Orange vs R comparative

Create an Orange file and an R file providing analysis on this dataset. This may require research on your part to learn how. It should include the following:

1. Regression Analysis for a predictor
2. Model building (eliminating any variables that are un-necessary, etc)
3. A testing set and training set
4. Analysis of the model

Compare the analysis from using the two different software. In particular, describe:

A. Which do you believe is easier?
B. Which tells you more?
C. Which would be better for convincing others of your conclusions?

Include files for both methods, and a write-up of your work. I am the audience of your work for this project.

R makes use of some very helpful packages. It makes statistical work much easier and reduces the amount of code that is required to get the same output.

Download packages
library(Metrics)
library(janitor)
library(tidyverse)
library(caret)
library(Rcmdr)
library(leaps)

Cleaning the data

When first loading in the data, it is observed that there are many decimal places however it makes sense to round to a specific decimal place for different values.

Promotional spend, competitor spend and sales should all be rounded to 2 decimal places
Average temperature needs to be rounded to 1 decimal place
Online traffic and foot traffic needs to be rounded to the nearest value

However, you must first check for null and duplicated values in your columns to make sure that you are able to continue the regression without further issues

This can be achieved by the following code

R Regression – Comparison.R | data | complex_retail_sales_dataset

Source on Save

```r
#Save the data, 'complex_retail_sales_dataset' as data
data <- complex_retail_sales_dataset

#Check for null values
print(colSums(is.na(data)))

#Check for duplicated values
print(sum(duplicated(data)))

print(colnames(data))

#Create a list of columns that are going to be rounded to 2 decimal places
col_2_round <- c('PromotionSpend', 'CompetitorSpend', 'Sales')
col_1_round <- c('AvgTemperature')
col_0_round <- c('FootTraffic', 'OnlineTraffic')

#Perform the following rounding to each columns that were listed above
data[, col_2_round] <- round(data[, col_2_round], digits = 2)
data[, col_1_round] <- round(data[, col_1_round], digits = 1)
data[, col_0_round] <- round(data[, col_0_round], digits = 0)
```

Regression Analysis

Correlation Matrix

Activate R Commander

Statistics -> summaries -> correlation matrix

It can also be achieved with the code below

```r
#Create correlation matrix (using r commander)
#Exclude sales which is the output
cor(complex_retail_sales_dataset[,c("AvgTemperature","CompetitorSpend",
                                    "FootTraffic","Holiday","OnlineTraffic",
                                    "PromotionSpend","Weekday")],
                                    use="complete")
```

The following is the output

```
                AvgTemperature CompetitorSpend FootTraffic     Holiday
AvgTemperature      1.00000000     -0.05364001  0.07102930 -0.01033932
CompetitorSpend    -0.05364001      1.00000000 -0.02752884 -0.11981482
FootTraffic         0.07102930     -0.02752884  1.00000000  0.12977910
Holiday            -0.01033932     -0.11981482  0.12977910  1.00000000
OnlineTraffic      -0.11285322     -0.01245323 -0.08116507  0.01501078
PromotionSpend     -0.04401577     -0.06417293  0.17033554  0.06489184
Weekday            -0.02283833      0.03805129 -0.01654827  0.02656752
                OnlineTraffic PromotionSpend     Weekday
AvgTemperature     -0.11285322    -0.04401577 -0.02283833
CompetitorSpend    -0.01245323    -0.06417293  0.03805129
FootTraffic        -0.08116507     0.17033554 -0.01654827
Holiday             0.01501078     0.06489184  0.02656752
OnlineTraffic       1.00000000     0.11915058 -0.03727461
PromotionSpend      0.11915058     1.00000000 -0.03399605
Weekday            -0.03727461    -0.03399605  1.00000000
```

