Economic Exploration of Retail Sales Volume Predictions

Andrew Huang & Miguel Flores

12-11-2024

Project Contributions:

Andrew Huang: Introduction, Exploratory Data Analysis, Correlation Analysis, Full, Reduced, and Null Model Regression Analysis, Residual Analysis

Miguel Flores: K-fold Cross-Validation, Setup for Classification, PCA, K-Nearest Neighbors, Confusion Matrix, Conclusion, Formatting/Editing

Introduction

For our report, we are using retail sales data that shows five retail stores and the sales data associated with those stores. Our goal is to create effective prediction models for sales volume using only economic data. This means that our predictors in our model will focus on values related to inventory, demand, and price, all of which are related to economic theory and concepts. Theoretically this means that we should see price and demand working together to influence our quantity (sales volume) prediction, which is represented as our "Units.Sold" variable in our analysis. While searching for an effective model fit, we will be paying close attention to how these variables interact with each other within our models. We have broken our project into two parts. In part one we will perform an exploratory data analysis before attempting to find a great predictive linear model fit. We want to find what variables will be important for reducing the error between our model and the actual values for units sold. Naturally this step entails creating multiple linear regression models and testing their effectiveness with cross validation. For part two we take a different approach to prediction which instead plans to give us a more general classification prediction for sales volume. The idea is that this approach removes any assumptions of linearity/distribution and instead relies on unsupervised learning techniques to make predictions. For this method we will perform both a principal component analysis for dimension reductionality and a k-nearest neighbors algorithm to classify new values based on sales volume.

Part 1: Searching for a Predictive Linear Model

Exploratory Data Analysis

```
## # A tibble: 3 x 7
##   Inventory.Level Units.Sold Units.Ordered Demand.Forecast Price Discount
##             <int>      <int>         <int>           <dbl> <dbl>    <int>
## 1             231        127            55           135.   33.5       20
## 2             204        150            66           144.   63.0       20
## 3             102         65            51            74.0  28.0       10
## # i 1 more variable: Competitor.Pricing <dbl>
```

We have read in our full retail dataset which has 15 variables and 73,100 observations. Above, we have removed all categorical variables from our dataset to focus our analysis on continuous variables. While we understand the significance these categorical variables may hold, for the sake of our analysis, we want to concentrate our predictive analysis on numeric columns, because we want to see how great of a model we

can create using variables that theoretically pertain to conceptually economic ideas. We have also found no missing or null values in our data so we did not have to remove any observations.

From our new reading of the retail data we can see that all of our values are numeric in that they either have class type 'int' or 'dbl.' We observe that there are 73,100 observations in total across the seven variables: "Inventory.Level", "Units.Sold", "Units.Ordered", "Demand.Forecast", "Price", "Discount", "Competitor.Pricing".

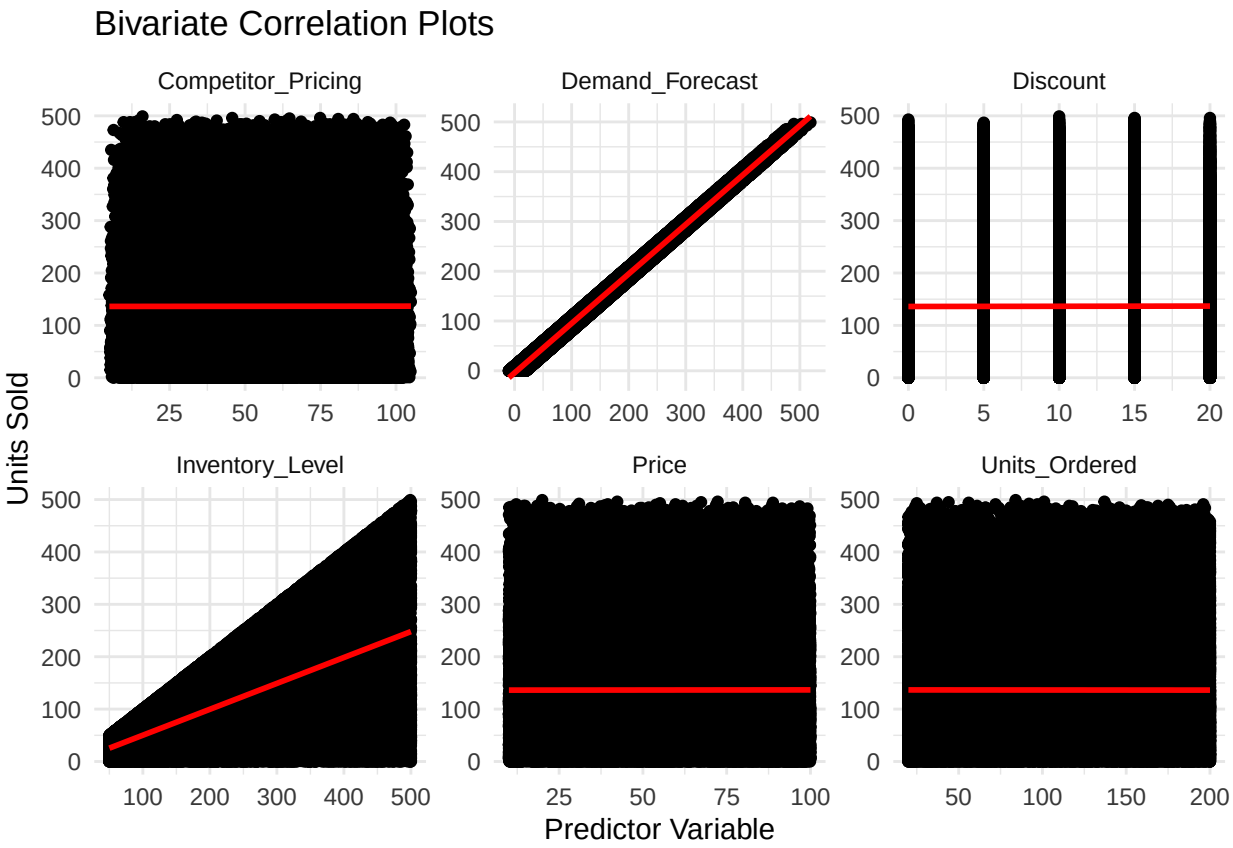
From our data summary we observe that the majority of variables have medians and means with similar values except for "Demand.Forecast", which shows that the data for this variable is skewed right, and that there may be outliers lying within.

```
## # A tibble: 7 x 7
##   Variable             Min `Q1.25%` Median  Mean `Q3.75%`   Max
##   <chr>              <dbl>    <dbl>  <dbl> <dbl>    <dbl> <dbl>
## 1 Inventory.Level    50       162    273   274.     387    500
## 2 Units.Sold          0        49    107   136.     203    499
## 3 Units.Ordered      20        65    110   110.     155    200
## 4 Demand.Forecast    -9.99     53.7  113.  141.     208.   519.
## 5 Price              10        32.6   55.0  55.1     77.9  100
## 6 Discount            0         5     10    10.0     15     20
## 7 Competitor.Pricing  5.03     32.7   55.0  55.1     77.8  105.
```

