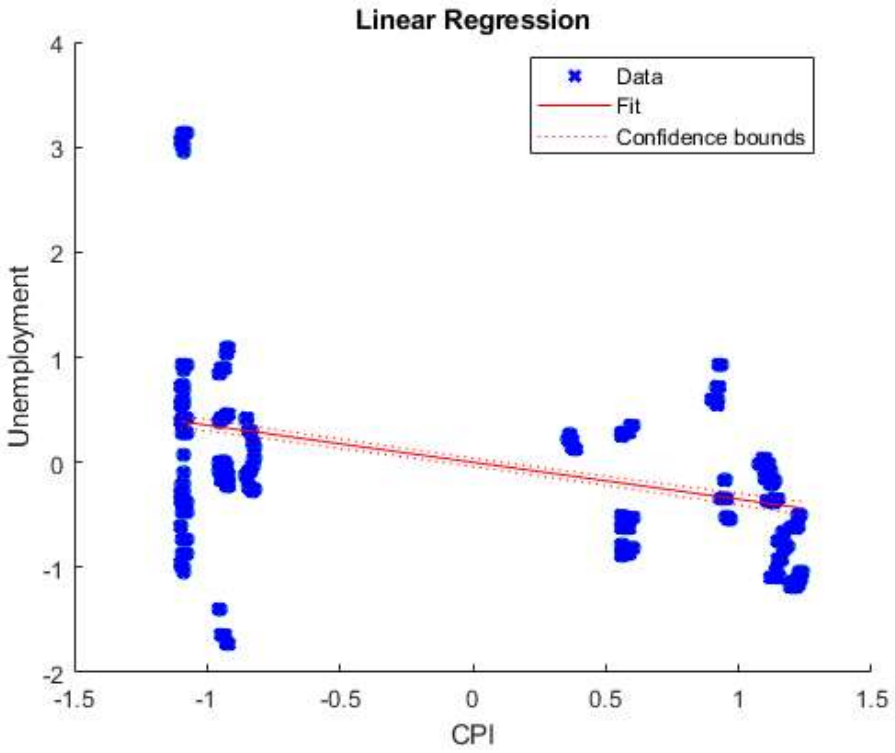
```
r_square =

    0.1172    0.3154
    0.1294    0.2130
    0.1244    0.3285
    0.1224    0.2863
    0.1288    0.1811
```