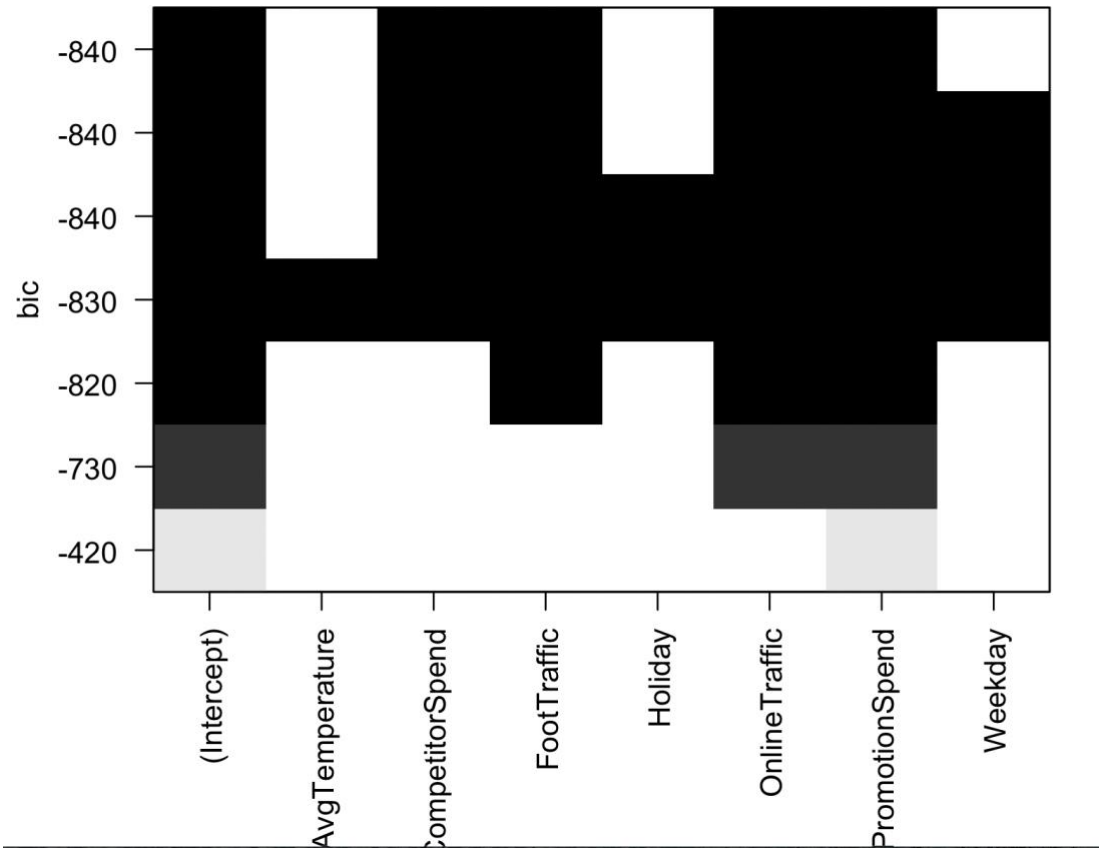
After looking at the correlation matrix between all variables there are no values that show an extremely high correlation which allows us to include all of the variables in the model.

The next step is to look at the BIC graph with the variables to get a good idea of which variables are going to be used to create the greatest model.

Active r commander
Models -> subset model selection

This can be achieved by using the following code

```
Rcmdr>  plot(regsubsets(Sales ~ AvgTemperature + CompetitorSpend + FootTraffic +
Holiday + OnlineTraffic + PromotionSpend + Weekday,
data=complex_retail_sales_dataset, nbest=1, nvmax=8), scale='bic')
```



From the graph above, it seems that removing average temperature, holiday and weekend from the model is going to promote the most effective response.
To ensure this, perform a linear regression model with these variables and look at their p values.

Active R Commander

Select dataset as complex_retail_sales_dataset

Go to statistics -> fit models -> linear models -> select all variables but sales for explanatory ->pick sales as the response variable

```
Call:
lm(formula = Sales ~ AvgTemperature + CompetitorSpend + FootTraffic +
    Holiday + OnlineTraffic + PromotionSpend + Weekday, data = complex_retail_sales_dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-117.384  -32.445   -1.387   32.221  139.724

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     223.85118   30.25256   7.399 1.46e-12 ***
AvgTemperature   -0.14285    0.60778  -0.235    0.814
CompetitorSpend  -0.16768    0.02952  -5.680 3.25e-08 ***
FootTraffic       0.31871    0.02998  10.631  < 2e-16 ***
Holiday          -5.86781    7.21036  -0.814    0.416
OnlineTraffic     0.46919    0.01571  29.871  < 2e-16 ***
PromotionSpend    1.52100    0.02688  56.594  < 2e-16 ***
Weekday          -2.28661    1.49715  -1.527    0.128
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 51.13 on 292 degrees of freedom
Multiple R-squared:  0.9459,    Adjusted R-squared:  0.9446
F-statistic: 729.4 on 7 and 292 DF,  p-value: < 2.2e-16
```

From the model above alongside the BIC we have enough reason to remove AvgTemperature, Holiday & Weekday from the model. This will result in the following changes to the model below.

```
Call:
lm(formula = Sales ~ CompetitorSpend + FootTraffic + OnlineTraffic +
    PromotionSpend, data = complex_retail_sales_dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-117.282  -33.947   -2.026   34.907  134.653

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     210.58841   26.13266   8.058 1.94e-14 ***
CompetitorSpend  -0.16613    0.02926  -5.678 3.27e-08 ***
FootTraffic       0.31591    0.02969  10.642  < 2e-16 ***
OnlineTraffic     0.47014    0.01562  30.107  < 2e-16 ***
PromotionSpend    1.52156    0.02683  56.716  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 51.14 on 295 degrees of freedom
Multiple R-squared:  0.9453,    Adjusted R-squared:  0.9446
F-statistic:  1275 on 4 and 295 DF,  p-value: < 2.2e-16
```

Our model has now been selected, this can be represented in the equation below.

210.58841 – 0.16613*CompetitorSpend + 0.31591*FootTraffic + 0.47014*OnlineTraffic + 1.52156*PromotionSpend

Promotion spend has the highest correlation with an increasing value which makes sense as the more money that is spent on marketing represents more sales as your audience grows. It also makes sense that competitor spend is a negative because as this value decreases you are going to gain a greater market share and get more sales.