Correlation Analysis of Bivariate Models

Below we continue our exploration of our variables with bivariate correlation analysis in which we plot our target ("Units.Sold") against every covariate individually. Before conducting a regression analysis, we want to understand the potential predictive power each covariate has on our sales volume to get an understanding of each variable's potential importance to the overall model.

```
## `geom_smooth()` using formula = 'y ~ x'
```



From these plots we can see that only Demand.Forecast and Inventory.Level have any visible correlation to Units.Sold. All other charts seem to show no direct correlation between the predictor and target variables, including price, which is a surprising finding considering its expected effect on quantity in economic theory. Going forward we will consider these results in the creation of our reduced regression model.

Regression Analysis

Full Model

Our Full Model: $Y = B_0 + B_1x_1(Inventory.Level) + B_2X_2(Units.Ordered) + B_3X_3(Demand.Forecast) + B_4X_4(Price) + B_5X_5(Discount) + B_6X_6(Competitor.Pricing) + \epsilon$ This is our full model because it considers every (chosen) variable in our dataset.

Summary Statistics

```
## # A tibble: 7 x 5
##   Variable            Estimate Std..Error  t.value Pr...t..
##   <chr>                  <dbl>      <dbl>    <dbl>    <dbl>
## 1 (Intercept)        -5.05       0.128     -39.3   0
## 2 Inventory.Level     0.00421    0.000304   13.9   1.27e-43
## 3 Units.Ordered       0.000329   0.000610    0.540 5.89e- 1
## 4 Demand.Forecast     0.991      0.000361 2744.    0
## 5 Price              -0.00217    0.0111     -0.195 8.45e- 1
## 6 Discount            0.00348    0.00450     0.772 4.40e- 1
## 7 Competitor.Pricing  0.00369    0.0110      0.334 7.38e- 1
```

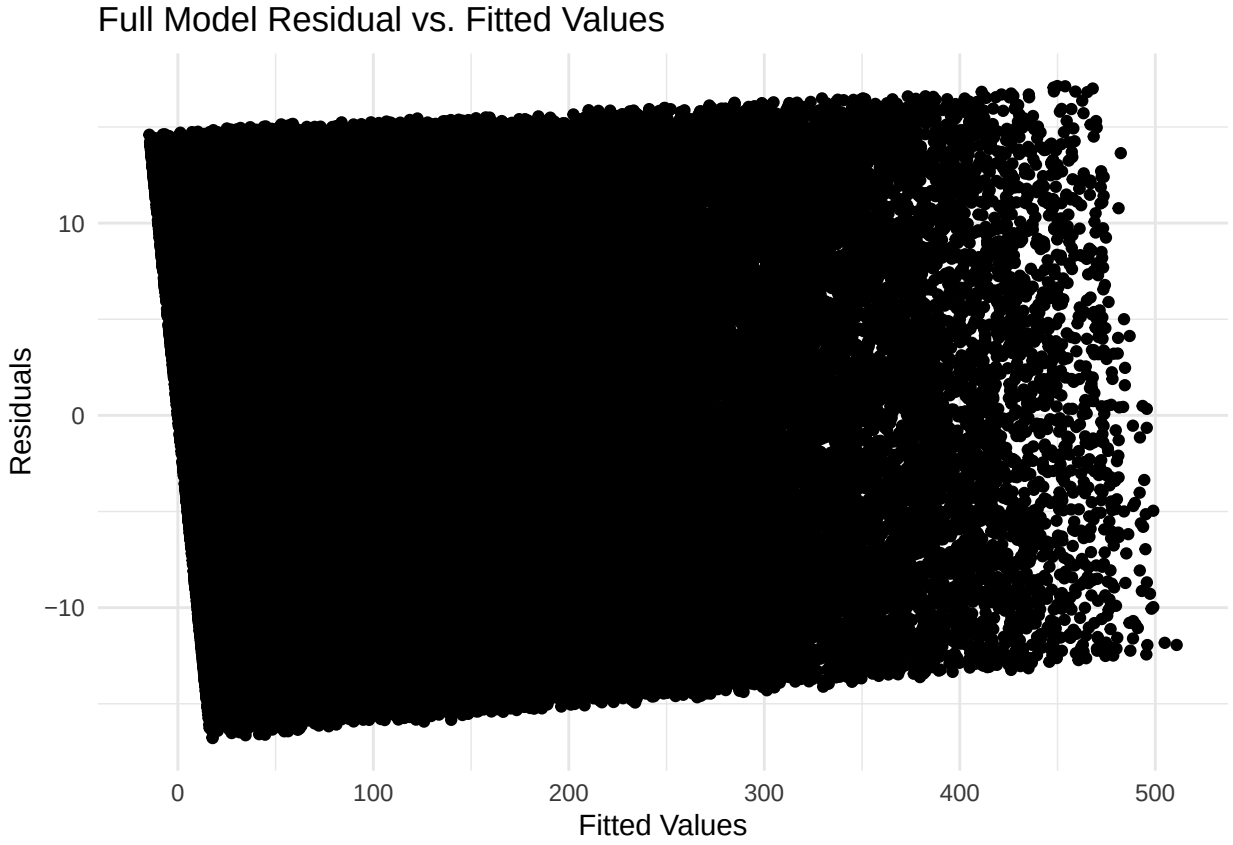
While we can't claim that they are simultaneously significant, it is worth nothing that "Inventory.Level" and "Demand.Forecast" have p-values less than a typical $\alpha = 0.5$. The effect of demand on volume also appears to be about a 1 unit increase in volume per increase in demand. The effect size for inventory is much smaller.