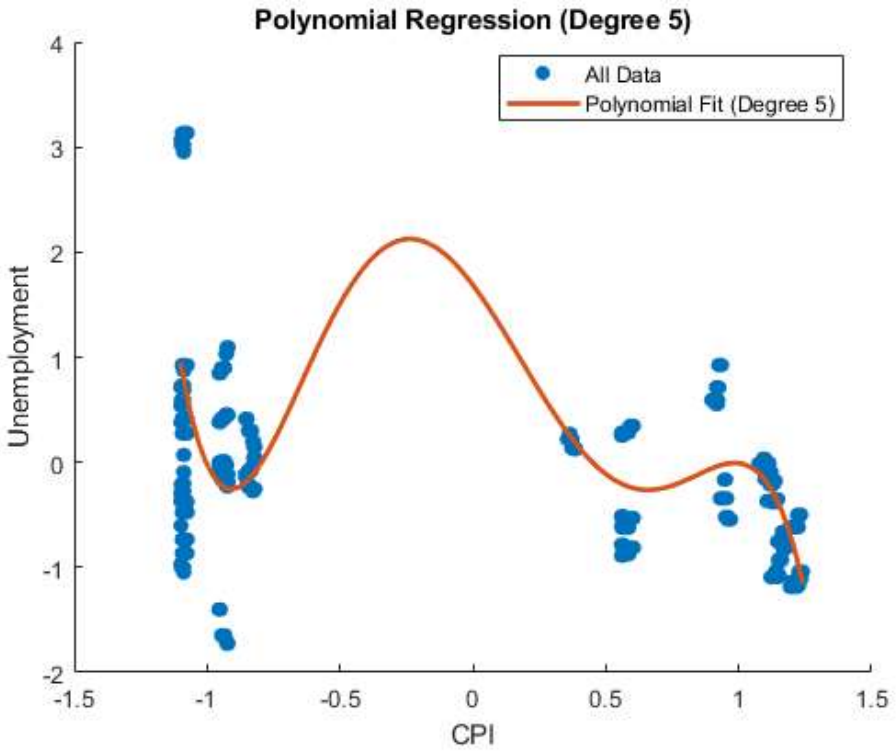
As shown in codes above, we created a linear regression line model and a five order polynomial model using the K-fold (K=5) cross-validation. The R^2 for the linear regression model are 0.1172, 0.1294, 0.1244, 0.1224 and 0.1288. These values from k-fold cross-validation are similar to the result obtained from a single train-test split, so the linear model's performance is consistent and robust across different subsets of the data. On the other hand, the R^2 for polynomial regression models are 0.3154, 0.2130, 0.3285, 0.2863 and 0.1811. It seems that R^2 in the polynomial model is larger than that in the linear model in general, which implies that the polynomial regression model provides a better fit to the data than the linear model. To further analyse the fitness of prediction between CPI and unemployment rate for these two models, we will calculate the average R^2 for each model over all folds and visualize both models.

```
% Averaging fit for each model

fits_kfold = mean(r_square);

% Plotting cross-validated prediction accuracy for each model
figure(7);
bar(fits_kfold);
xticks(1:num_models);
xticklabels({'Linear Regression', 'Polynomial Regression'});
ylabel('Average R-squared');
title('Model Comparison in K-fold Cross-Validation');

% Scatter plot of CPI versus Unemployment for all folds with average regression line
figure(8);

% Combine all data for scatter plot
allData = zscore([predictor, response]);  % Centering and scaling

% Model 1: Linear Regression
fit_linear_all = fitlm(allData(:, 1), allData(:, 2));

% Model 2: Polynomial Regression
fit_poly_all = polyfit(allData(:, 1), allData(:, 2), order);
x_values = linspace(min(allData(:, 1)), max(allData(:, 1)), 100); % 100 points for a smoother curve
fit_test_all = polyval(fit_poly_all, x_values);

% Scatter plot with regression line for Model 1: Linear Regression
scatter(allData(:, 1), allData(:, 2), 'filled', 'DisplayName', 'All Data');
hold on;
plot(fit_linear_all, 'DisplayName', 'Linear Fit', 'LineWidth', 2);
title('Linear Regression');
xlabel('CPI');
ylabel('Unemployment');
legend('Location', 'Best');
hold off;

% Scatter plot with regression line for Model 2: Polynomial Regression
figure(9)
scatter(allData(:, 1), allData(:, 2), 'filled', 'DisplayName', 'All Data');
hold on;
plot(x_values, fit_test_all, 'DisplayName', ['Polynomial Fit (Degree ', num2str(order), ')'], 'LineWidth', 2);
title(['Polynomial Regression (Degree ', num2str(order), ')']);
xlabel('CPI');
```

```
ylabel('Unemployment');
legend('Location', 'Best');
hold off;
```







Note: We don t really have interest in the details of models, only in its prediction accuracy. Thus, we are not writing out the prediction equation here. We simply retain the model objects. As shown in the figures above, the average R^2 for the regression line model and five-order polynomial model are 0.1244 and 0.2648 respectively.Visually, the polynomial model also looks better than the regression model. In conclusion, the analysis conducted on the Walmart sales data for the year 2010 using K-fold cross-validation reveals that the Polynomial Regression model, with a polynomial order of five, consistently outperforms the Linear Regression model in predicting unemployment based on the Consumer Price Index (CPI).

