Regression

Justin Hardy & Benji Frenkel

Description

Linear Regression is a concept in statistics that is powerful in the context of machine learning. Its goal is to determine how much we can expect a given y-value to change for every change in the x-value. Within the R language, we can build Linear Regression Models that will not only determine this linear relationship between two (or more) variables in a given data set, but also provide us with various useful information to help us gauge how strong of a relationship these variables have by providing us with useful coefficients.

The main strength of Linear Regression is that it's both incredibly simple and powerful. This is especially true for data sets that follow a linear pattern.

The main weakness of Linear Regression is that it has considerably high bias, at the expense of having low variance. This is purely due to the fact that it tries to fit the data to a linear shape, which can become problematic when the data being used doesn't always follow a linear shape.

In this assignment, we'll explore linear regression in detail. I've picked out a data set online that consists of web data collected on Online Shoppers for, what is presumably, a retail store.

Modeling

Data Set Setup

Starting out, we'll load our data set into R.

```
# data set input
ShopperIntentions <- read.csv("online_shoppers_intention.csv")
```

I'll go ahead and create a factor for various qualitative values, that I feel like I'll want to use later in the assignment.

```
# data set cleanup
ShopperIntentions$Month <- factor(ShopperIntentions$Month, levels=c("Jan", "Feb", "Mar", "Apr", "May",
ShopperIntentions$VisitorType <- factor(ShopperIntentions$VisitorType, levels=c("Returning_Visitor", "N
ShopperIntentions$Weekend <- factor(ShopperIntentions$Weekend)
```

Dividing Into Train / Test

Next, we'll split the data into train & test as per the machine learning process.

```
# train/test division
i <- sample(1:nrow(ShopperIntentions), nrow(ShopperIntentions)*0.8, replace=FALSE)
train <- ShopperIntentions[i,]
test <- ShopperIntentions[-i,]
```

Data Exploration / Graphing

We'll be exploring the data within our train data set, which makes up 80% of the shopper intentions data set. The following are various details/statistics about the data set itself:

Rows / Columns Info:

```
## 'data.frame':    9864 obs. of  18 variables:
##  $ Administrative         : int  0 0 3 1 5 0 0 0 4 1 ...
##  $ Administrative_Duration: num  0 0 37 12.1 98.3 ...
##  $ Informational          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ProductRelated         : int  13 13 62 45 279 31 18 22 3 62 ...
##  $ ProductRelated_Duration: num  598 222 1193 591 9485 ...
##  $ BounceRates            : num  0 0 0.00308 0 0.00311 ...
##  $ ExitRates              : num  0.0167 0.0205 0.0154 0.013 0.0186 ...
##  $ PageValues             : num  0 0 21.602 0 0.652 ...
##  $ SpecialDay             : num  0 0 0 0 0 0.4 0 0.6 0 0 ...
##  $ Month                  : Factor w/ 12 levels "Jan","Feb","Mar",..: 11 11 11 9 11 5 8 5 11 12 ...
##  $ OperatingSystems       : int  2 2 2 2 2 2 2 3 3 1 ...
##  $ Browser                : int  10 2 2 2 2 2 2 2 2 1 ...
##  $ Region                 : int  7 1 1 1 1 1 3 7 1 6 ...
##  $ TrafficType            : int  2 2 1 2 2 3 13 13 4 2 ...
##  $ VisitorType            : Factor w/ 3 levels "Returning_Visitor",..: 2 2 1 1 1 1 1 1 1 1 ...
##  $ Weekend                : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 2 1 1 1 1 1 ...
##  $ Revenue                : logi  FALSE FALSE TRUE TRUE FALSE FALSE ...
```