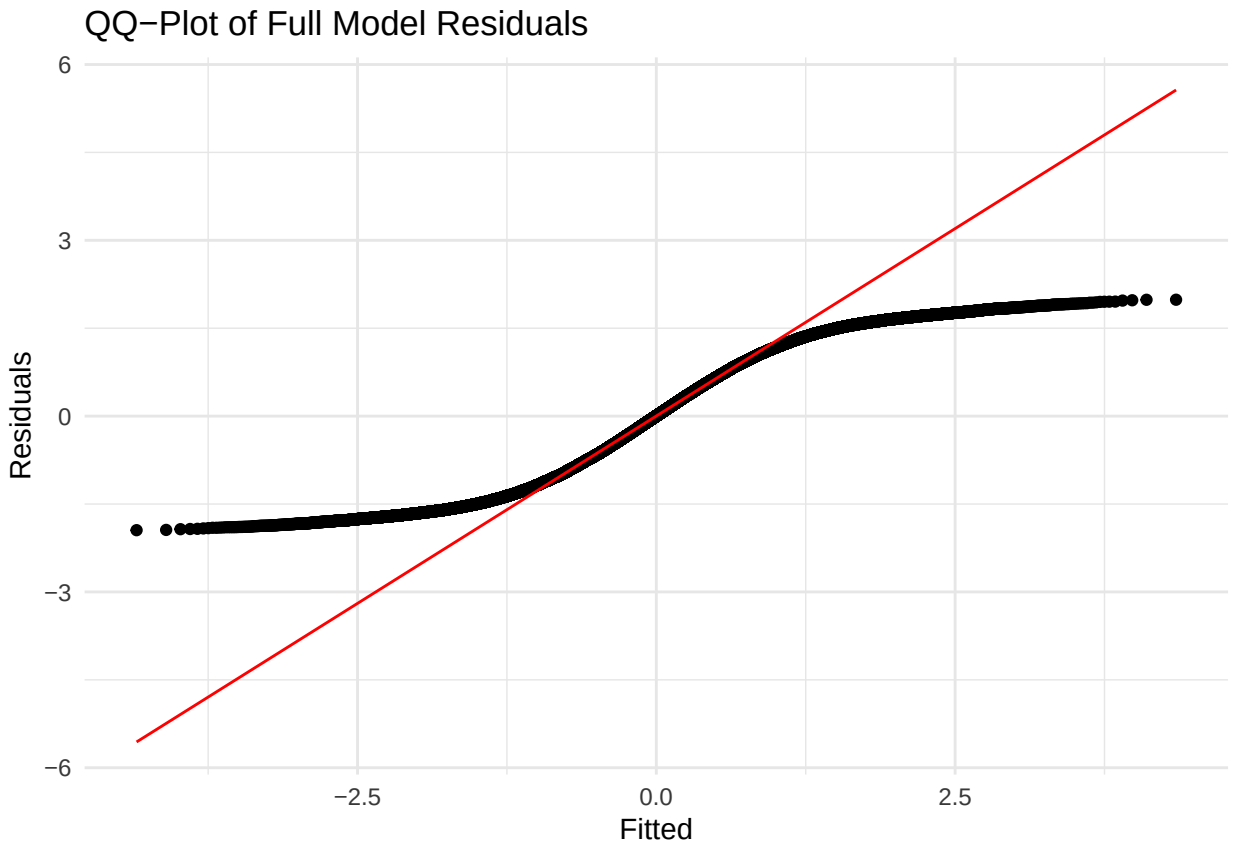
R Squared and Adjusted R Squared

```
## # A tibble: 2 x 2
##   Statistic          Estimate
##   <chr>                 <dbl>
## 1 R-Squared             0.994
## 2 Adjusted R-Squared    0.994
```

Above we observe a high R^2 value of 0.994, which means 99.4% of of the volume's variability can be explained by our model. This suggests that we have a very strong fit. Additionally R^2 and Adj R^2 have similar values meaning that model predictors are meaningful.

Residual Analysis





Our residual analysis for the full model shows that the residual assumptions of homoscedasticity and normality have been violated. The scatterplot depicts a violation of homoscedasticity as the spread of residuals increases with increasing fitted values. The qq-plot demonstrates non-normal residuals since the points stray from the red line. Ideally these points line up with the red line, essentially meaning that the real residuals follow a theoretical normal distribution's residual pattern. Given that both of these were violated, in an inferential study, we may have to put a pause on our analysis to transform the model. However, our goal is predictive, and these assumptions, while they should not be taken lightly, do not necessarily change our trajectory. Instead, to measure our predictions, we will use MSE to analyze our model fit, a metric that does not necessarily rely on these assumptions to move forward.

Reduced Model

Our Reduced Model: $Y = B_0 + B_1 x_1(Inventory.Level) + B_2 X_2(Demand.Forecast) + \epsilon$ We've chosen this as our reduced model because as we noted earlier, the included variables are the only variables that have any visible correlation to sales volume.

Summary Statistics

```
## # A tibble: 3 x 5
##   Variable        Estimate Std..Error t.value Pr...t..
##   <chr>              <dbl>      <dbl>   <dbl>    <dbl>
## 1 (Intercept)     -4.89      0.0746     -65.6 0
## 2 Demand.Forecast  0.991     0.000361  2744.  0
## 3 Inventory.Level  0.00421   0.000304    13.9 9.64e-44
```