Part 3(b)

```
%Load data
data = readtable("WALMART_SALES_DATA.csv");

%Convert date format
data.Date = datetime(data.Date, 'InputFormat', 'dd-MM-yyyy');
data.Year = year(data.Date);
```

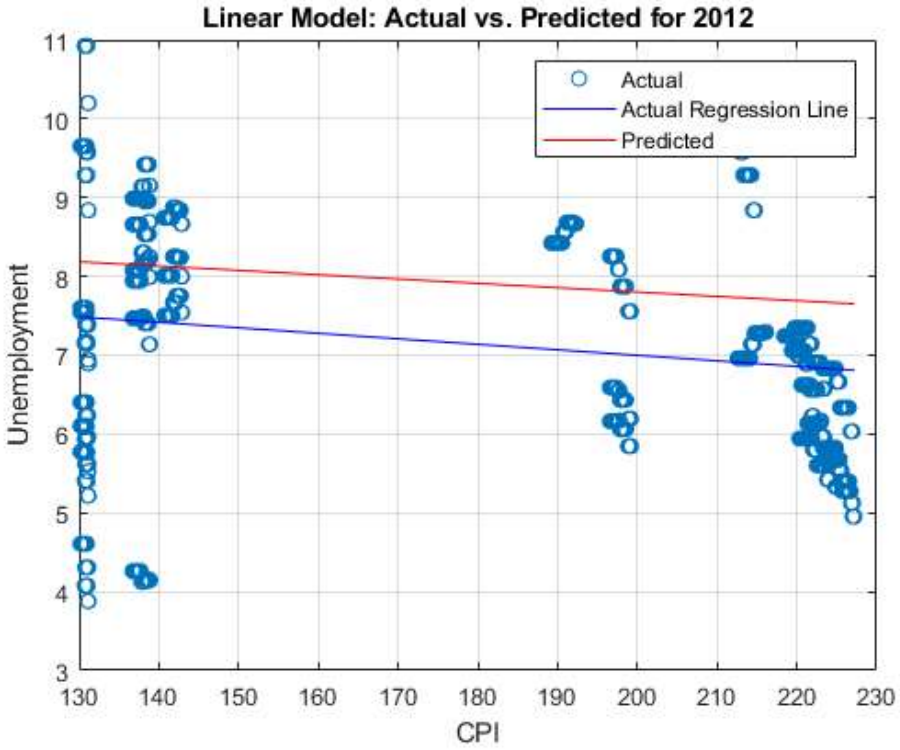
To create a machine learning model, we will be using 2010 and 2011 data to train the model in order to predict 2012 values. The success of these predictions will then be determined by comparing the predictions to the actual 2012 data provided. As shown in Part 2, the linear regression models for 2010-2012 data have quite a low r-squared value indicating that the relationship between CPI and Unemployment data is alomost not linear and a different model should be used to more accurately fit and predict Unemployment indices. Neural networks can model complex, non-linear relationships between input and output variables thus in this section, we will be comparing predictions made by training a linear model versus predictions made using a neural network.

Linear Model

```
%Extracting training data (2010-2011)
trainData = data(year(data.Date) ~= 2012, {'CPI', 'Unemployment'});
%Extracting testing data (2012)
testData = data(year(data.Date) == 2012, {'CPI', 'Unemployment'});

%Remove outliers from training and testing data
trainData = rmoutliers(trainData);
testData = rmoutliers(testData);

%Isolate CPI and Unemployment variables - we will use CPI as the predictor
%variable (X variable)
trainX = trainData.CPI;
trainY = trainData.Unemployment;
testX = testData.CPI;
```

```
testY = testData.Unemployment;

%Normalise data to ensure a consistent scale
minCPI = min(trainX);
maxCPI = max(trainX);
minUnemployment = min(trainY);
maxUnemployment = max(trainY);

% Apply Min-Max Scaling
trainX_normalised = (trainX - minCPI) ./ (maxCPI - minCPI);
trainY_normalised = (trainY - minUnemployment) ./ (maxUnemployment - minUnemployment);

% Apply the same scaling to the test data
testX_normalised = (testX - minCPI) ./ (maxCPI - minCPI);
testY_normalised = (testY - minUnemployment) ./ (maxUnemployment - minUnemployment);


%Creating a linear model using training data in order to predict
%2012 Unemployment rates
linearMdl = fitlm(trainX_normalised, trainY_normalised);

%Predicting 2012 unemployment rates based on trained linear model
lmPredictions_normalised = predict(linearMdl, testX_normalised);

%Un-normalising the predictions so they are the same scale to be compared
%with the actual values
lmPredictions = lmPredictions_normalised.*(maxUnemployment - minUnemployment) + minUnemployment;

%Obtain actual linear model for 2012 data
actualLinearMdl = fitlm(testX, testY);
coefficients = actualLinearMdl.Coefficients.Estimate;
intercept = coefficients(1);
slope = coefficients(2);
%This line will be plotted to compare against the predicted linear model
yValues = intercept + slope*testX;

figure(10);
%Plotting the actual 2012 test data points
plot(testX, testY, 'o', 'DisplayName', 'Actual');
hold on;
%Plotting the actual 2012 linear model regression line
plot(testX, yValues, 'b-', 'DisplayName', 'Actual Regression Line');
%Plotting the predicted 2012 linear model regression line
plot(testX, lmPredictions, 'r-', 'DisplayName', 'Predicted');
xlabel('CPI');
ylabel('Unemployment');
title('Linear Model: Actual vs. Predicted for 2012');
legend('show');
grid on;


%Assess performance of model predictions by calculating mean squared error
mse = mean((lmPredictions - testY).^2);

fprintf('Linear Model MSE: %f\n', mse);
```

```
Linear Model MSE: 2.610554
```

