Store Branches Sales Regression Analysis

Renad Gharz

19/06/2022

1. Overview of Data

The data sample consists of 4 continuous quantitative variables and 896 total observations. The purpose of this analysis is to build a regression model to predict the store's sales in USD using the store area, the items available, and the daily customer count (average) as predictor variables.

The **Store__Area** variable represents the physical area of the store. The values are originally given in square yards (yd^2), however, for better readability, it was converted to squared feet (ft^2) by multiplying the squared yards values by 9.

The **Items__Available** variable represents the number of different items available in the store.

The **Daily__Customer__Count** variable represents the average number of customers who visited the store in a month.

the **Stores__Sales** variable represents the sales made in a store, in USD currency.

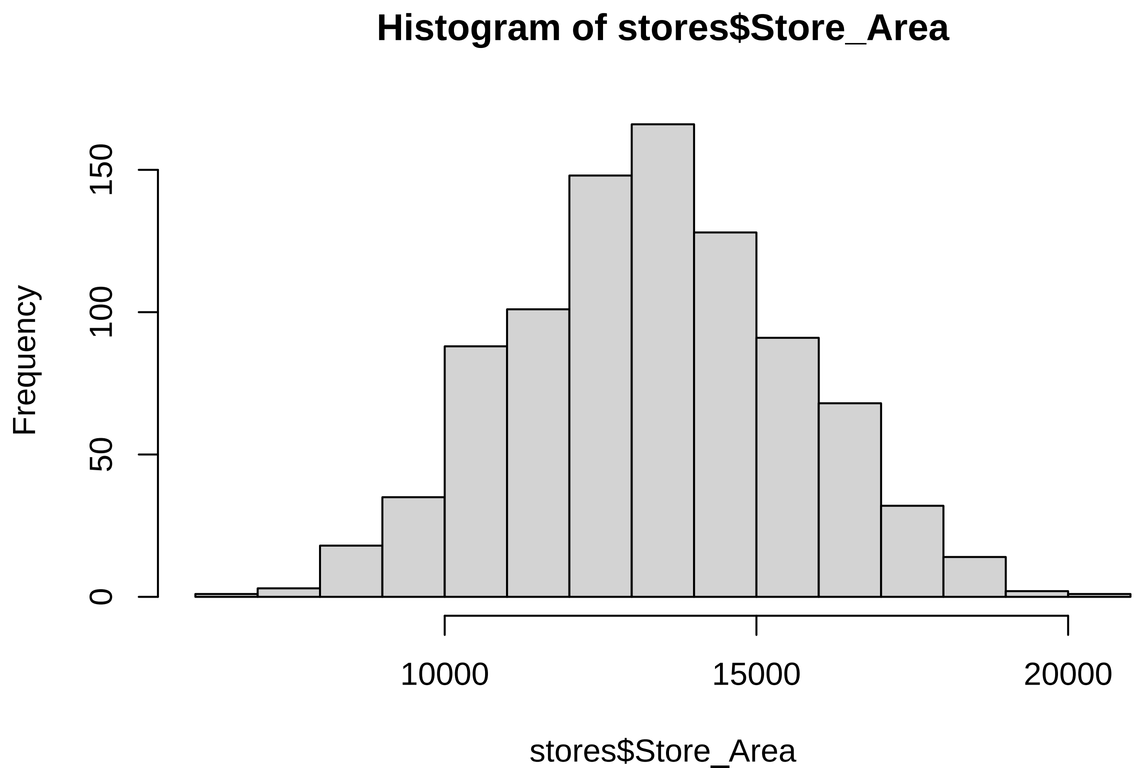
```
##   Store_Area Items_Available Daily_Customer_Count Store_Sales
## 1      14931            1961                  530       66490
## 2      13149            1752                  210       39820
## 3      12060            1609                  720       54010
## 4      13059            1748                  620       53730
## 5      15930            2111                  450       46620
## 6      12978            1733                  760       45260
```