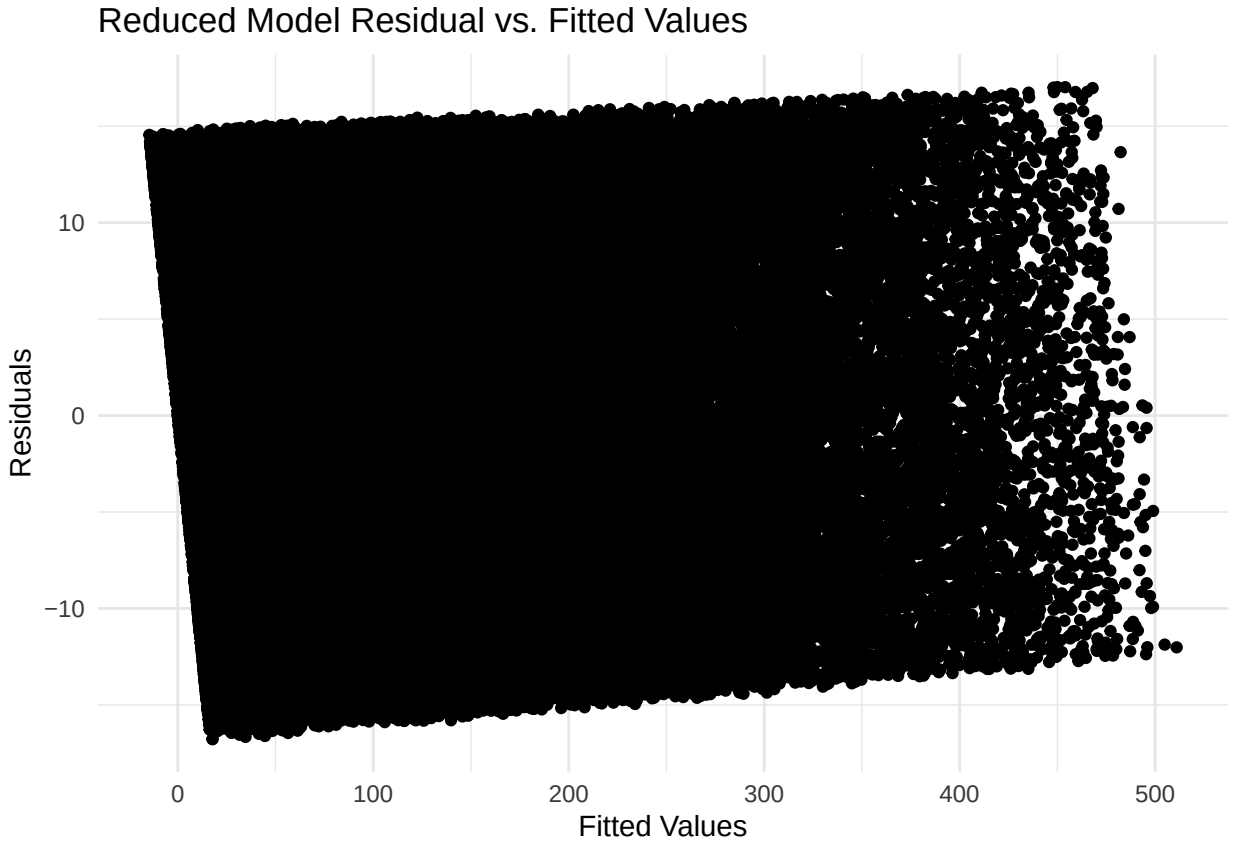
Again we notice that "Inventory.Level" and "Demand.Inventory" appear to possibly but not certainly be significant variables under $\alpha = 0.5$. Similarly, the effect of demand on volume still appears to be about a 1 unit increase in volume per increase in demand. The effect size for inventory is again much smaller.

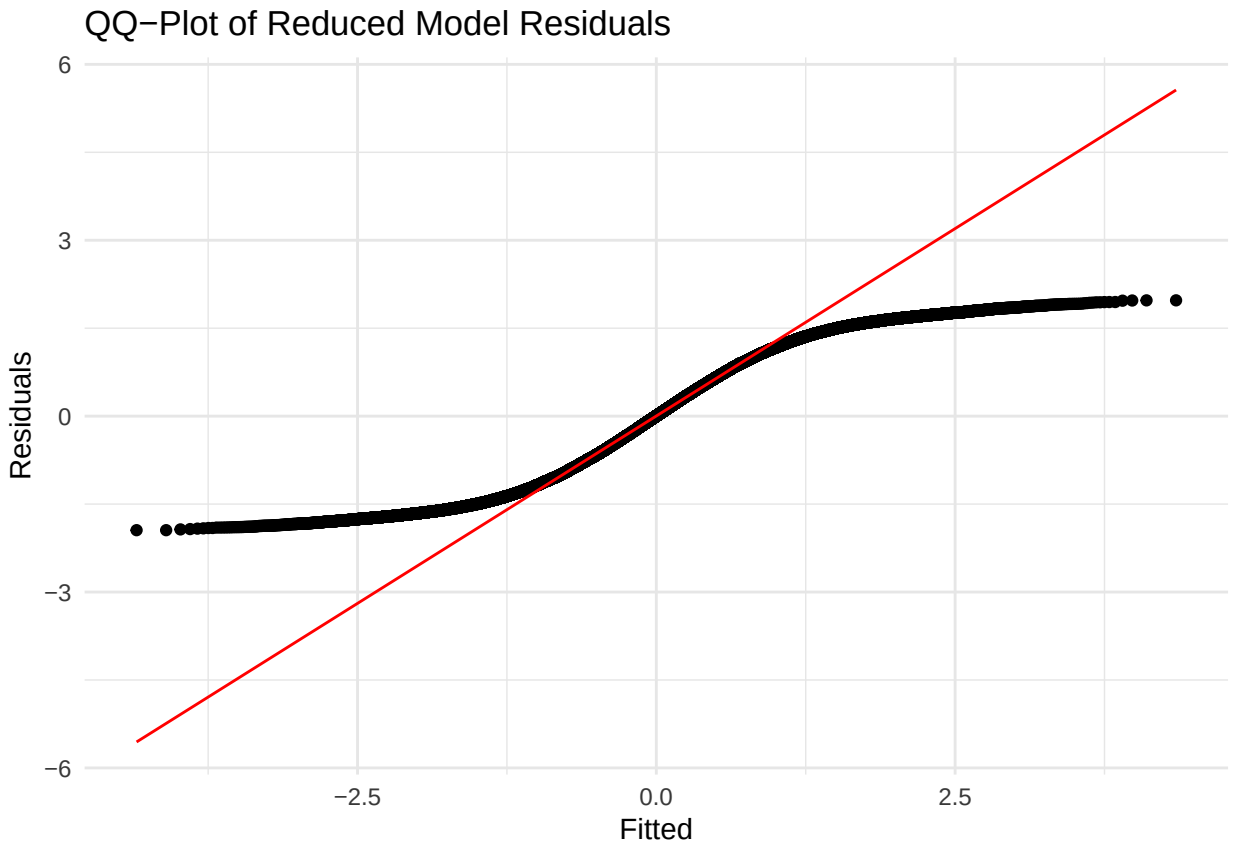
R Squared and Adjusted R Squared

```
## # A tibble: 2 x 2
##   Statistic          Estimate
##   <chr>                 <dbl>
## 1 R-Squared             0.994
## 2 Adjusted R-Squared    0.994
```

We also again observe a high R^2 value of 0.994, which means 99.4% of of the volume's variability can be explained by our model, suggesting that we have a very strong fit. Additionally R^2 and Adj R^2 also have similar values. Interestingly, the full model has a very slightly higher R^2 while this reduced model has a very slightly higher Adj R^2 value.