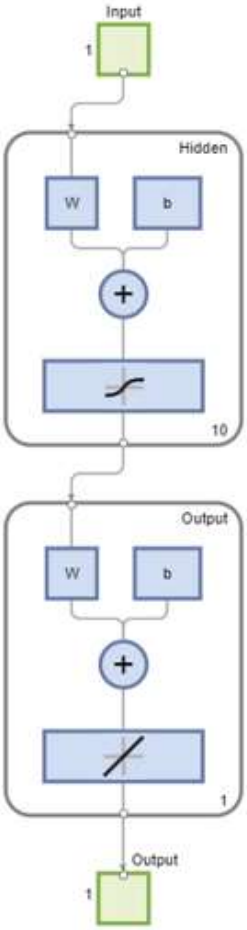


This MSE value indicates that on average, the squared differences between predicted values and actual values are relatively large. Thus, we will use a Neural Network to more accurately predict 2012 Unemployment indices.

Neural Network

```
%Creating a neural network with 1 hidden layer and 10 neurons
rng('default');
net = feedforwardnet(10);

%Train neural network (transpose trainX and trainY into column vectors)
net = train(net, trainX_normalised', trainY_normalised');
view(net)

%Estimate targets using the trained network
nnPredictions_normalised = net(testX_normalised');
nnPredictions = nnPredictions_normalised.*(maxUnemployment - minUnemployment) + minUnemployment;

%Assess performance of neural network by determining mean squared error
%(original scale)
perf = perform(net, testY', nnPredictions)
fprintf('Neural Network MSE: %f\n', perf);
```

```
perf =

    2.0190

Neural Network MSE: 2.019035
```



Network Diagram

Training Results

Training finished: Met validation criterion

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	10	1000
Elapsed Time	-	00:00:00	-
Performance	2.76	0.0327	0
Gradient	3.83	0.000199	1e-07
Mu	0.001	1e-05	1e+10
Validation Checks	0	6	6

Training Algorithms

Data Division: Random dividerand
Training: Levenberg-Marquardt trainlm
Performance: Mean Squared Error mse
Calculations: MEX