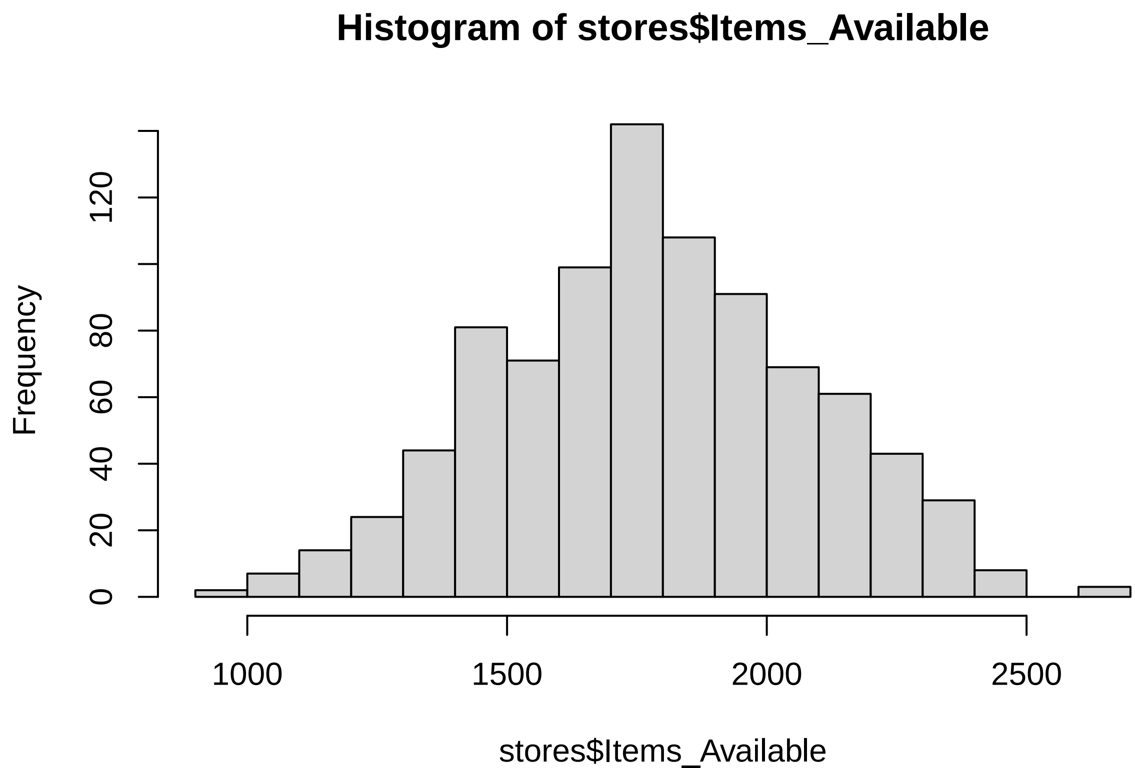
2. Exploratory Analysis

Central tendencies of the sample:

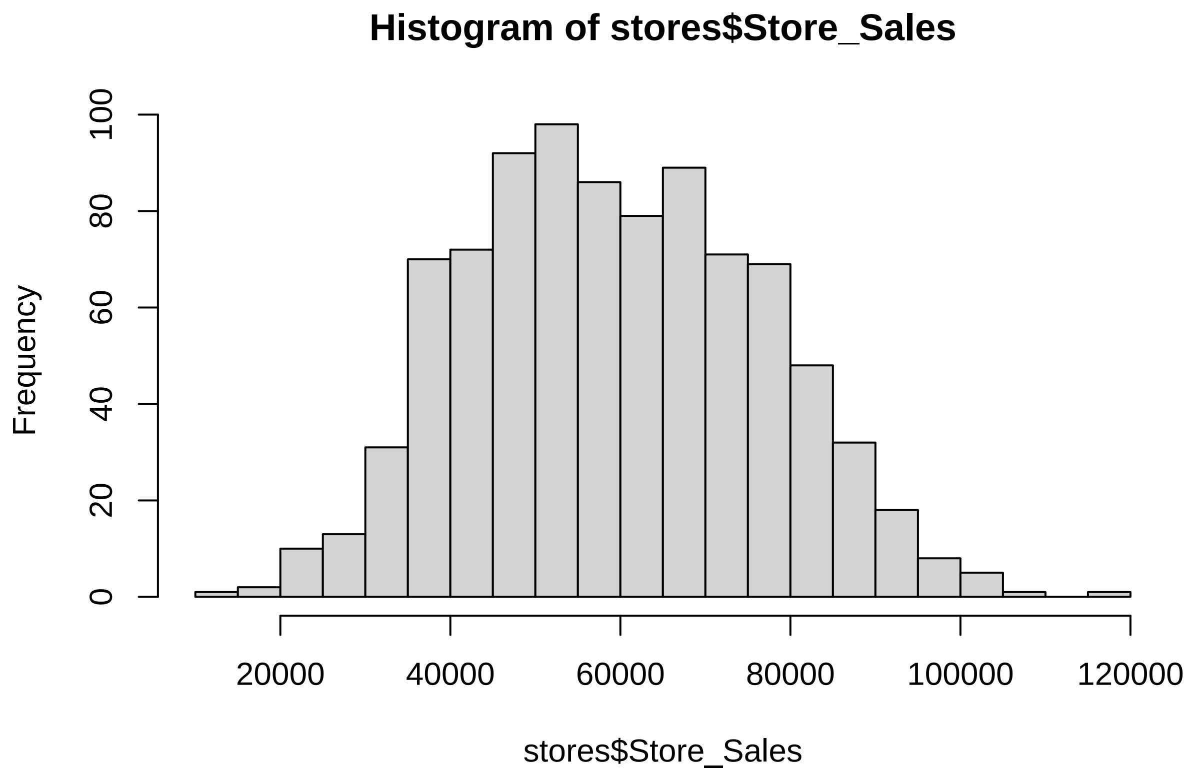
```
##    Store_Area    Items_Available Daily_Customer_Count  Store_Sales
##  Min.   : 6975   Min.   : 932    Min.   :  10.0       Min.   : 14920
##  1st Qu.:11851   1st Qu.:1576    1st Qu.: 600.0       1st Qu.: 46530
##  Median :13293   Median :1774    Median : 780.0       Median : 58605
##  Mean   :13369   Mean   :1782    Mean   : 786.4       Mean   : 59351
##  3rd Qu.:14882   3rd Qu.:1983    3rd Qu.: 970.0       3rd Qu.: 71873
##  Max.   :20061   Max.   :2667    Max.   :1560.0       Max.   :116320
```

From the central tendencies table above, we can see that the means medians of all 4 variables are relatively equal indicating that the sample data is relatively symmetrical. We can further confirm this by looking at the shape of each variable's histogram which support the relative symmetry claim.