Sample of First Five Rows:

```
##       Administrative Administrative_Duration Informational
## 9725               0                 0.00000             0
## 9172               0                 0.00000             0
## 8297               3                37.00000             0
## 5784               1                12.06667             0
## 11597              5                98.35000             0
##       Informational_Duration ProductRelated ProductRelated_Duration BounceRates
## 9725                       0             13                 598.5000 0.000000000
## 9172                       0             13                 221.5000 0.000000000
## 8297                       0             62                1193.0000 0.003076923
## 5784                       0             45                 591.2641 0.000000000
## 11597                      0            279                9485.3746 0.003113824
##        ExitRates PageValues SpecialDay Month OperatingSystems Browser Region
## 9725  0.01666667  0.0000000          0   Nov                2      10      7
## 9172  0.02051282  0.0000000          0   Nov                2       2      1
## 8297  0.01538462 21.6018009          0   Nov                2       2      1
## 5784  0.01302863  0.0000000          0   Sep                2       2      1
## 11597 0.01856189  0.6517818          0   Nov                2       2      1
##       TrafficType       VisitorType Weekend Revenue
## 9725            2       New_Visitor   FALSE   FALSE
## 9172            2       New_Visitor   FALSE   FALSE
## 8297            1 Returning_Visitor   FALSE    TRUE
## 5784            2 Returning_Visitor   FALSE    TRUE
## 11597           2 Returning_Visitor    TRUE   FALSE
```

Sample of Last Five Rows:

```
##       Administrative Administrative_Duration Informational
## 2768               0                       0             0
## 624                0                       0             0
## 1260               0                       0             0
## 10900              0                       0             0
## 1558               1                       3             0
##       Informational_Duration ProductRelated ProductRelated_Duration BounceRates
## 2768                       0             35                1386.717  0.02000000
## 624                        0              6                  13.000  0.13333333
## 1260                       0              6                 546.000  0.00000000
## 10900                      0             18                 327.650  0.01111111
## 1558                       0             14                 128.000  0.00000000
##        ExitRates PageValues SpecialDay Month OperatingSystems Browser Region
## 2768  0.03926984   14.96432          0   May                1       8      4
## 624   0.16666667    0.00000          0   Mar                2       2      1
## 1260  0.04000000   35.49000          0   Mar                2       2      3
## 10900 0.04444444    0.00000          0   Nov                1       1      1
## 1558  0.02857143    0.00000          0   Mar                2       2      1
##       TrafficType       VisitorType Weekend Revenue
## 2768           20 Returning_Visitor    TRUE    TRUE
## 624             3 Returning_Visitor   FALSE   FALSE
## 1260            1 Returning_Visitor   FALSE    TRUE
## 10900           1 Returning_Visitor   FALSE   FALSE
## 1558            2 Returning_Visitor   FALSE   FALSE
```

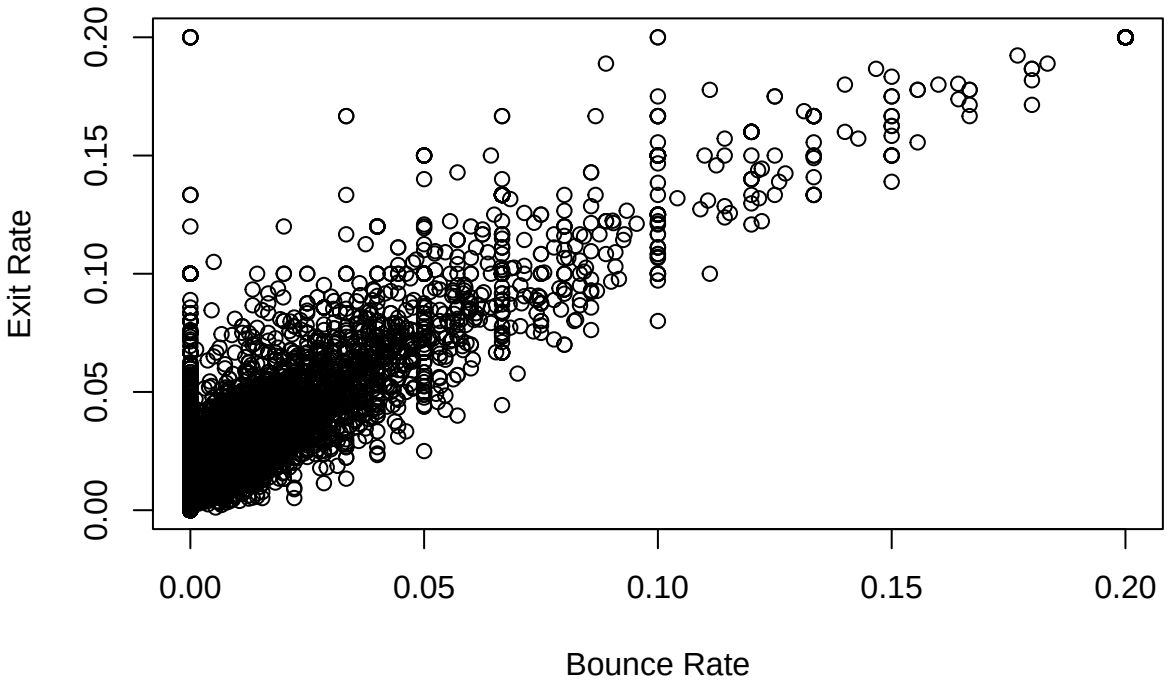
NA Count:

```
## [1] "Number of NAs: 0"
```