Now that the model has been finalised with feature selection complete, the next step is to test the effectiveness of the model. There are many different ways of choosing between your test and your training data. The holdout method which usually involves 70-80% of the data being used for testing while the 20-30% after is going to be used for testing. There is cross validation which splits the dataset into k subsets of equal size which is what is going to take place in this model. This is going to be done by splitting the data into 50% test and 50% training using a random sample. This can be done with the code below

```
41  #Create a test and training split to test the effectivenss of the model
42  rand_sample <- createDataPartition(data$Sales, p = 0.50, list = FALSE)
```

This step creates a data split with a random 50/50 between test and training. Now it is time to save the data as different variables that will be used when creating the model with just the test and then comparing to the test data.

```
44  #Create the training and testing sets
45  train_set <- data[rand_sample, ]
46  test_set <- data[-rand_sample, ]
```

Now that those variables have been saved, create the model with the variables that were used before and the train data.

```
48  #make the model
49  model <- lm(Sales ~ PromotionSpend + CompetitorSpend + FootTraffic + OnlineTraffic, data = train_set)
```

With the model, make predictions using the test set of the data split. Using the code below generate predictions for the model.

```
#Generate the predictions
predictions <- predict(model, newdata = test_set)
```

Analysis of the predictions

There are many ways of analysing the results of your predictions. Looking at the R^2 and the Mean Absolute Error which is going to take the difference of predicted values from the real values and taking the squared positive value of that.

```
#find the R^2 value
R2(predictions, test_set$Sales)

#find the MAE value
MAE(predictions, test_set$Sales)
```

This returned values of
R^2 – 0.9510692
MAE – 39.60793

An R^2 of 0.9510692 suggests a very high correlation between the change in the variables in the model and the change in the value of sales.

Meanwhile the MAE represents the change in values from real values versus predicted values, this was found with a value of 39.60793. To determine how good of a result this is, lets do a comparison to a baseline test looking at the mean of sales and the median of sales. To compare that MAE with the MAE that was found using the model. Use the code below

```
# Calculate the mean and median of Sales in the training set
mean_sales <- mean(train_set$Sales)
median_sales <- median(train_set$Sales)

# Create vectors of mean and median predictions for the test set
mean_predictions <- rep(mean_sales, nrow(test_set))
median_predictions <- rep(median_sales, nrow(test_set))

# Calculate the MAE for the mean and median predictions
mae_mean <- MAE(mean_predictions, test_set$Sales)
mae_median <- MAE(median_predictions, test_set$Sales)

# Print the MAE for the mean and median predictions
print(paste("MAE for mean prediction:", mae_mean))
print(paste("MAE for median prediction:", mae_median))
```

These both returned a MAE of about 182 which is far higher than the MAE which we found. This is a very good sign for our predictions, from the test and training data.

The last step is to use the k-fold method to look at the effectiveness of the linear regression model. Using a k value of 10 is standard for models with the number of rows that are found in this dataset. First set the parameters of the train control and then implement the model that was found before using feature selection. Once this is complete print out the model and look at the metrics that are used to determine the effectiveness of a model.

```
#Use the k-fold method
train_control <- trainControl(method = "cv",   number = 10)

#implement the method
model <- train(Sales ~ PromotionSpend + CompetitorSpend + FootTraffic + OnlineTraffic,
               data = train_set, method = "lm", trControl = train_control)

print(model)
```

An r-squred value of 0.9394104 was produced which is a very high correlation.
MAE was at 43.29771 which was again very low compared to the baseline model which suggests that our model performs much greater than if all variables had remained.

The last value that came was the RMSE which is the Root Mean Squared Error. This is found by finding the square of the real – predicted values divided by the number of occurrences and taking the square root of this number. In this case it comes out to be 54.4642.

Orange Linear Regression

Clean the data

This needs to be performed in excel.

There are a few ways it could be done in orange. Using a python script or a formula however using a tool such as excel does make this a lot quicker.

Load your data into excel

Change the data type to numerical for the values that need to be

Then change the decimal place to fit suitable places, so that the new data looks like below

	A	B	C	D	E	F	G	H
1	AvgTemperature	Holiday	PromotionSpend	OnlineTraffic	FootTraffic	Weekday	CompetitorSpend	Sales
2	22.5	0	328.65	973	519	4	297.69	1239.03
3	19.3	0	211.99	1426	577	2	326.02	1340.75
4	23.2	0	407.80	1264	577	4	231.04	1587.87
5	27.6	1	174.82	940	583	4	204.07	1124.36
6	18.8	1	229.47	1195	554	2	101.61	1300.88

Following this a change needs to be made to the data so that any duplicates or NA variables are accounted for.