Histogram of stores$Store_Area
Frequency
0
50
100
150
10000
15000
20000
stores$Store_Area


Histogram of stores$Items_Available
Frequency
0
20
40
60
80
120
1000
1500
2000
2500
stores$Items_Available

Histogram of stores$Daily_Customer_Count
Frequency
0
20
40
60
80
100
0
500
1000
1500
stores$Daily_Customer_Count



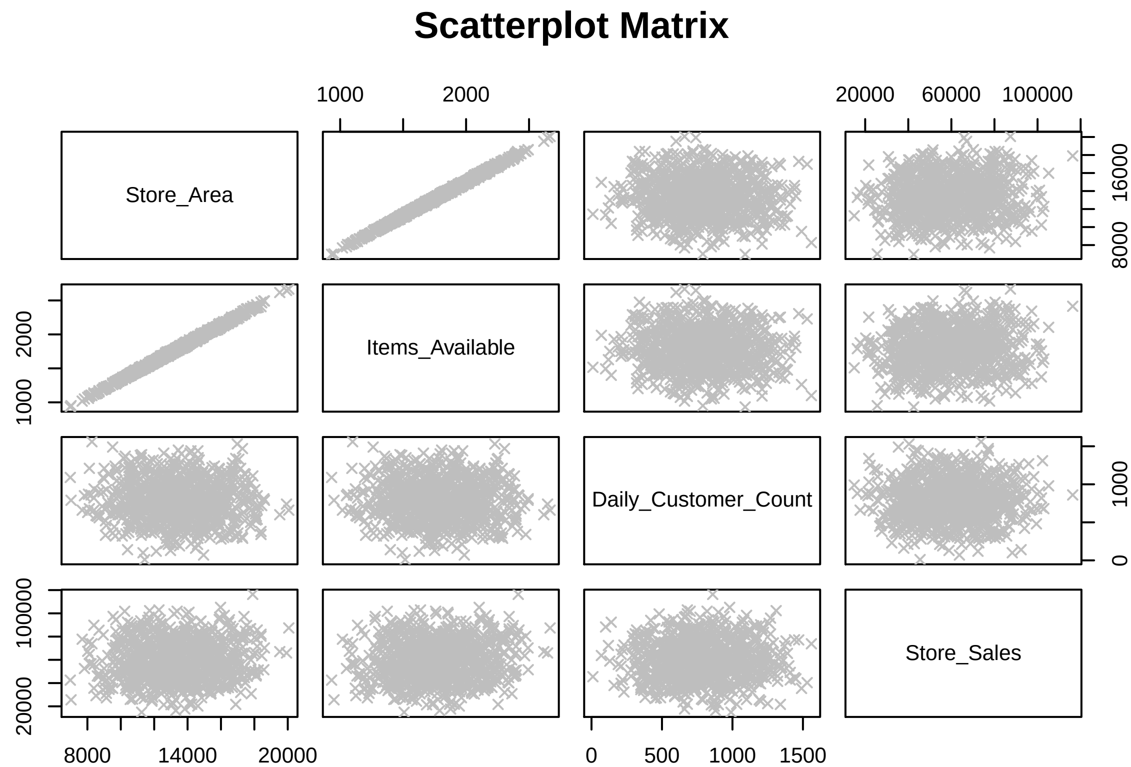
As shown in the histograms, the data is still symmetrically distributed. Although we can see a few outlier points on some of the histograms, they are negligible and do not dramatically skew or influence th shape of the data distribution.

Descriptive statistics of the sample:

```
## Warning in describeBy(stores): no grouping variable requested
```

```
##                      vars   n     mean       sd  median  trimmed      mad   min
## Store_Area              1 896 13368.69  2252.13 13293.0 13356.36  2241.69  6975
## Items_Available         2 896  1782.04   299.87  1773.5  1780.30   300.23   932
## Daily_Customer_Count    3 896   786.35   265.39   780.0   784.25   266.87    10
## Store_Sales             4 896 59351.31 17190.74 58605.0 59056.66 18636.28 14920
##                         max  range skew kurtosis     se
## Store_Area            20061  13086 0.03    -0.29  75.24
## Items_Available        2667   1735 0.03    -0.29  10.02
## Daily_Customer_Count   1560   1550 0.07    -0.27   8.87
## Store_Sales          116320 101400 0.15    -0.47 574.30
```

The descriptive statistics also support this as the skew factors are fairly low for 3 of the 4 variables, with the store sales variable being the exception having a 0.15 skew, which is still an acceptable level.




```
##                         Store_Area Items_Available Daily_Customer_Count
## Store_Area               1.0000000      0.99889075         -0.041423095
## Items_Available          0.9988908      1.00000000         -0.040978117
## Daily_Customer_Count    -0.0414231     -0.04097812          1.000000000
## Store_Sales              0.0974738      0.09884943          0.008628708
##                         Store_Sales
## Store_Area              0.097473795
## Items_Available         0.098849435
## Daily_Customer_Count    0.008628708
## Store_Sales             1.000000000
```

The scatterplot matrix shows that apart from one relation (Store_Area-Items_Available), the remaining predictor variables have little to no linear relationship with the response variable, which could result in a less accurate regression model (due to collinearity). This is further backed up by the correlation table which indicates low correlation between most of the variables.