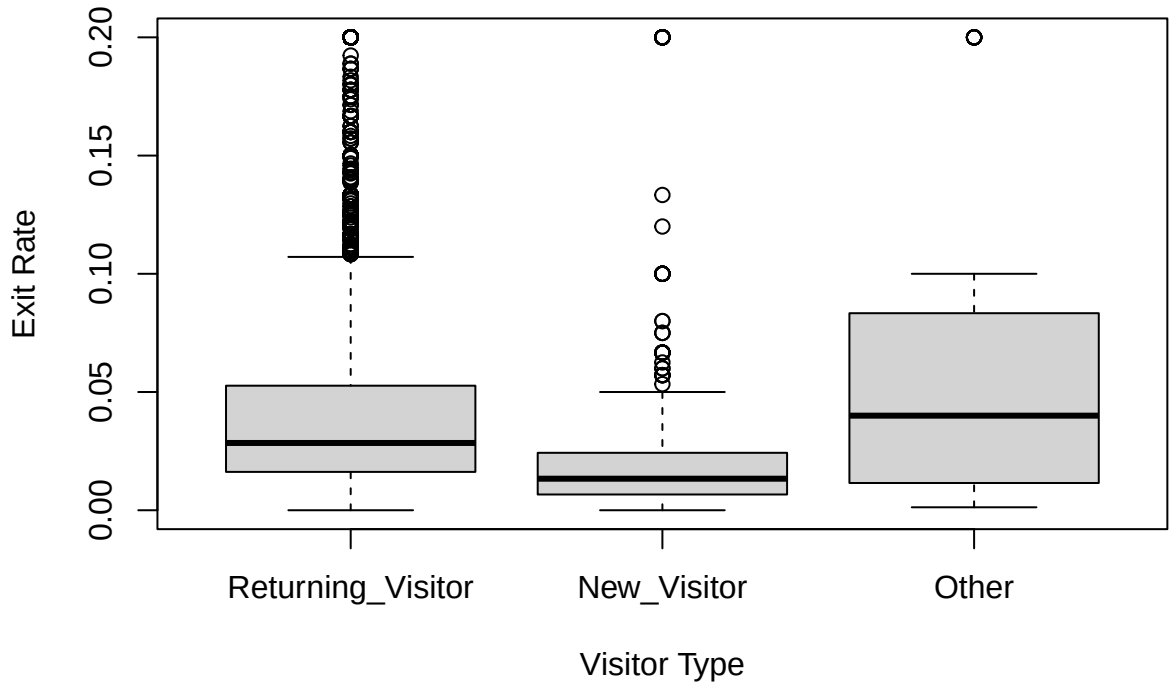
General Summary:

```
##  Administrative  Administrative_Duration Informational
##  Min.   : 0.00   Min.   :   0.00         Min.   : 0.0000
##  1st Qu.: 0.00   1st Qu.:   0.00         1st Qu.: 0.0000
##  Median : 1.00   Median :   7.00         Median : 0.0000
##  Mean   : 2.31   Mean   :  79.46         Mean   : 0.5049
##  3rd Qu.: 4.00   3rd Qu.:  92.19         3rd Qu.: 0.0000
##  Max.   :27.00   Max.   :3398.75         Max.   :16.0000
##
##  Informational_Duration ProductRelated   ProductRelated_Duration
##  Min.   :   0.00        Min.   :  0.00   Min.   :    0.0
##  1st Qu.:   0.00        1st Qu.:  7.00   1st Qu.:  183.1
##  Median :   0.00        Median : 18.00   Median :  588.4
##  Mean   :  34.34        Mean   : 31.58   Mean   : 1187.3
##  3rd Qu.:   0.00        3rd Qu.: 37.00   3rd Qu.: 1456.8
##  Max.   :2549.38        Max.   :686.00   Max.   :63973.5
##
##   BounceRates         ExitRates         PageValues        SpecialDay
##  Min.   :0.000000   Min.   :0.00000   Min.   :  0.000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.01429   1st Qu.:  0.000   1st Qu.:0.00000
```

```
##  Median :0.003125   Median :0.02517   Median :  0.000   Median :0.00000
##  Mean   :0.022165   Mean   :0.04307   Mean   :  5.939   Mean   :0.06121
##  3rd Qu.:0.017143   3rd Qu.:0.05000   3rd Qu.:  0.000   3rd Qu.:0.00000
##  Max.   :0.200000   Max.   :0.20000   Max.   :361.764   Max.   :1.00000
##
##      Month      OperatingSystems    Browser           Region
##  May    :2683   Min.   :1.000    Min.   : 1.000   Min.   :1.000
##  Nov    :2409   1st Qu.:2.000    1st Qu.: 2.000   1st Qu.:1.000
##  Mar    :1525   Median :2.000    Median : 2.000   Median :3.000
##  Dec    :1376   Mean   :2.119    Mean   : 2.354   Mean   :3.137
##  Oct    : 458   3rd Qu.:3.000    3rd Qu.: 2.000   3rd Qu.:4.000
##  Sep    : 363   Max.   :8.000    Max.   :13.000   Max.   :9.000
##  (Other):1050
##   TrafficType                VisitorType    Weekend      Revenue
##  Min.   : 1.00   Returning_Visitor:8449   FALSE:7567   Mode :logical
##  1st Qu.: 2.00   New_Visitor      :1351   TRUE :2297   FALSE:8340
##  Median : 2.00   Other            :  64                TRUE :1524
##  Mean   : 4.08
##  3rd Qu.: 4.00
##  Max.   :20.00
##
```