Residual Analysis





While we may have hoped that with the reduction of our model we would solve the issues of non-normality and heteroscedasticity, unfortunately the problem persists. Like with the previous model, this doesn't stop our analysis and we will proceed with the direction of our report.

Null Model

Our Null Model: $Y = B_0 + \epsilon$ This is the null model because it removes all predictor variables and provides a baseline comparison for our other models.

Summary Statistics

```
## # A tibble: 1 x 5
##   Variable    Estimate Std..Error t.value Pr...t..
##   <chr>          <dbl>      <dbl>   <dbl>    <dbl>
## 1 (Intercept)     136.      0.403    339.        0
```

We see above that the intercept is significant at $\alpha = 0.05$, making it an important inclusion for our model just as it had been for the past two models. The effect size is also large just like before indicating that our "Units.Sold" may be a largely static quantity that only changes by large amounts when other variables (such as for example demand) change by large amounts.

R Squared and Adjusted R Squared

```
## # A tibble: 2 x 2
##   Statistic          Estimate
##   <chr>                 <dbl>
## 1 R-Squared                 0
## 2 Adjusted R-Squared        0
```

Our summary data shows that the reduced model has worse fit with an R^2 of 0 meaning none of the volume's variability can be explained by the intercept. Given this information we are led to believe that because the

R^2 values of our previous two models are higher, that those models are a better fit.

Cross-Validation

We can confirm our suspicion with a k-fold cross-validation algorithm, two of which are seen in the table below. Essentially these algorithms split the data "k" number of times into k "folds", giving each fold the opportunity to be the test dataset in a train-test split of our data. After training the model on a subset of the data, we measure its success with a test (fold) subset of the data and repeat the process multiple times taking the average of the calculated prediction metric used for each iteration. For our purposes we are using MSE as our metric and a value of k=5.

```
## # A tibble: 3 x 2
##   Model       MSE
##   <chr>     <dbl>
## 1 Full       74.4
## 2 Reduced    74.4
## 3 Null    11864.
```

We can see that the full model and reduced models have a significantly smaller MSE compared to the null model, meaning they are a much better fit with less error between actual values and predicted values. Between the full and reduced model, the k-fold algorithm tells us that the reduced model is a slightly better predictive fit. Given that this model is also smaller and easier to use, we undoubtedly would rather use the reduced model with only two covariates. Between all of these models, we conclude that the reduced model is the best predictive fit for measuring sales volume. Implicitly, this means that of our continuous economics-related variables, only demand and inventory are relevant in the prediction of anticipated retail sales volume for these stores!

Part II: Classification of Sales

Now that we have a strong predictive linear regression model to predict sales volume, we take another approach to solidify our predictions in a way that does not involve linear assumptions at all and instead uses patterns in our data to give us a classification estimate of general sales volume.

Setup for Classification

For setup we split the "Units.Sold" target variable into a 3 part categorical variable. Low is 0-165, Medium is 165-330, High is 330+. These amounts can be adjusted depending on what store owners may determine to be low, medium, and high sales, but for now we will consider these numbers to be valid. Afterwards we delete the "Units.Sold" variable as it has been replaced with this categorical variable.

```
## # A tibble: 3 x 7
##   Inventory.Level Units.Ordered Demand.Forecast Price Discount
##             <int>         <int>           <dbl> <dbl>    <int>
## 1             379           154            363.  93.0       15
## 2             460            70            401.  91.1        5
## 3             465            77            370.  91.3       20
## # i 2 more variables: Competitor.Pricing <dbl>, Sales.Volume <ord>
```

Principal Component Analysis Dimension Reduction

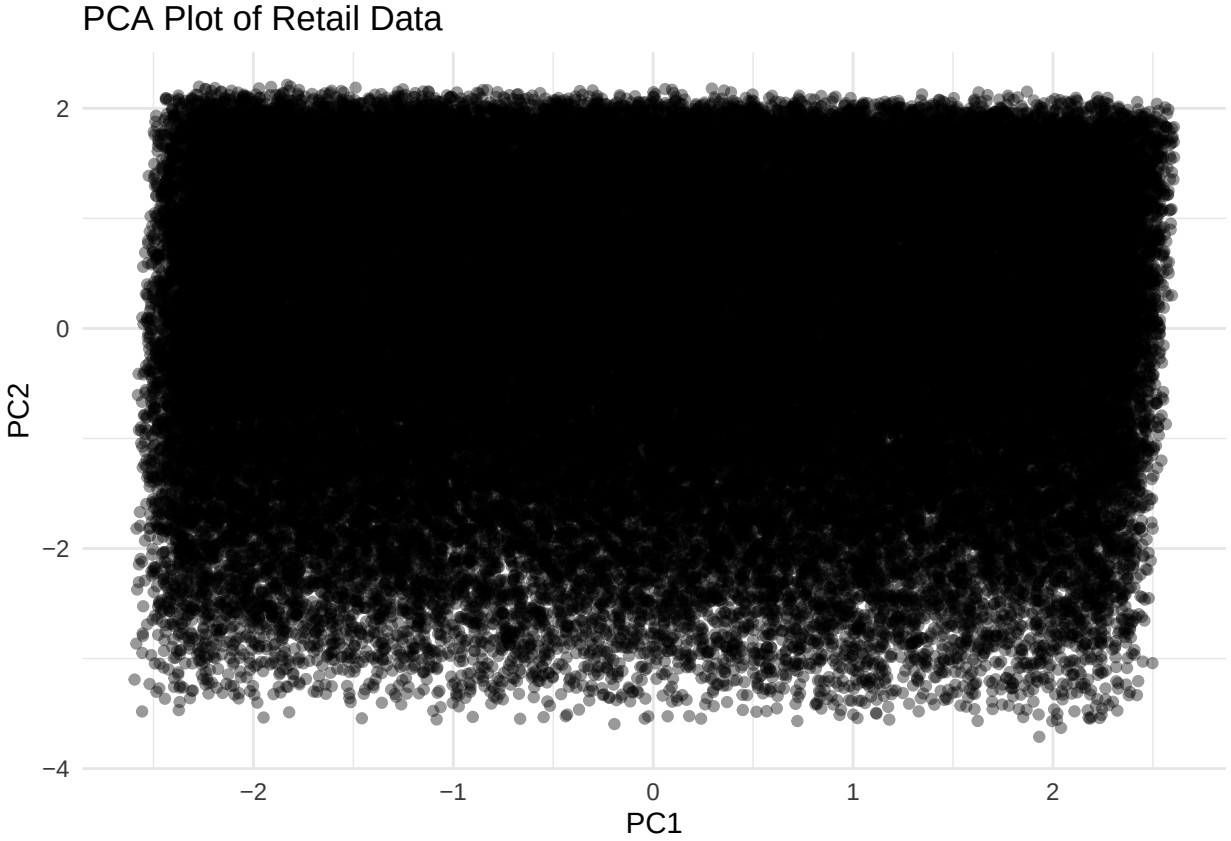
Below, we perform a PCA dimension reduction, essentially breaking down our data by its most relevant components in which relevancy is defined as the amount of variance it explains. The first component explains the most and so on. The values in the table represent how much each component is influenced by every included variable. It appears as if Price and Competitor.Price influence PC1 the most while Demand.Forecast and Inventory.Level influence PC2 the most.

```
## # A tibble: 6 x 6
##        PC1       PC2      PC3      PC4      PC5        PC6
##      <dbl>     <dbl>    <dbl>    <dbl>    <dbl>      <dbl>
## 1 -0.0213  -0.707    -0.00414  0.00147  0.707    0.000135
## 2 -0.00627  0.000326 -0.709   -0.705   -0.00254  0.000273
## 3 -0.0137  -0.707    -0.00465  0.00702 -0.707    0.0000395
## 4 -0.707    0.0176    0.00193  0.00463 -0.00383  0.707
## 5 -0.00260 -0.00858   0.705   -0.709   -0.00305  0.000318
## 6 -0.707    0.0174    0.00198  0.00404 -0.00374 -0.707
```