3. Model 1 - Raw Data

In order to make sure that we end up with the most accurate model with the highest prediction power, a few different models will be built with various adjustments. Model 1 is the simplest model and will use the raw data without any transformations (except for the square yards to square feet conversions for Store_Area).

```
regression_model_1 <-
  lm(data=stores,
     formula=
       Store_Sales~Store_Area+Items_Available+Daily_Customer_Count)
```

```r
model_1_summary <- summary(regression_model_1)
model_1_summary
```

```
##
## Call:
## lm(formula = Store_Sales ~ Store_Area + Items_Available + Daily_Customer_Count,
##     data = stores)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43689 -12834   -567  12882  52405
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          48567.7788  3908.8280  12.425   <2e-16 ***
## Store_Area              -4.3373     5.3989  -0.803    0.422
## Items_Available         38.2342    40.5469   0.943    0.346
## Daily_Customer_Count     0.8046     2.1592   0.373    0.710
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17130 on 892 degrees of freedom
## Multiple R-squared:  0.01065,    Adjusted R-squared:  0.007321
## F-statistic:   3.2 on 3 and 892 DF,  p-value: 0.02276
```

```r
model_1_summary$coefficients
```

```
##                         Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept)          48567.7788448 3908.828046 12.4251510 8.314683e-33
## Store_Area              -4.3372855    5.398934 -0.8033596 4.219809e-01
## Items_Available         38.2341546   40.546944  0.9429602 3.459566e-01
## Daily_Customer_Count     0.8046133    2.159199  0.3726443 7.095017e-01
```

```r
model_1_summary$adj.r.squared
```

```
## [1] 0.007320654
```

Since this is a multiple regression (more than 1 predictor variable), it is better to use the adjusted R-squared than the multiple R-squared. As we add more predictor variables to models, the R-squared will always go up, thus we need to compensate for this to make sure our model is not inaccurate, thus we would use the adjusted R-squared which accounts for every additional predictor variable added, giving us a more accurate estimation of the model's prediction accuracy.

The R-squared measures how much variance the model accounts for, thus the higher the value, the better the model will be at predicting the response variable. The adjusted R-squared is 0.73% which is extremely low, indicating that the model has almost no prediction power to accurately regress the response variable.

As suspected previously, collinearity seems to be a factor here as the **Store_Area** and the **Items_Available** predictors are almost perfectly correlated. This pairwise collinearity is affecting how we interpret the coefficients, which is why the Store_Area coefficient is negative which does not make sense from a domain perspective.

We can verify this property by looking at the Variance Inflation Factor of model's predictors.

```r
vif(regression_model_1)
```

```
##           Store_Area      Items_Available Daily_Customer_Count
##           451.054471           451.037905             1.001791
```

As shown, the 2 correlated predictors demonstrate a very high VIF (greater than 10), which indicates multicollinearity in the model. In order to make the model more parsimonuous, we can run new models by dropping one of those 2 collinear variables to improve the model.

4. Model 2 - Adjusting for Collinearity

In Model 2, we will compare 2 submodels in which one of the 2 collinear variables will be dropped in each model to see which one demonstrates better prediction power.

4.1 Model 2a - Dropping Store_Area

We will first drop the **Store_Area** variable.

```
stores2a <- stores %>% select(-Store_Area)
head(stores2a)
```

```
##   Items_Available Daily_Customer_Count Store_Sales
## 1            1961                  530       66490
## 2            1752                  210       39820
## 3            1609                  720       54010
## 4            1748                  620       53730
## 5            2111                  450       46620
## 6            1733                  760       45260
```

```
regression_model_2a <- lm(data=stores2a,
                         formula=
                           Store_Sales~Items_Available+Daily_Customer_Count)

model_2a_summary <- summary(regression_model_2a)
model_2a_summary
```

```
##
## Call:
## lm(formula = Store_Sales ~ Items_Available + Daily_Customer_Count,
##     data = stores2a)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43104 -12913   -660  12686  53308
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          4.855e+04  3.908e+03  12.424  < 2e-16 ***
## Items_Available      5.697e+00  1.910e+00   2.982  0.00294 **
## Daily_Customer_Count 8.227e-01  2.159e+00   0.381  0.70321
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17120 on 893 degrees of freedom
## Multiple R-squared:  0.009932,   Adjusted R-squared:  0.007715
## F-statistic: 4.479 on 2 and 893 DF,  p-value: 0.0116
```

```
model_2a_summary$coefficients
```

```
##                          Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept)          48552.888556 3908.007928 12.4239483 8.370588e-33
```

```
## Items_Available          5.696570    1.910426   2.9818316 2.943120e-03
## Daily_Customer_Count     0.822695    2.158653   0.3811149 7.032086e-01
```

```
model_2a_summary$adj.r.squared
```

```
## [1] 0.007714851
```

In this new model we can see that the adjusted R-squared has minutely improved over the model that used the raw data to 0.77%, however this model is still not good enough.