Graphs:



Simple Linear Regression Model

We'll start by making a simple linear regression model, where we use the ExitRates column as our target, and the BounceRates column as our predictor. ExitRates simply describes the percentage of paegs that were the last visited, while BounceRates describes the rate at which a users enter the site from that page and also leave from that same page. We'll then generate a summary of the model, so we can see the residuals and what R thinks about the correlation between the two columns.

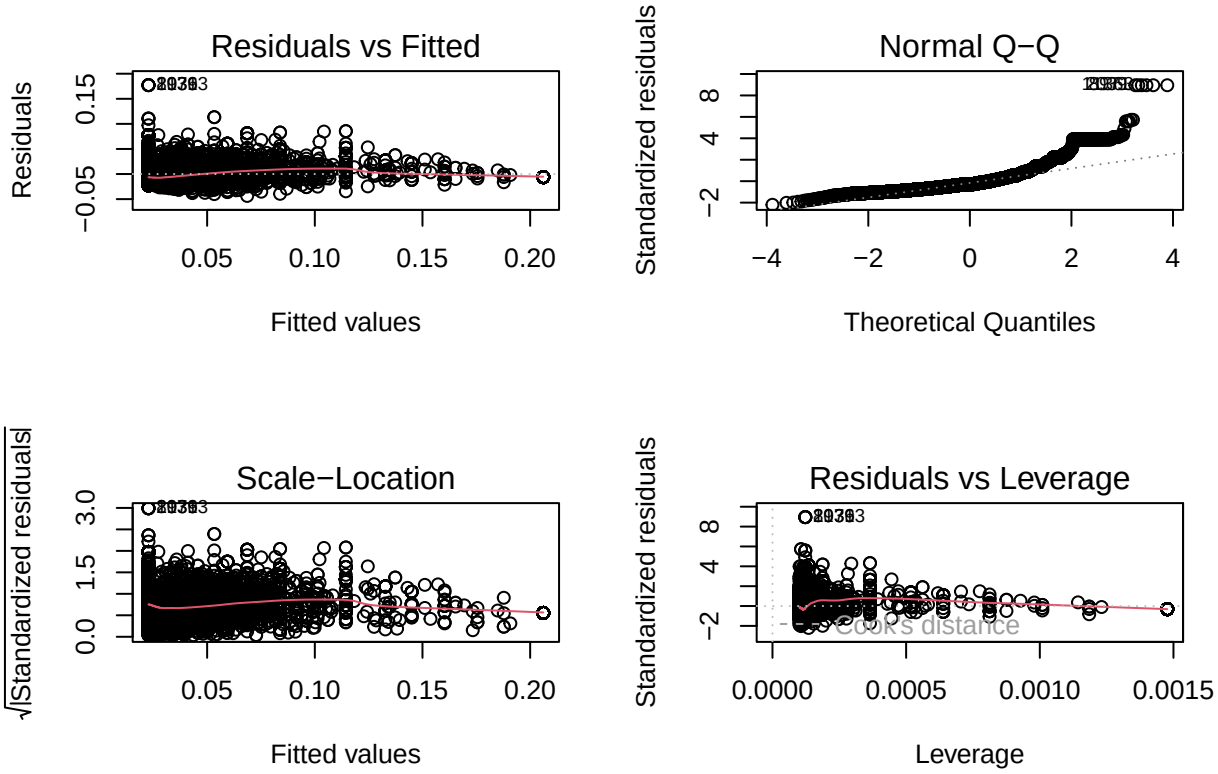
```
# linear regression model
lm_simple <- lm(ExitRates~BounceRates, data=train)

# summary
summary(lm_simple)
```

```
##
## Call:
## lm(formula = ExitRates ~ BounceRates, data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.043589 -0.012229 -0.006087  0.005816  0.177244
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0227557  0.0002194   103.7   <2e-16 ***
```

```
## BounceRates 0.9166571  0.0041294   222.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0198 on 9862 degrees of freedom
## Multiple R-squared:  0.8332, Adjusted R-squared:  0.8332
## F-statistic: 4.928e+04 on 1 and 9862 DF,  p-value: < 2.2e-16
```

In the above summary, we can see that our R-squared value comes out to a value of 0.8336. Of course, we'd like this to be as close as possible to 1, so this isn't necessarily bad, but could be better. Looking at our coefficients, R seems to think this predictor is good for the model, as indicated by the significant code, which gave it three *'s. We can also observe that our degree of freedom is fairly high, with a low RSE.

Now that we've created the model and looked at the summary of it, we'll plot the residuals.



We'll go through each plot in detail, so that we can better understand what these plots tell us about the data.

Residuals vs Fitted

This plot aims to show us whether or not there exists a non-linear relationship between the residuals.