Below we highlight the fact that about 60% of the data's overall variance is explained by the first 2 principal components so for the purposes of our report, we will only take into consideration PC1 and PC2.

```
## # A tibble: 3 x 4
##   Variable Standard_Deviation Proportion_of_Variance Cumulative_Variance
##   <chr>                 <dbl>                  <dbl>               <dbl>
## 1 PC1                    1.41                  0.332               0.332
## 2 PC2                    1.26                  0.265               0.597
## 3 PC3                    1.00                  0.167               0.764
```

At this point, the "Sales.Volume" variable we created earlier has been carefully added to the PCA dataframe and will be referred to as "Volume".

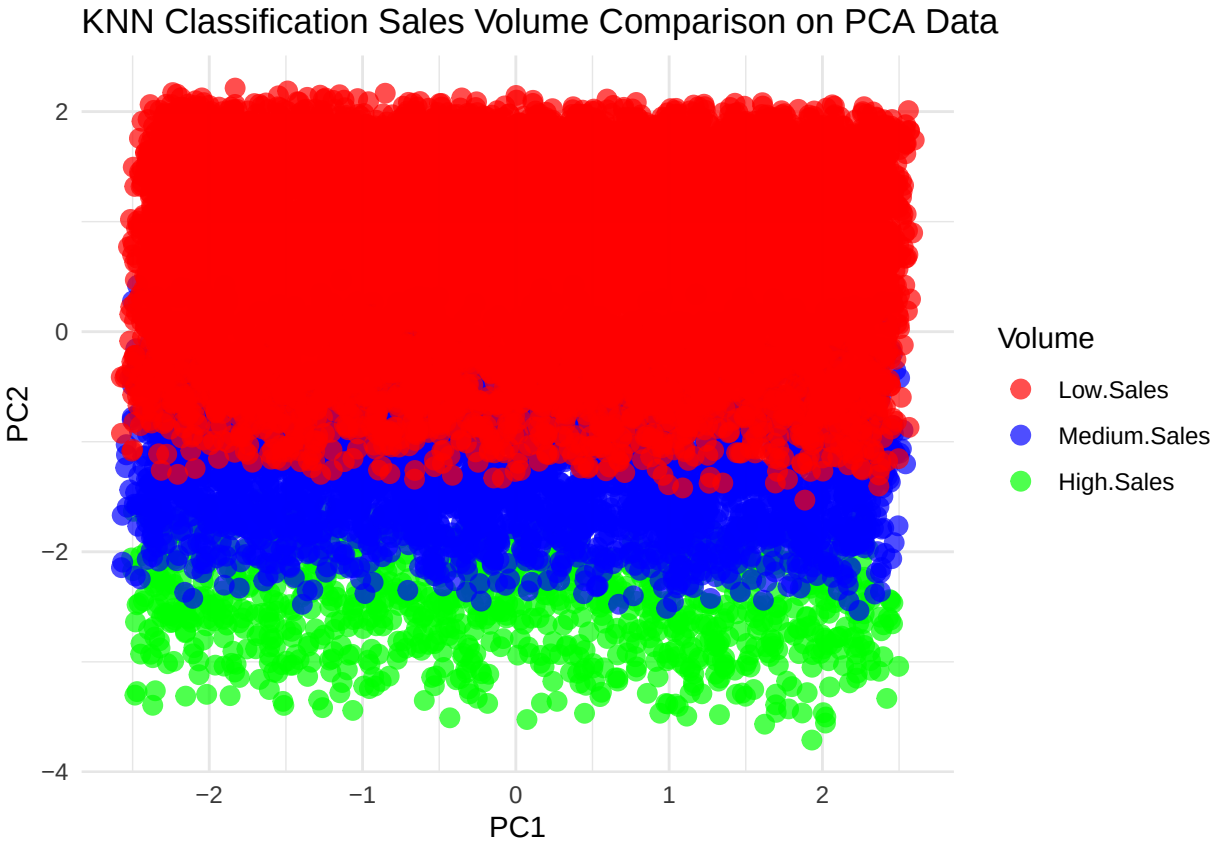


Above is the initial PCA plot with no labels attached, simply showing how much each point varies as a result of PC1 and PC2.

K-Nearest Neighbors

Next we have implemented a k-nearest neighbors algorithm using the PCA plot from before to help with classification of our sales volume. Like k-fold cross validation, this splits our data into training and test

subsets. Classification is done by placing an observation into the group in which most of its closest points are classified as. K here represents how many of these closest points are considered. We chose 5 neighbors and receive the plot below:



Confusion Matrix Validation We test the validity of this classification with a confusion matrix below comparing how test values were classified versus the actual class in which they belong to.

```
## # A tibble: 3 x 4
##   Predicted    Low.Sales Medium.Sales High.Sales
##   <fct>            <int>        <int>      <int>
## 1 Low.Sales         8917         1047          0
## 2 Medium.Sales       917         2588        211
## 3 High.Sales           0          165        775
```

Accuracy:

```
## [1] 0.8399453
```

Our accuracy is at about 84%, which for our purposes is a great accuracy rate and means that our classification works most of the time. This means we can safely rely on our classification for the sake of this retail scenario.