4.2 Model 2b - Dropping Items_Available

We can now move on to dropping the **Items_Available** variable.

```
stores2b <- stores %>% select(-Items_Available)
head(stores2b)
```

```
##   Store_Area Daily_Customer_Count Store_Sales
## 1      14931                  530       66490
## 2      13149                  210       39820
## 3      12060                  720       54010
## 4      13059                  620       53730
## 5      15930                  450       46620
## 6      12978                  760       45260
```

```
regression_model_2b <- lm(data=stores2b,
                          formula=
                            Store_Sales~Store_Area+Daily_Customer_Count)

model_2b_summary <- summary(regression_model_2b)
model_2b_summary
```

```
##
## Call:
## lm(formula = Store_Sales ~ Store_Area + Daily_Customer_Count,
##     data = stores2b)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43014 -12916   -683  12654  53518
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          4.870e+04  3.906e+03  12.470  < 2e-16 ***
## Store_Area           7.480e-01  2.544e-01   2.940  0.00336 **
## Daily_Customer_Count 8.219e-01  2.159e+00   0.381  0.70353
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17130 on 893 degrees of freedom
## Multiple R-squared:  0.009662,   Adjusted R-squared:  0.007444
## F-statistic: 4.356 on 2 and 893 DF,  p-value: 0.0131
```

```
model_2b_summary$coefficients
```

```
##                          Estimate   Std. Error    t value     Pr(>|t|)
## (Intercept)          4.870473e+04 3905.8866344 12.4695703 5.143580e-33
## Store_Area           7.480384e-01    0.2544131  2.9402508 3.364059e-03
```

```
## Daily_Customer_Count 8.218818e-01    2.1589878  0.3806792 7.035318e-01
```

```r
model_2b_summary$adj.r.squared
```

```
## [1] 0.007443851
```

Although this model has an improved adjusted R-squared, it is negligible compared to the original model and not as good as the model where we dropped **Store__Area**. Because the model is still not satisfactory and we know that the relationship between the predictors and the response variables is very weak. We can perform some transformations on the data to try to linearize the non-linear relationships.

5. Model 3 - Log-Transformed Data

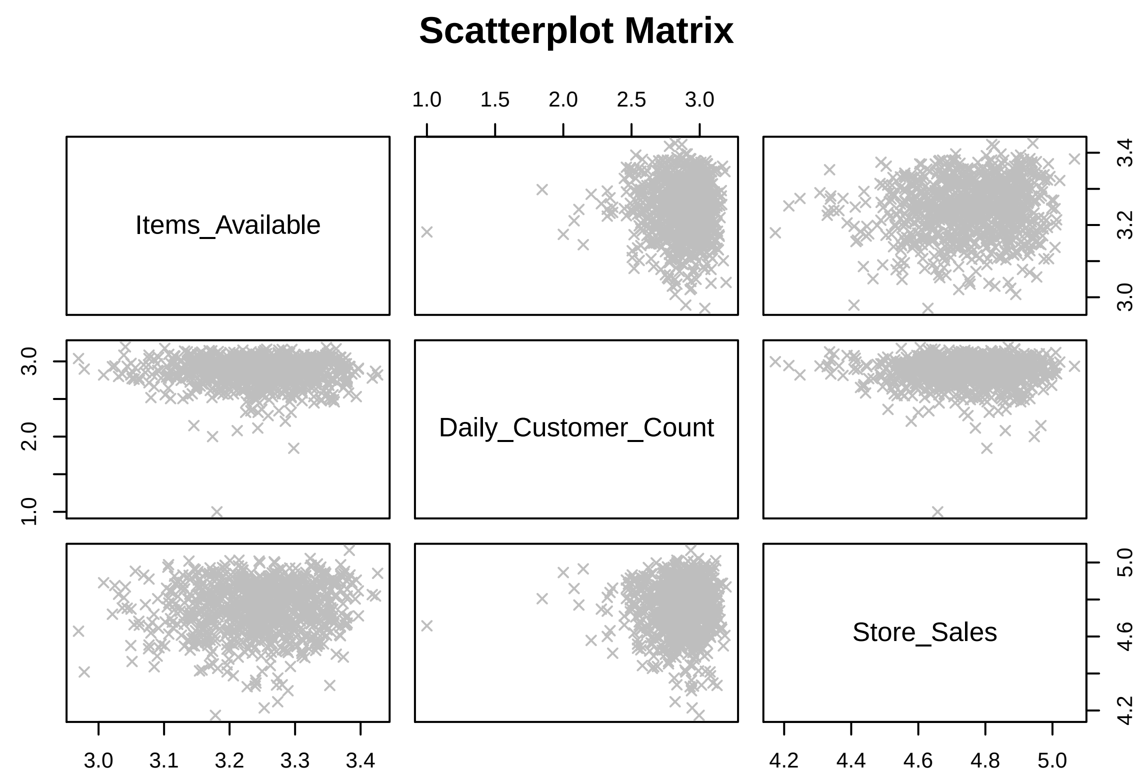
We can start by transforming the data using the most basic log transformation with a base of 10 ($\log_{10} x$).

```r
stores3 <- log10(stores2a)
head(stores3)
```

```
##   Items_Available Daily_Customer_Count Store_Sales
## 1        3.292478             2.724276    4.822756
## 2        3.243534             2.322219    4.600101
## 3        3.206556             2.857332    4.732474
## 4        3.242541             2.792392    4.730217
## 5        3.324488             2.653213    4.668572
## 6        3.238799             2.880814    4.655715
```