Go to data -> highlight all your data -> remove duplicates

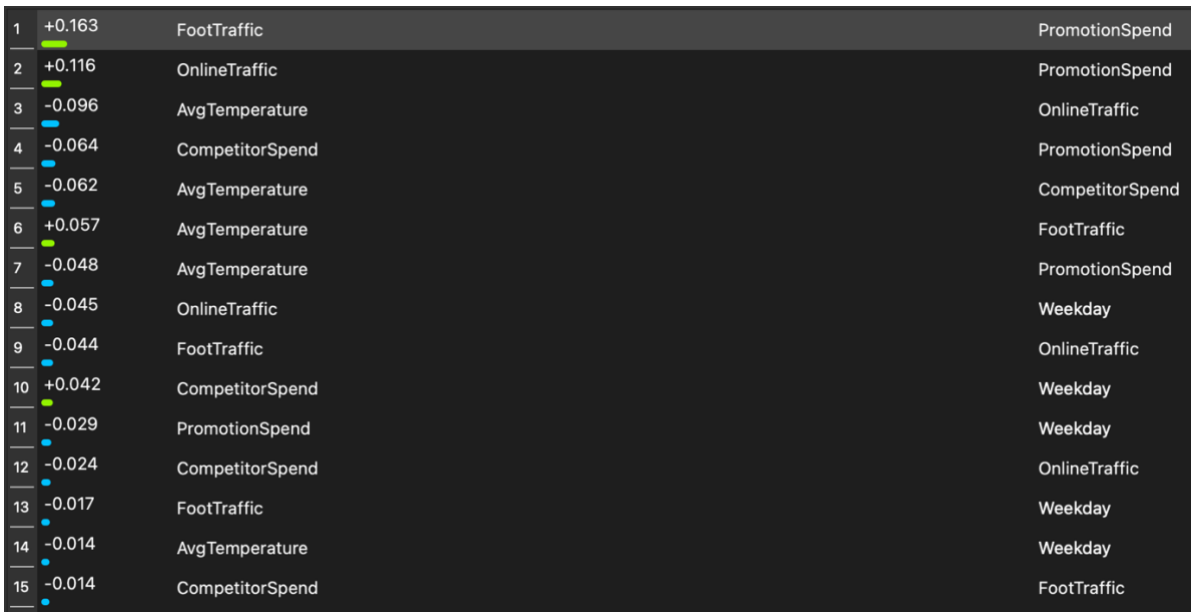
The data showed no duplicates so nothing was removed from the table

Go to data -> Sort -> sort -> select column -> order from smallest to largest so if there are any missing values