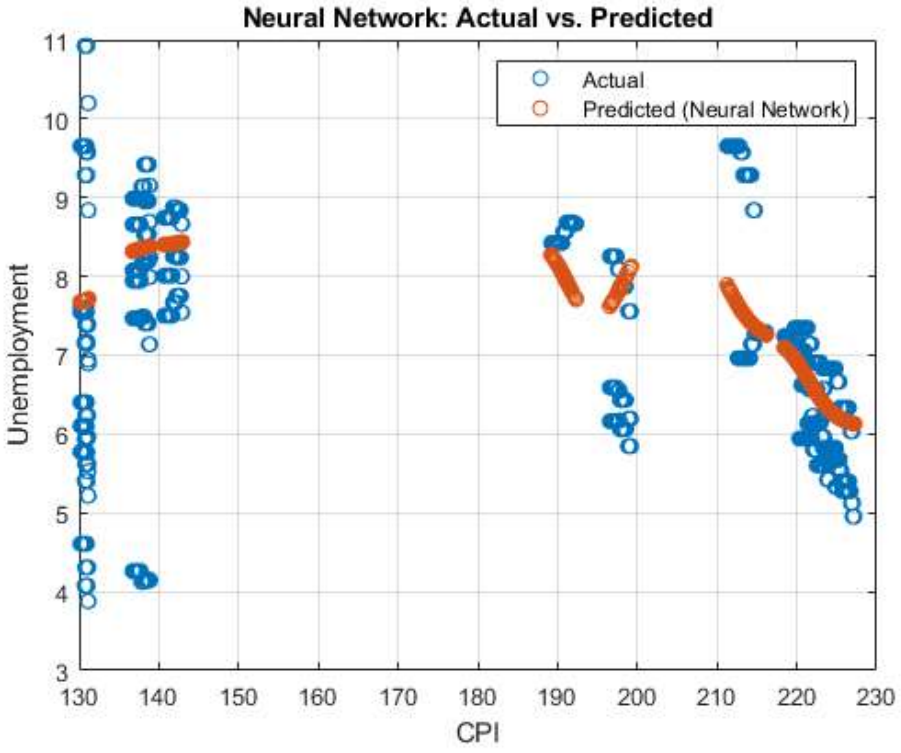
Training Plots

Performance | Training State

Error Histogram | Regression

To evaluate the model performance, we have used Mean Squared Error (MSE). A lower MSE indicates that, on average, the squared differences between the predicted values and the actual values are smaller. The linear model exhibited an MSE of 2.610554 while the neural network model exhibited an MSE of 2.019035. This lower MSE value suggests that the neural network was more accurate in predicting 2012 Unemployment indices; however, both MSE values are still quite similar.

```
figure(11);
%Plotting the actual 2012 test data points
plot(testX, testY, 'o', 'DisplayName', 'Actual');
hold on;
%Plotting the predictiosn from neural network
plot(testX, nnPredictions, 'o', 'DisplayName', 'Predicted (Neural Network)');
xlabel('CPI');
ylabel('Unemployment');
title('Neural Network: Actual vs. Predicted');
legend('show');
grid on;
```