We can check if the transformation has linearized the data using a scatterplot matrix.

```r
plot(stores3,
     main="Scatterplot Matrix",
     col="grey",
     pch=4)
```




```r
cor(stores3)
```

```
##                      Items_Available Daily_Customer_Count  Store_Sales
## Items_Available           1.00000000          -0.033012460  0.098658128
## Daily_Customer_Count     -0.03301246           1.000000000 -0.001322143
## Store_Sales               0.09865813          -0.001322143  1.000000000
```

The scatterplot matrix indicates that even with the transformation, the data still does not shown a sign of linear relationship between the predictors and the response variable. However, we can still run a model with the log transformed data to check whether there has been an improvement in the accuracy.

```r
regression_model_3 <- lm(data=stores3,
                         formula=
                           Store_Sales~Items_Available+Daily_Customer_Count)

model_3_summary <- summary(regression_model_3)
model_3_summary
```

```
##
## Call:
## lm(formula = Store_Sales ~ Items_Available + Daily_Customer_Count,
##     data = stores3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56814 -0.08495  0.01431  0.10485  0.28741
##
## Coefficients:
```

```
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           4.171008   0.210073  19.855  < 2e-16 ***
## Items_Available       0.178254   0.060160   2.963  0.00313 **
## Daily_Customer_Count  0.001448   0.024905   0.058  0.95366
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1356 on 893 degrees of freedom
## Multiple R-squared:  0.009737,   Adjusted R-squared:  0.007519
## F-statistic:  4.39 on 2 and 893 DF,  p-value: 0.01266
```

```r
model_3_summary$coefficients
```

```
##                         Estimate Std. Error     t value     Pr(>|t|)
## (Intercept)          4.171007993 0.21007333 19.85500981 5.987623e-73
## Items_Available      0.178253670 0.06016037  2.96297483 3.127660e-03
## Daily_Customer_Count 0.001447822 0.02490525  0.05813321 9.536555e-01
```

```r
model_3_summary$adj.r.squared
```

```
## [1] 0.00751934
```

The log-transformed data demonstrates a slightly weaker R-squared than Model 2a where we dropped the **Stores__Area** variable. The reason for this is likely due to the response variable being disproportionately larger than the predictor variables which likely causes the scatterplots to be clustered together in an odd-looking vertical stack.