The data showed no N/A / missing values that needed to be removed from the data

Regression Analysis

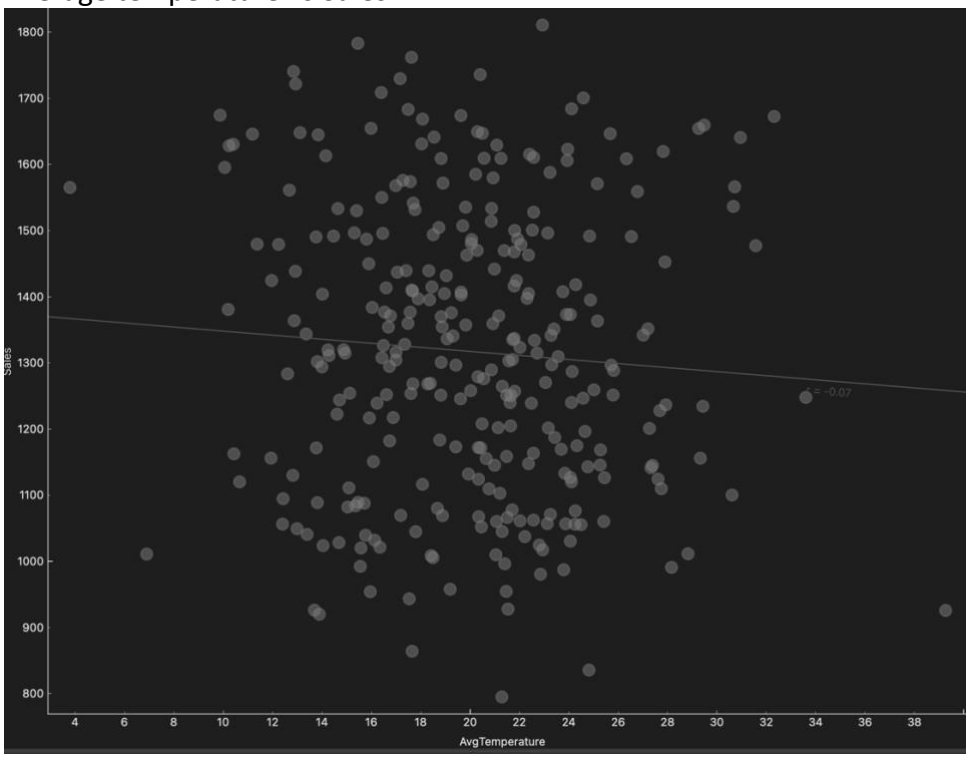
Correlation Matrix



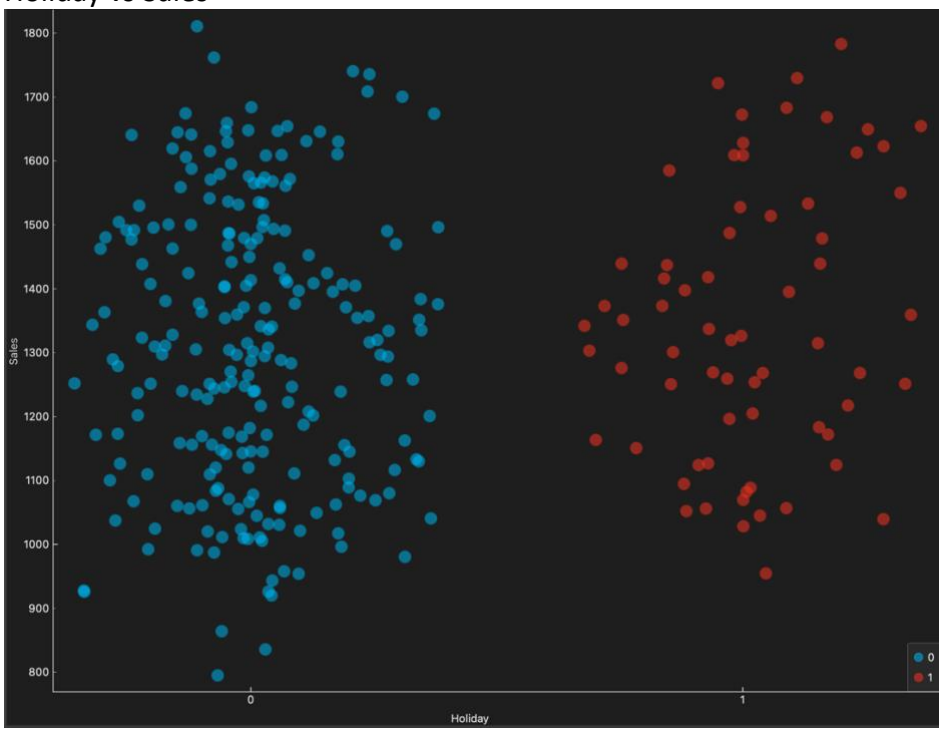
There are no highly correlated values. This means that there is minimal redundant data amongst the variables. None of the variables need to be removed because they correlate too highly with one another.

Another way to check the influence that the different variables have on sales is through looking at different scatterplots.