Given what we know about what each principal component is largely made out of, it isn't a big assumption to say that PC1 mainly represents price/competitor price while PC2 mainly represents demand/inventory levels in the PCA plot. So then in our KNN classification, if we what we said in part 1 about the predictive power of demand/inventory is true, we should be seeing almost no predictive power from price/competitor price(the x-axis) and should only see variation in classification because of demand/inventory (y-axis). So then it is not surprising that this is exactly what we see in our KNN classification as each group can almost clearly be divided horizontally with increasing y-values! This once again confirms our results from part 1 about how important demand and inventory are as predictors for sales volume while price and other variables have no significant prediction power.

Conclusion

It is worth noting that the results from part 1 come from a linear model, when in reality a linear model may not actually be the absolute best fit. However, it is undeniable that the selected reduced model is still great and has an incredibly low MSE value, indicating high predictive power. To ensure that this linearity hurdle would not be a burden on our analysis, part 2's classification approach made no assumptions about linearity and yet we got an interpretation similar to part 1. Therefore in our pursuit of effective prediction models, we created 2 distinct models (a linear model and a classification model) that both have high predictive power and both led to the interpretation that only demand and inventory have any predictive utility. Perhaps price has limited information because of the short time-horizon being measured in our sales volume (single day), giving it more inelastic properties and not having any effect on quantity. However, without making any extreme assumptions, we cannot confidently say why price has no influence here. It can be said though, that it makes the job of store management easier knowing that if they ever want to predict sales volume for a particular day, they only need to obtain values for their demand forecast and inventory levels without analyzing the more complex relationship between demand and price.

```
knitr::opts_chunk$set(echo = TRUE)
#Loading in Libraries
library(tidyverse)
library(dplyr)
library(ggplot2)
library(class)
library(tibble)
library(factoextra)
retail <- read.csv("~/Downloads/retail_store_inventory.csv")
retail <- as_tibble(retail)
#Creates a subset of the data without categorical variables
retail <- subset(retail, select = -c(Date, Store.ID, Product.ID, Category, Region, Weather.Condition, H
retail <- as_tibble(retail)
head(retail,3)
#Implements a function that gives us summary statistics of every variable
summary_data <- as.data.frame(do.call(rbind, lapply(retail, function(column) {
  if (is.numeric(column)) {
    c(Min = min(column, na.rm = TRUE),
      Q1 = quantile(column, 0.25, na.rm = TRUE),
      Median = median(column, na.rm = TRUE),
      Mean = mean(column, na.rm = TRUE),
      Q3 = quantile(column, 0.75, na.rm = TRUE),
      Max = max(column, na.rm = TRUE))
  } else {
    c(Levels = length(unique(column)),
      Most_Frequent = names(sort(table(column), decreasing = TRUE)[1]),
      Frequency = max(table(column)))
  }
})))
summary_data <- as_tibble(summary_data, rownames = "Variable") #tibble transformation of data frame
summary_data
#Creates a list of bivariate models
models <- list(
  Inventory_Level = lm(Units.Sold ~ Inventory.Level, data = retail),
  Units_Ordered = lm(Units.Sold ~ Units.Ordered, data = retail),
  Demand_Forecast = lm(Units.Sold ~ Demand.Forecast, data = retail),
  Price = lm(Units.Sold ~ Price, data = retail),
  Discount = lm(Units.Sold ~ Discount, data = retail),
```

```
  Competitor_Pricing = lm(Units.Sold ~ Competitor.Pricing, data = retail))
#Binds the models into a dataframe
model_results <- do.call(rbind, lapply(names(models), function(model_name) {
    model <- models[[model_name]]
    if (length(coef(model)) > 1) {
      predictor <- names(coef(model))[-1]
      data.frame(
        Model = model_name,
        Response = retail$Units.Sold,
        Variable = retail[[predictor]],
        Predictor = predictor)
 }  else {
      data.frame(
        Model = model_name,
        Response = retail$Units.Sold,
        Variable = rep(NA, nrow(retail)),
        Predictor = "Intercept")
    }
  })
)
#Plots the correlation plots
Response_plot <- ggplot(model_results, aes(x = Variable, y = Response)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  facet_wrap(~Model, scales = "free") +
  theme_minimal() +
  labs(title = "Bivariate Correlation Plots", x = "Predictor Variable", y = "Units Sold")
Response_plot
#Full model with all relevant variables included as a predictor variable
full <- lm(Units.Sold~., retail)
full_model_summary <- summary(full)

#Creates dataframe for important coefficients
main_summary_table <- data.frame(full_model_summary$coefficients)%>%
  as_tibble(rownames = "Variable")

main_summary_table <- as_tibble(main_summary_table) #Creates tibble of the dataframe for cleaner look
main_summary_table
#This gives us the R-Squared and Adjusted R-Squared in a tibble of its own
additional_stats_table <- tibble(
  Statistic = c("R-Squared", "Adjusted R-Squared"),
  Estimate = c(full_model_summary$r.squared, full_model_summary$adj.r.squared))
additional_stats_table
#Plots the Residual vs. Fitted Values
ggplot(mapping = aes(full$fitted, full$residuals)) +
    theme_minimal() + geom_point()+
    labs(title="Full Model Residual vs. Fitted Values",
    x="Fitted Values", y="Residuals")

#Plots the QQ plot for the full model's residuals
ggplot(mapping = aes(sample = rstandard(full))) +
    theme_minimal() + stat_qq() +
    stat_qq_line(color = 'red')+
```

```
    labs(title = "QQ-Plot of Full Model Residuals ",
    x="Fitted", y="Residuals")