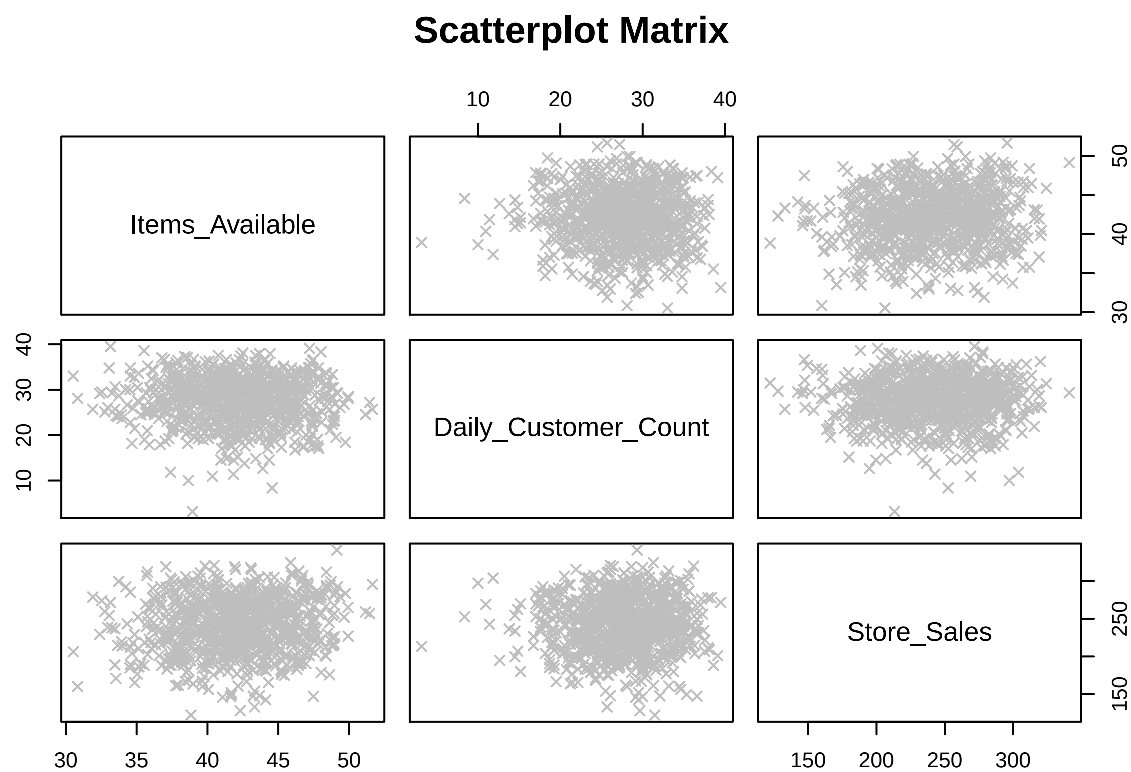
6. Model 4 - Square-Root Transformation

To address this disproportion between the variables, we can square root the variables to try linearizing the relationships.

```r
stores4 <- sqrt(stores2a)
#stores4$Store_Sales <- sqrt(stores4$Store_Sales)
head(stores4)
```

```
##   Items_Available Daily_Customer_Count Store_Sales
## 1        44.28318             23.02173    257.8565
## 2        41.85690             14.49138    199.5495
## 3        40.11234             26.83282    232.4005
## 4        41.80909             24.89980    231.7973
## 5        45.94562             21.21320    215.9167
## 6        41.62932             27.56810    212.7440
```

```r
plot(stores4,
     main="Scatterplot Matrix",
     col="grey",
     pch=4)
```




```r
cor(stores4)
```

```
##                      Items_Available Daily_Customer_Count Store_Sales
## Items_Available           1.00000000          -0.040119192 0.099243048
## Daily_Customer_Count     -0.04011919           1.000000000 0.004046966
## Store_Sales               0.09924305           0.004046966 1.000000000
```

```r
regression_model_4 <- lm(data=stores4,
                         formula=
                           Store_Sales~Items_Available+Daily_Customer_Count)

model_4_summary <- summary(regression_model_4)
model_4_summary
```

```
##
## Call:
## lm(formula = Store_Sales ~ Items_Available + Daily_Customer_Count,
##     data = stores4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -115.780  -25.352    1.273   27.849   92.939
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          197.21086   15.86755  12.429  < 2e-16 ***
## Items_Available        1.00147    0.33519   2.988  0.00289 **
```

```
## Daily_Customer_Count   0.05805    0.24056   0.241   0.80937
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.95 on 893 degrees of freedom
## Multiple R-squared:  0.009914,   Adjusted R-squared:  0.007696
## F-statistic: 4.471 on 2 and 893 DF,  p-value: 0.0117
```

```
model_4_summary$coefficients
```

```
##                          Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)          197.21085792 15.8675471 12.4285661 7.968446e-33
## Items_Available        1.00146800  0.3351877  2.9877828 2.886980e-03
## Daily_Customer_Count   0.05804875  0.2405573  0.2413095 8.093707e-01
```

```
model_4_summary$adj.r.squared
```

```
## [1] 0.007696305
```

The square-root transformation shows that there definitely is a slight improvement in the linear relationships between the predictors and the response variable. However, even with that the model's prediction power has not really improved as the R-squared is still ~0.77%.