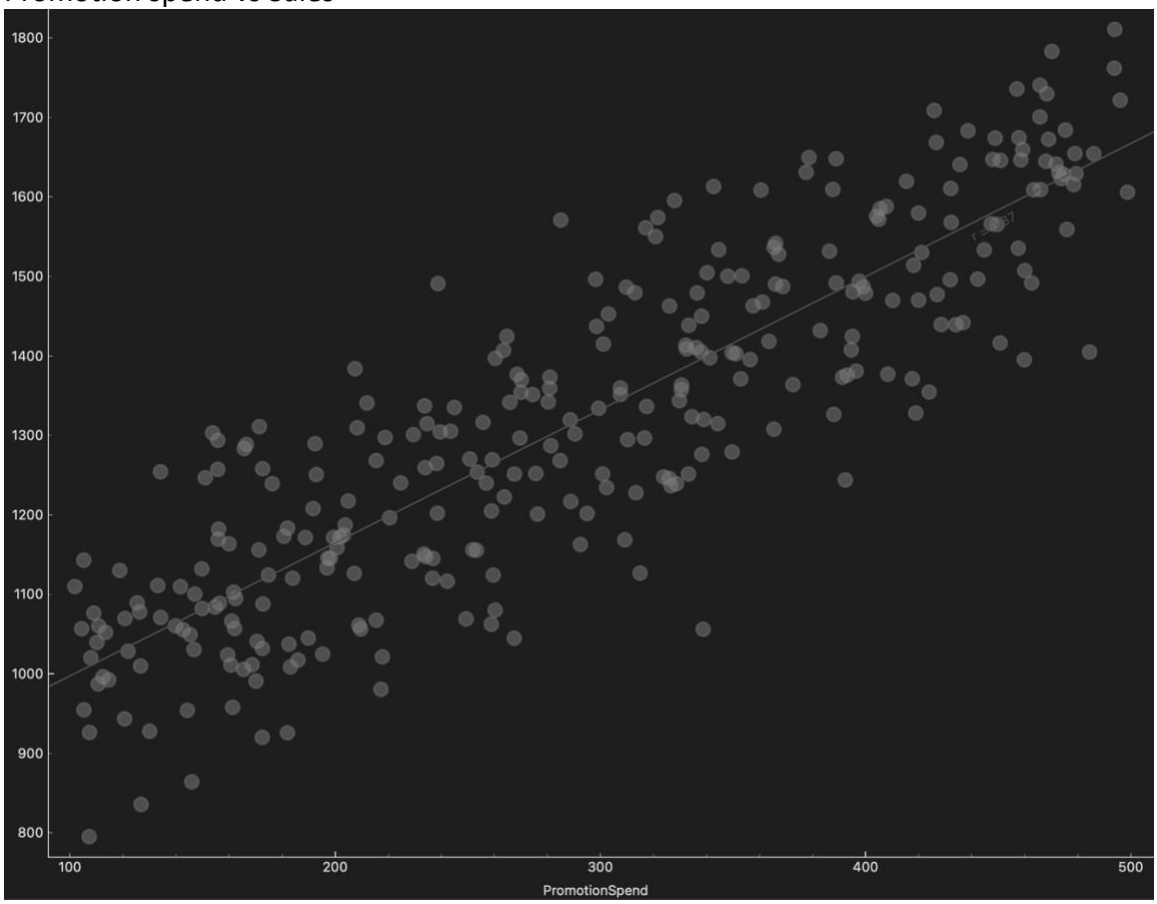
Average temperature vs Sales



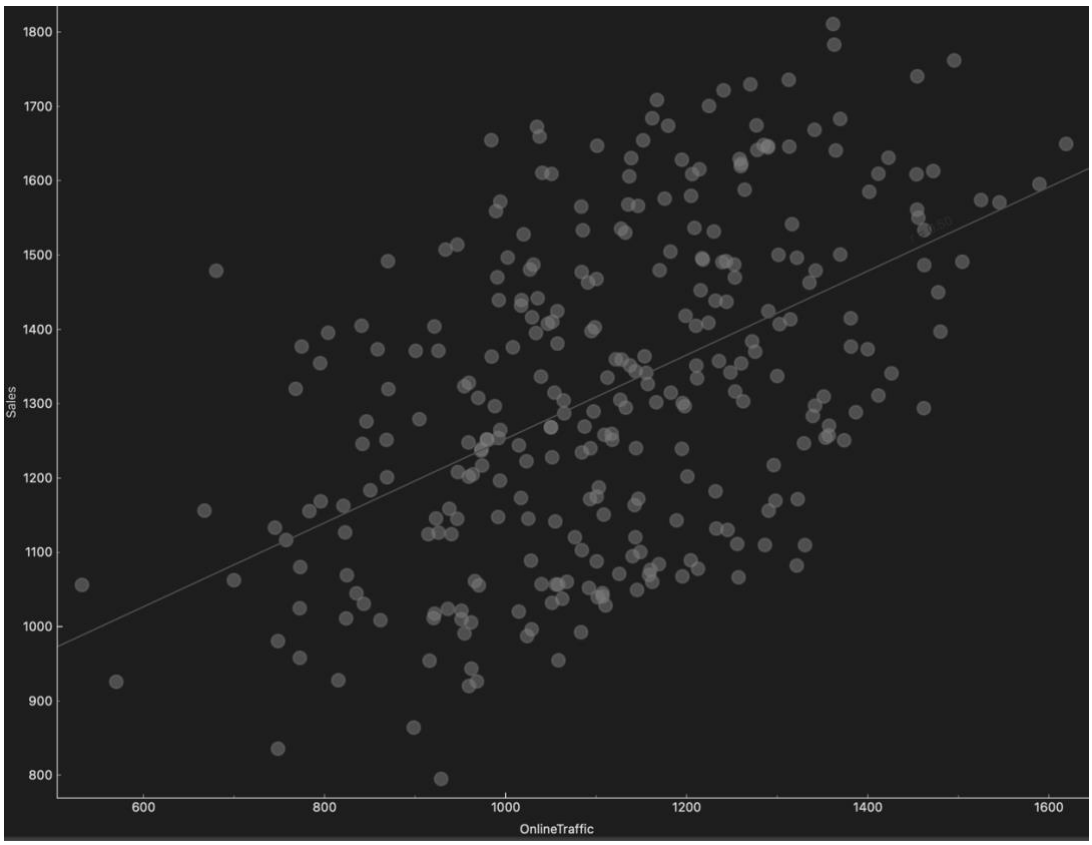
Holiday vs Sales



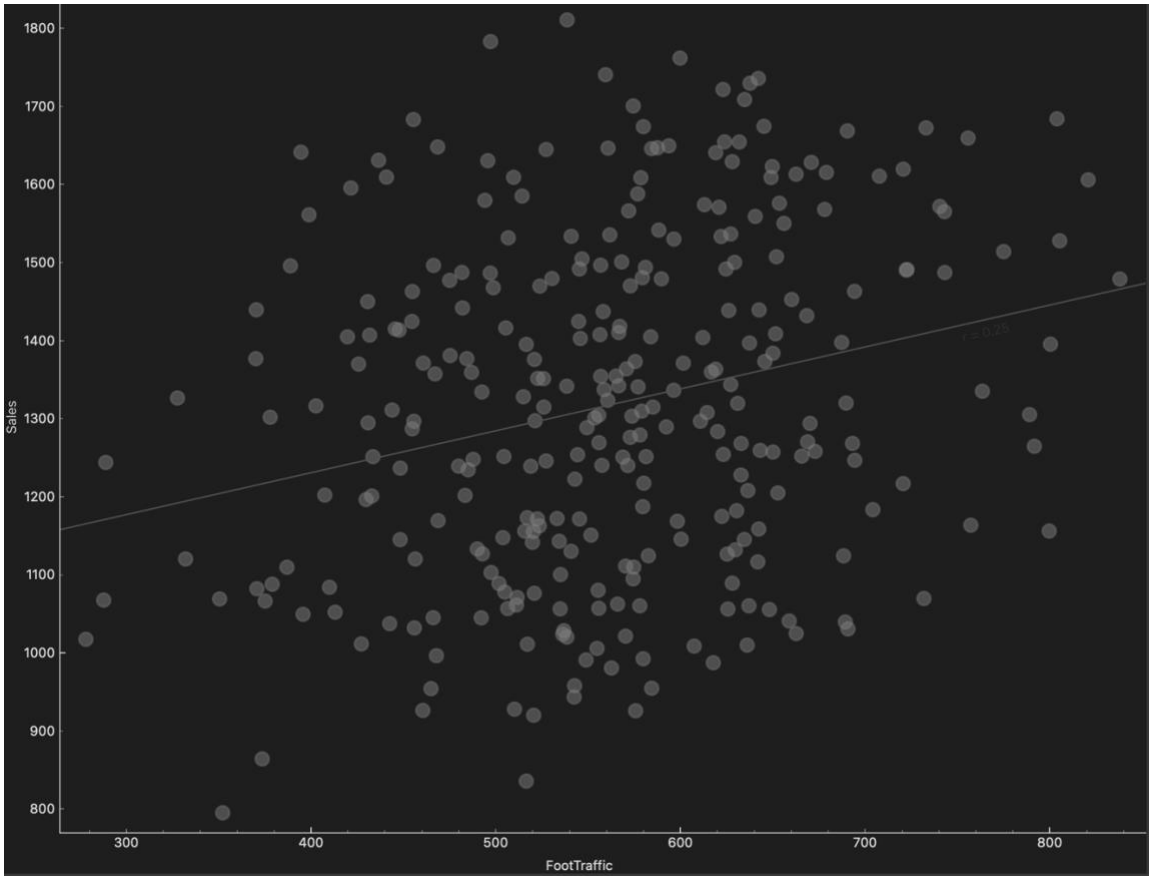
Promotion spend vs Sales



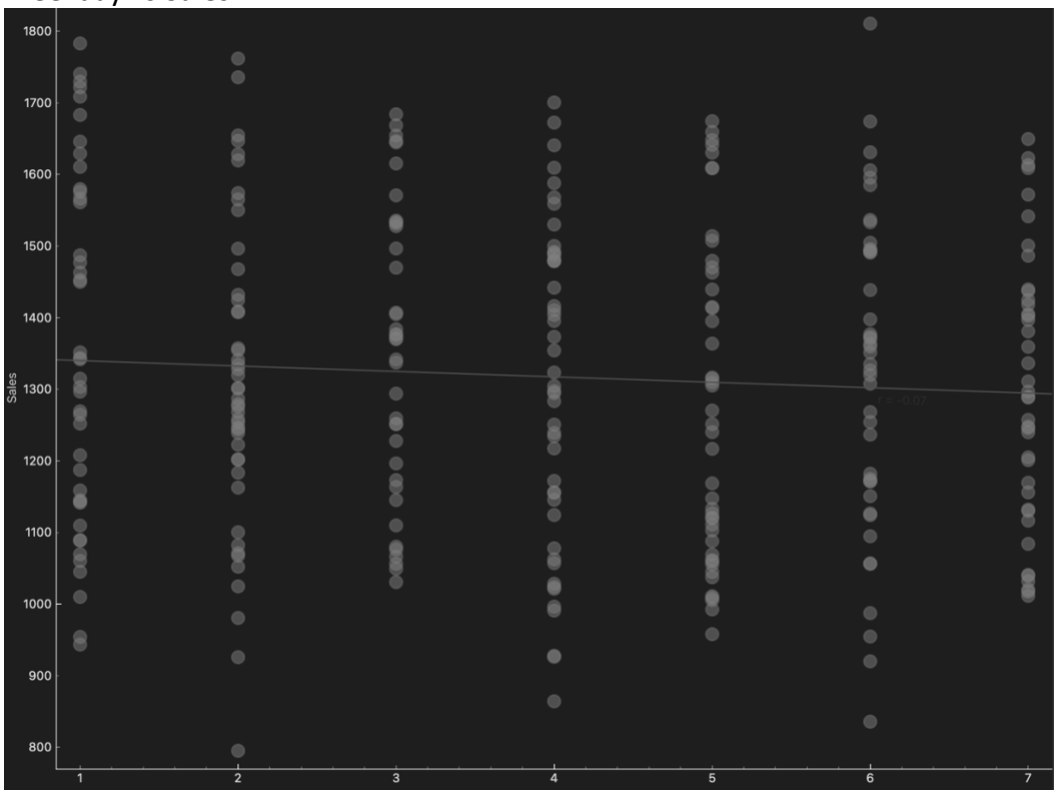
Online traffic vs Sales



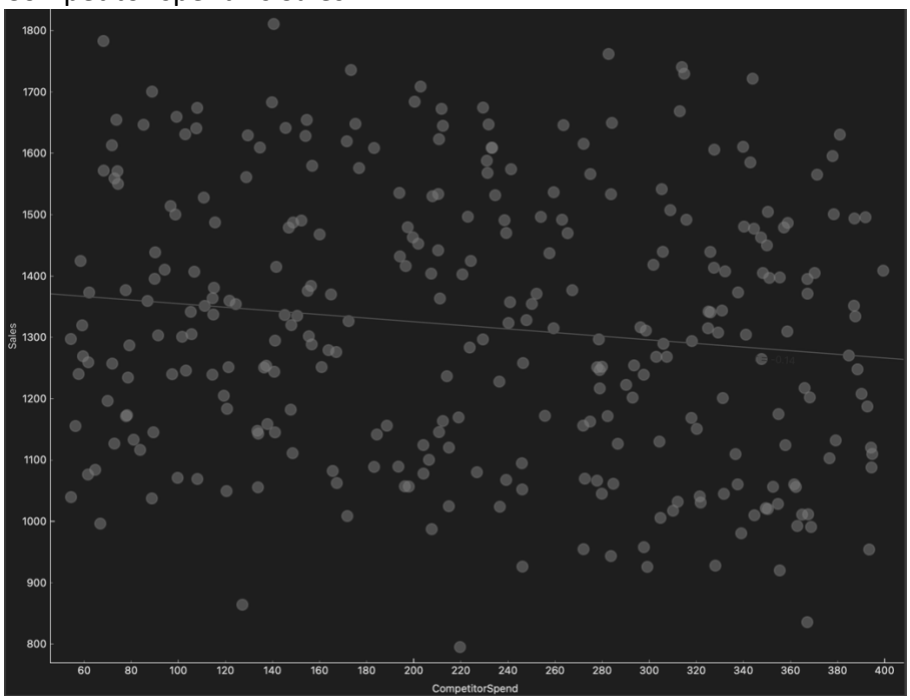
Foot traffic vs sales



Weekday vs Sales

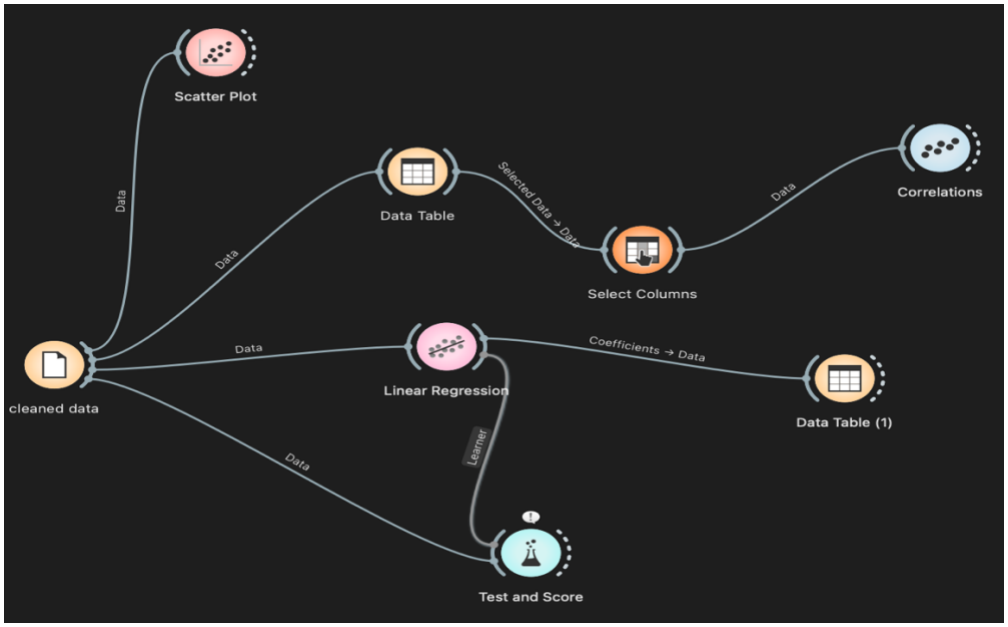


Competitor spend vs Sales



This is a very useful and interactive component of orange where you are able to customise the graphs dependent on different variables. For example you are able to highlight the points based on certain categorical values or ranges of values, this can help create more information to graphs.

The strongest correlation between a variable and sales seems to be PromotionalSpend which makes sense as the company spends more money on marketing there is a greater audience who knows about the product and higher levels of purchases. Average temperature, weekday and holiday were all variables which showed a very low correlation and might need to be removed from the model.



The next step is to look at the linear regression of the model review its effectiveness. Including all variables from the original data this is what the equation would look like.

	name	coef