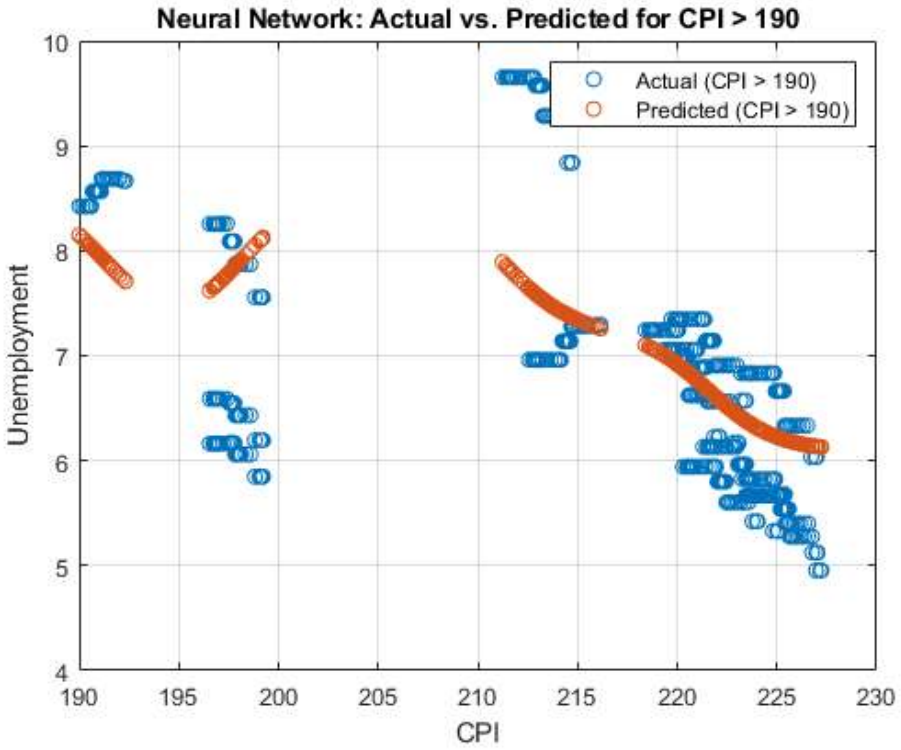


As seen in the figure 'Neural Network: Actual vs. Predicted', the predictions were a lot closer to actual values for high CPI values, specifically for CPI > 190.

```
% Filtering CPI > 190
cpi_threshold = 190;
indices = testX > cpi_threshold;
testX_filtered = testX(indices);
testY_filtered_actual = testY(indices);

% Normalize the filtered data
testX_filtered_normalised = (testX_filtered - minCPI) ./ (maxCPI - minCPI);

% Make predictions on the filtered data
nnPredictions_filtered_normalised = net(testX_filtered_normalised');
```

```
nnPredictions_filtered = nnPredictions_filtered_normalised.*(maxUnemployment - minUnemployment) + minUnemployment;

% Visualize the results for the filtered data
figure(12);
plot(testX_filtered, testY_filtered_actual, 'o', 'DisplayName', 'Actual (CPI > 190)');
hold on;
plot(testX_filtered, nnPredictions_filtered, 'o', 'DisplayName', 'Predicted (CPI > 190)');
xlabel('CPI');
ylabel('Unemployment');
title('Neural Network: Actual vs. Predicted for CPI > 190');
legend('show');
grid on;

perf = perform(net, testY_filtered_actual', nnPredictions_filtered)
fprintf('Neural Network MSE for CPI > 190: %f\n', perf);
```

```
perf =

    0.5518

Neural Network MSE for CPI > 190: 0.551842
```



We isolated the predictions for instances where CPI was greater than 190 as seen in the figure labeled 'Neural Network: Actual vs. Predicted for CPI > 190'. The MSE for the neural network model dropped significantly to 0.5518; this lower MSE indicates that the predicted values are much closer to the actual values. This shows that the neural network is better for predicting the target variable compared to the linear model, especially for higher CPI values.

Custom function for Part 2

```
function results = analyzeYear(year, dateFormat)
    % Read the data
    data = readtable('WALMART_SALES_DATA.csv');
    % Convert date to the desired format
    data.Date = datetime(data.Date, 'InputFormat', dateFormat);
    % Filter data for the specified year
    startDate = datetime(['01-01-' num2str(year)], 'InputFormat', dateFormat);
    endDate = datetime(['31-12-' num2str(year)], 'InputFormat', dateFormat);
    dataYear = data(data.Date >= startDate & data.Date <= endDate, :);
```

```
    % Calculate average CPI and unemployment for each store
    avgCPI = zeros(45, 1);
    avgUnemployment = zeros(45, 1);
    for store = 1:45
        storeData = dataYear(dataYear.Store == store, :);
        avgCPI(store) = mean(storeData.CPI);
        avgUnemployment(store) = mean(storeData.Unemployment);
    end
    % Fit a linear regression model
    linearModel = fitlm(avgCPI, avgUnemployment);
    % Store the results in a structure
    results.avgCPI = avgCPI;
    results.avgUnemployment = avgUnemployment;
    results.linearModel = linearModel;
end
```

University of Sydney Statement Concerning Plagiarism

```
% I certify that:

% (1) I have read and understood the University of Sydney Academic
% Dishonesty and Plagiarism Policy  (found by following the  Academic
% Honesty link on the Unit web page).';

% (2) I understand that failure to comply with the Student Plagiarism:
% Coursework Policy and Procedure can lead to the University commencing
% proceedings against me for potential student misconduct under Chapter 8
% of the University of Sydney By-Law 1999 (as amended).';

% (3) This Work is substantially my own, and to the extent that any part
% of this Work is not my own I have indicated that it is not my own by
% Acknowledging the Source of that part or those parts of the Work as
% described in the assignment instructions.';

% NAME1: Alicia Pan
% SID1: 520452677

% NAME2: Yancheng Lin
% SID2: 510022956

% NAME3: Zihan Ding
% SID3: 530658748

% DATE: 13-10-2023
```

Published with MATLAB® R2023a