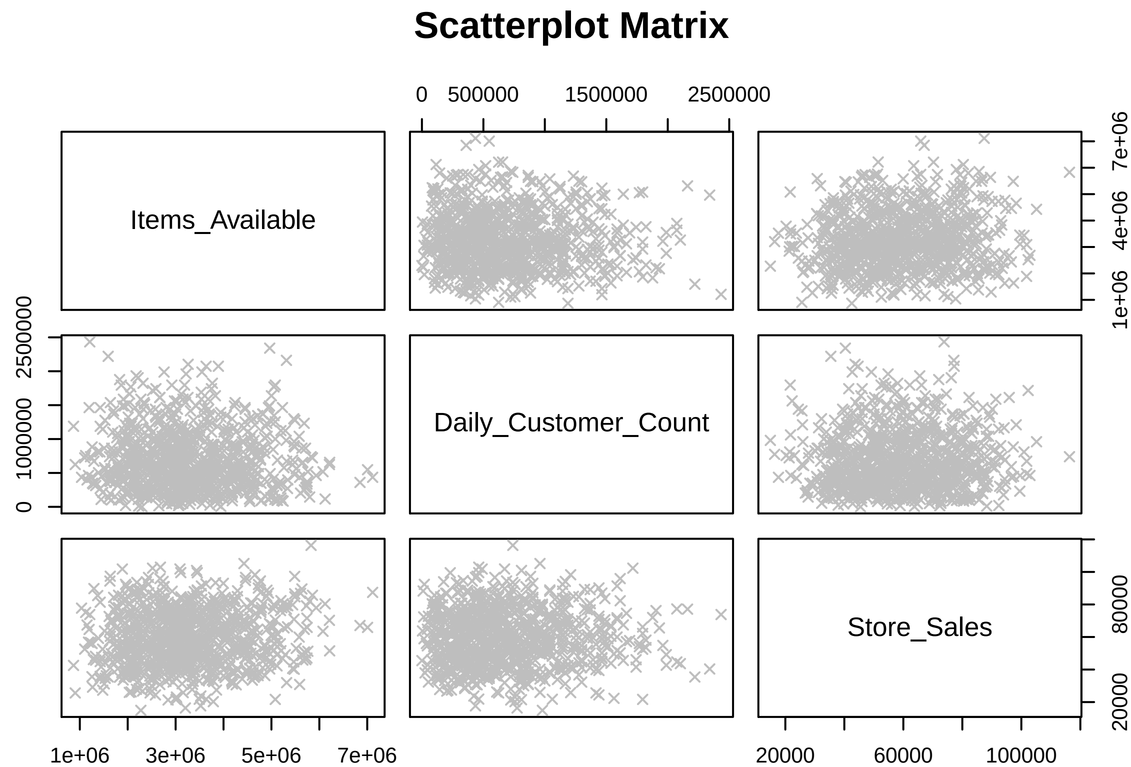
6. Model 5 - Power Transformation

Another transformation we can try is the power transformation to try and bring up the predictor variables into a relatively similar range as the response variable.

```
stores5 <- stores2a
stores5$Items_Available <- stores5$Items_Available^2
stores5$Daily_Customer_Count <- stores5$Daily_Customer_Count^2
#stores4$Store_Sales <- sqrt(stores4$Store_Sales)
head(stores5)
```

```
##   Items_Available Daily_Customer_Count Store_Sales
## 1         3845521               280900       66490
## 2         3069504                44100       39820
## 3         2588881               518400       54010
## 4         3055504               384400       53730
## 5         4456321               202500       46620
## 6         3003289               577600       45260
```

```
plot(stores5,
     main="Scatterplot Matrix",
     col="grey",
     pch=4)
```




```r
cor(stores5)
```

```
##                      Items_Available Daily_Customer_Count Store_Sales
## Items_Available           1.00000000         -0.038112995 0.102268756
## Daily_Customer_Count     -0.03811299          1.000000000 0.009870064
## Store_Sales               0.10226876          0.009870064 1.000000000
```

```r
regression_model_5 <- lm(data=stores5,
                         formula=
                           Store_Sales~Items_Available+Daily_Customer_Count)

model_5_summary <- summary(regression_model_5)
model_5_summary
```

```
##
## Call:
## lm(formula = Store_Sales ~ Items_Available + Daily_Customer_Count,
##     data = stores5)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -42966 -12934   -600  12787  52742
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.362e+04  2.070e+03  25.903  < 2e-16 ***
## Items_Available      1.639e-03  5.310e-04   3.086  0.00209 **
```

```
## Daily_Customer_Count 5.479e-04  1.324e-03   0.414  0.67902
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17120 on 893 degrees of freedom
## Multiple R-squared:  0.01065,    Adjusted R-squared:  0.008433
## F-statistic: 4.806 on 2 and 893 DF,  p-value: 0.008395
```

```r
model_5_summary$coefficients
```

```
##                          Estimate   Std. Error    t value      Pr(>|t|)
## (Intercept)          5.362251e+04 2.070129e+03 25.9029846 8.757267e-111
## Items_Available      1.638809e-03 5.310369e-04  3.0860549  2.090860e-03
## Daily_Customer_Count 5.478509e-04 1.323520e-03  0.4139346  6.790214e-01
```

```r
model_5_summary$adj.r.squared
```

```
## [1] 0.008432935
```

We can see that the power transformation on the predictor variables has slightly improved the linear relationship between the predictors and response variable. Although, the R-squared has improved to 0.84%, it still indicates the model is very weak, despite the transformations performed on the data.

7. Model 6 - Removing the Weak Predictor

One consistent factor across the various models is that the **Daily_Customer_Count** has been a very weak coefficient, thus in a last attempt at building an optimal model, we can drop that variable from our model to try to make it as parsimonious as possible.

From a domain perspective, this makes sense as you could a store could have fewer customers who buy more items during one visit, instead of a many customers visiting and only buying one or two items.

```r
stores6 <- stores2a %>% select(-Daily_Customer_Count)
head(stores6)
```

```
##   Items_Available Store_Sales
## 1            1961       66490
## 2            1752       39820
## 3            1609       54010
## 4            1748       53730
## 5            2111       46620
## 6            1733       45260
```

```r
regression_model_6 <- lm(data=stores6,
                         formula=
                           Store_Sales~Items_Available)

model_6_summary <- summary(regression_model_6)
model_6_summary
```

```
##
## Call:
## lm(formula = Store_Sales ~ Items_Available, data = stores6)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43026 -12902   -627  12670  53388
```

```
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    49252.984   3447.710   14.29  < 2e-16 ***
## Items_Available     5.667      1.908    2.97  0.00306 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17120 on 894 degrees of freedom
## Multiple R-squared:  0.009771,   Adjusted R-squared:  0.008664
## F-statistic: 8.822 on 1 and 894 DF,  p-value: 0.003056
```

```
model_6_summary$coefficients
```

```
##                     Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept)     49252.983647 3447.710322 14.285708 7.476366e-42
## Items_Available     5.666734    1.907909  2.970128 3.056357e-03
```

```
model_6_summary$adj.r.squared
```

```
## [1] 0.008663572
```

```
model_6_summary$r.squared
```

```
## [1] 0.009771211
```

In this case, because we are down to a single predictor, we can use the multiple R-squared as a measure of the strength of Model 6 because we do not need to compensate or adjust for having multiple predictors (since there is only one). Thus, although the model still has a relatively weak accuracy, it is much better than any of the previous models.

8. Conclusion & Takeaways

To conclude, although the final model selected (Model 6) still has relatively weak accuracy (0.98%), we managed to significantly improve it by 34.25% from the first model looked at by dropping weak predictors and transforming variables to end up with a more parsimonious model.

Despite the different transformations applied to try to linearize the sample's variable relationships, the models' power were always very weak. One of the possible explanations for this is that the predictors may not be that great of predictors for the average store sales in a month.

There are other variables that might have been more suited such as location of the store (such as downtown, suburb, remote town, etc.) instead of the daily customer count. The daily customer count might have added too much variance to the model as this variable could fluctuate tremendously by location and even by season (which were not provided in the sample). The representation of these 2 categorical variables (location and season) could have helped in improving the accuracy of the models since from a domain perspective, these 2 factors can be greatly important in determining a store branch's profits for a given month.