In this case, there doesn't seem to be any distinctive pattern in the plot, as the red-line indicated that there is a fairly linear relationship between the residuals and their fitted values.

If we're being generous, their may be a slight upwards parabola formed as there seems to be a slight rise near the median of the data. But it's fairly safe to assume we have a linear relationship here.

Normal Q-Q

This plot aims to show us whether or not he residuals are normally distributed. Generally, we want the residuals to be lined up neatly in a straight line. Generally, if this is close-enough to the case, the dashed line will be lined up nicely with the residuals.

It appears this is the case, with the exception of a couple of outliers between the theoretical quantiles between -1 and -2. Probably nothing to worry about, however.

Scale-Location

This plot aims to show us whether or not the residuals spread equally along the predictor's ranges. We want this plot's red line to be horizontal, with fairly equally spread points outside of it.

We can observe a slight dip at the beginning of our plot, but for the most part, the plot's red line remains fairly horizontal & straight. And regardless, the points are spread fairly equally in relation to the line, up until we get to fitted values of roughly 0.075~. At this point, we notice that as the value grows, the spread increases quite notably.

Residuals vs Leverage

This plot aims to help us understand whether or not our plot contains influential cases, based off of whether or not our standardized residuals lie outside of Cook's Distance.

We can observe that there aren't any dashed red lines denoting the 0.5 and 1 marks of Cook's Distance, and that our plot points lie very closely in the center of the Cook's Distance area. This simply means that not many of our points are influential to the regression. In other words, there are no influential cases.