#Reduced model with only correlated variables included as a predictor variable
reduced <- lm(Units.Sold~Demand.Forecast+Inventory.Level, retail)
reduced_model_summary <- summary(reduced)

#Creates dataframe for important coefficients
reduced_summary_table <- data.frame(reduced_model_summary$coefficients)%>%
  as_tibble(rownames = "Variable")

reduced_summary_table <- as_tibble(reduced_summary_table)
reduced_summary_table
#This gives us the R-Squared and Adjusted R-Squared in a tibble of its own
reduced_additional_stats_table <- tibble(
  Statistic = c("R-Squared", "Adjusted R-Squared"),
  Estimate = c(reduced_model_summary$r.squared, reduced_model_summary$adj.r.squared))
reduced_additional_stats_table
#Plots the Residual vs. Fitted Values
ggplot(mapping = aes(reduced$fitted, reduced$residuals)) +
    theme_minimal() + geom_point()+
    labs(title="Reduced Model Residual vs. Fitted Values",
    x="Fitted Values", y="Residuals")

#Plots the QQ plot for the reduced model's residuals
ggplot(mapping = aes(sample = rstandard(reduced))) +
    theme_minimal() + stat_qq() +
    stat_qq_line(color = 'red')+
    labs(title = "QQ-Plot of Reduced Model Residuals ",
    x="Fitted", y="Residuals")
#Null model with only the intercept included
null <- lm(Units.Sold~1, retail)
null_model_summary <- summary(null)


#Creates dataframe for important coefficients
null_main_summary_table <- data.frame(null_model_summary$coefficients) %>%
  as_tibble(rownames = "Variable")

null_main_summary_table <- as_tibble(null_main_summary_table)
null_main_summary_table
#This gives us the R-Squared and Adjusted R-Squared in a tibble of its own
null_additional_stats_table <- tibble(
  Statistic = c("R-Squared", "Adjusted R-Squared"),
  Estimate = c(null_model_summary$r.squared, null_model_summary$adj.r.squared))
null_additional_stats_table
#kfold cross validation algorithm implementation
kfold <- function(fit, data, k = 5) {
  n <- nrow(data)
  set.seed(2024)
  data <- data[sample(1:n, n),]
  groups <- split(data, (1:n) %% k)
  MSE <- c()
  for (i in 1:k) {
```

```
    train_data <- do.call('rbind', groups[-i])
    fit <- update(fit, data = train_data)
    yhat <- predict(fit, groups[[i]], type = "response")
    MSE[i] <- mean((groups[[i]]$Units.Sold - yhat)^2) #We are calculating MSE as our metric so this is
  }
  return(mean(MSE))
}
kfold_list <- c(kfold(full, retail, k = 5), kfold(reduced, retail, k = 5),kfold(null, retail, k = 5)) #
names <- c("Full", "Reduced", "Null")
kfold_table <- tibble(Model = names, MSE = kfold_list) #Creation of a tibble to compare MSE's
kfold_table
#Categorizes our response variable for classification purposes
Low.Sales <- retail %>%
  filter(Units.Sold <= 165)%>%
  mutate(Sales.Volume = "Low.Sales")
Medium.Sales <- retail %>%
  filter(165 < Units.Sold & Units.Sold <= 330)%>%
  mutate(Sales.Volume = "Medium.Sales")
High.Sales <- retail %>%
  filter(Units.Sold > 330)%>%
  mutate(Sales.Volume = "High.Sales")
retail <- rbind(High.Sales, Medium.Sales, Low.Sales)
retail <- retail %>%
  mutate(Sales.Volume = factor(Sales.Volume, levels = c("Low.Sales", "Medium.Sales", "High.Sales"), ord
retail <- subset(retail, select= -Units.Sold) #Deletes the Units.Sold variable which has been transform
head(retail, 3)
retail_scaled <- scale(retail[1:6]) #Scale retail data for purposes of fair comparison in PCA

pca <- prcomp(retail[1:6], center = TRUE, scale. = TRUE) #Conduct PCA dimension reductionality

pca_table <- data.frame(pca$rotation) #Gives us the loading values of principal components (influence)
tibble(pca_table) #returns tibble of the dataframe
pca_summary <- summary(pca) #summarizes PCA, gives us information about variance composition of each
pca_summary_table <- data.frame(
  Standard_Deviation = pca_summary$sdev,
  Proportion_of_Variance = pca_summary$importance[2, ],
  Cumulative_Variance = pca_summary$importance[3, ]
)
pca_summary_table <- as_tibble(pca_summary_table, rownames = "Variable") #tibble of variance computatio
head(pca_summary_table, 3)
pca_scores <- data.frame(pca$x[,1:2]) #Stores only PC1 and PC2 for our purposes
pca_scores$Volume <- retail$Sales.Volume #Stores Volume as a classifying value
pca_scores <- as_tibble(pca_scores, rownames = "Variable") #Creates tibble of this dataframe
ggplot(pca_scores, aes(x = PC1, y = PC2)) + #plots the pca plot (no classification) of PC1 and PC2
  geom_point(alpha = 0.4) +
  theme_minimal() +
  scale_color_manual(values = c("pink"))+
  labs(title = "PCA Plot of Retail Data", x = "PC1", y = "PC2")
set.seed(2024)
train_index <- sample(1:nrow(pca_scores), size = 0.8 * nrow(pca_scores)) #Splits the data for KNN analy

train_data <- pca_scores[train_index, ] #Stores the training data
test_data <- pca_scores[-train_index, ] #Stores the testing data
```

```r
train_x <- train_data[, c("PC1", "PC2")]
train_y <- train_data$Volume #The "response" or output of this classification is the volume label
test_x <- test_data[, c("PC1", "PC2")]
test_y <- test_data$Volume  #The "response" or output of this classification is the volume label

k <- 5 #For 5 nearest neighbors
knn_pred <- knn(train = train_x, test = test_x, cl = train_y, k = k) #Knn function implemented with the
#Plots the knn for which we set up previously
ggplot(test_data, aes(x = PC1, y = PC2, color = Volume)) +
  geom_point(alpha = 0.7, size = 3) +
  theme_minimal() +
  labs(title = "KNN Classification Sales Volume Comparison on PCA Data",
       x = "PC1", y = "PC2") +
  scale_color_manual(values = c("red", "blue", "green"))
conf_matrix_table <- table(Predicted = knn_pred, Actual = test_y) #Creates confusion matrix manually, c
conf_matrix_df <- as.data.frame(conf_matrix_table)
conf_matrix_tibble <- conf_matrix_df %>% #Converts the dataframe into a tibble
  pivot_wider(names_from = Actual, values_from = Freq) %>%
  rename(Predicted = Predicted)
conf_matrix_tibble
accuracy <- sum(diag(conf_matrix_table))/sum(conf_matrix_table) #Computes the accuracy comparing actual
accuracy
```