1	intercept	220.917
2	AvgTempera...	-0.142853
3	Holiday=0	2.9339
4	Holiday=1	-2.9339
5	PromotionSp...	1.521
6	OnlineTraffic	0.469185
7	FootTraffic	0.318709
8	Weekday	-2.28661
9	CompetitorS...	-0.167682

Then using random sampling where the training set is 66% of data and 33% test data it produces the following measurements of effectiveness.

Model	MSE	RMSE	MAE	MAPE	R2
Linear Regression	2735....	52.302	41.585	0.033	0.942

This linear regression model has a very high R^2 value as well as a low MAE, if you compare these to the values that were found in R.

R^2 – 0.9510692
MAE – 39.60793

The R^2 is only slightly lower with the Orange model and the MAE is only slightly higher. So according to these metrics including all of the variables produces a good model.

Discussion & Comparison

A. Which do you believe is easier?
B. Which tells you more?
C. Which would be better for convincing others of your conclusions?

A. If you want to look at the very basic information that is given to you, using Orange is likely the best tool. It is very quick because it reduces the amount of cod that is required which can speed the initial process of understanding your data. The correlation matrix is much easier to understand in orange as wel las producing very informative graphs occurs faster on orange.

B. RStudio tells more. While Orange is quite limited when it comes into more in depth statistical learning and analysis. R can go very deep and using a package such as RCommander takes the ease of use that is found in Orange alongside the power of the statistical tools that are found in R to produce a very useful tool. You can see from the

analysis that there is just more to explore in R. It makes feature selection a more in depth process and is definitely aimed at those who have a decent understanding of statistics.

C.I think that if you are working at a company and you wanted everyone to be able to understand how to deal with data, showing them excel and orange would be a great way of getting them up to scratch. However if you want to be able to understand statistical metrics such as BIC, AIC and building up residual based plots, R is going to be most useful. R can also be expanded very deep. R has many packages that utilise certain components of its programming software to create powerful statistical learning tools. If you want to build more complex models and visualise the data using a tool such as R is going to be easier after getting over initial bottlenecks that come with a programming language compared to a software tool such as Orange.