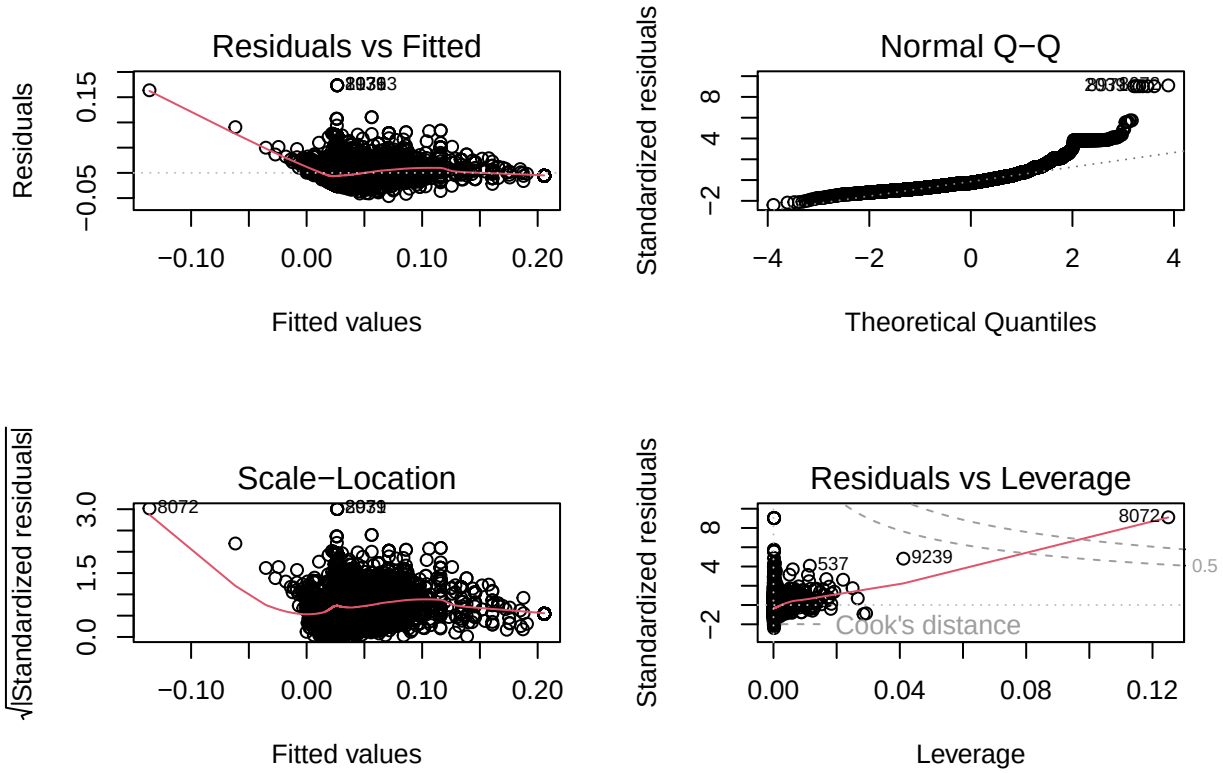
Multiple Linear Regression Model

The next linear regression model we'll create will use multiple predictors to predict the same target, ExitRates. We'll be adding the various Duration column values to the line-up of predictors, and seeing how this improves the model.

```
# linear regression model
lm_multiple <- lm(ExitRates~BounceRates+Administrative_Duration+Informational_Duration+ProductRelated_Du

# summary
summary(lm_multiple)
```

```
##
## Call:
## lm(formula = ExitRates ~ BounceRates + Administrative_Duration +
##     Informational_Duration + ProductRelated_Duration, data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.046033 -0.011854 -0.005188  0.006259  0.173547
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              2.645e-02  2.615e-04 101.175   <2e-16 ***
## BounceRates              8.963e-01  4.095e-03 218.899   <2e-16 ***
## Administrative_Duration -1.442e-05  1.209e-06 -11.927   <2e-16 ***
## Informational_Duration  -1.464e-06  1.484e-06  -0.987    0.324
## ProductRelated_Duration -1.727e-06  1.144e-07 -15.089   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.01922 on 9859 degrees of freedom
## Multiple R-squared:  0.8429, Adjusted R-squared:  0.8428
## F-statistic: 1.322e+04 on 4 and 9859 DF,  p-value: < 2.2e-16
```



The main change I'd like to note about the residuals is that now, we're able to not only observe the Cook's Distance markers, but also now we have a case that lies outside the Cook's Distance area (marked 8072). Although, for divisons of train/test, this won't be the case, we can still see outliers that lie close to the Cook's Distance dashed lines. We can also observe pretty significant dips at the beginning of the Residuals vs Fitted and Scale-Location plots, the latter forming - then breaking - the shape of a parabola. We can interpret this as there not being a linear relationship between the residuals and their fitted values for fitted values less than 0, to which afterwards the line straightens up significantly.

Additionally, we can see from the summary that Informational Duration isn't too great of a predictor for this model, so it may be best to remove it if we're looking at improving the model. Our R-squared on the other hand has increased considerably!

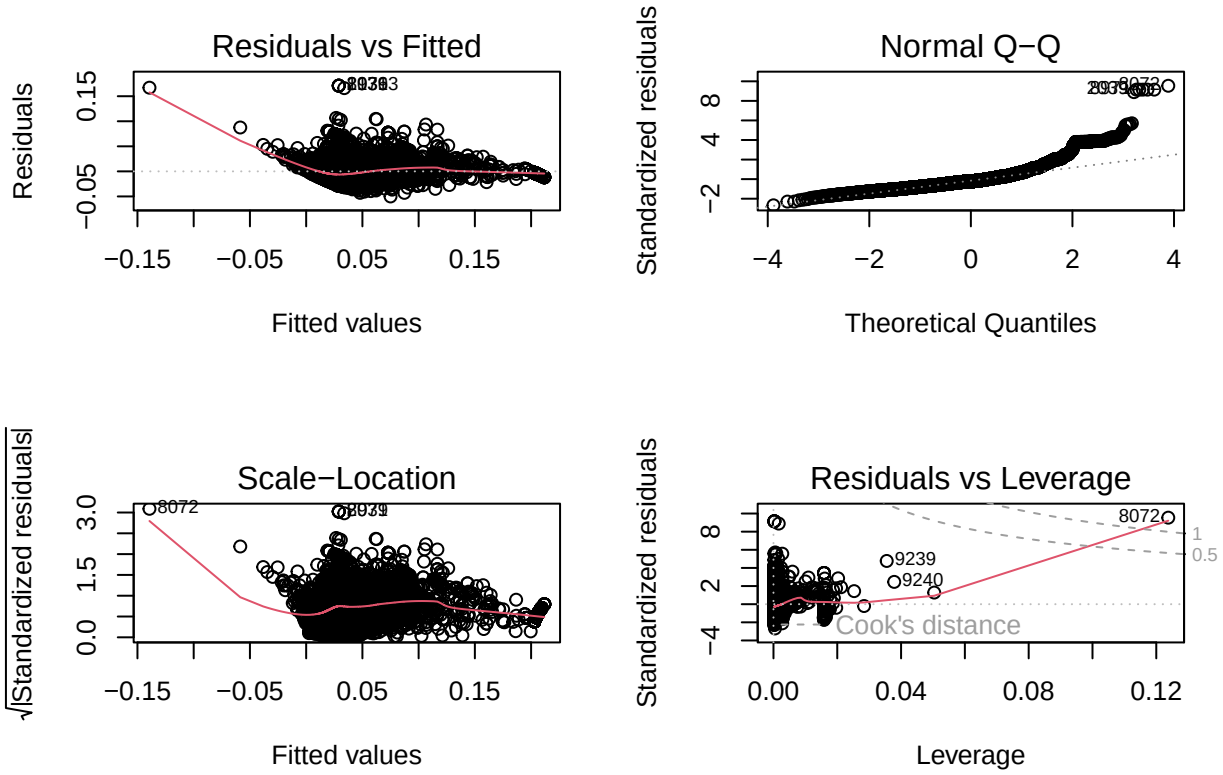
Final Linear Regression Model

We'll create one final linear regression model that will include various other columns as predictors. The most notable additions are columns pertaining to special days (holidays/weekends), and whether or not the visitor is new.

```
# linear regression model
lm_final <- lm(ExitRates~BounceRates+Administrative_Duration+ProductRelated_Duration+PageValues+Visitor
```

```r
# summary
summary(lm_final)
```

```
##
## Call:
## lm(formula = ExitRates ~ BounceRates + Administrative_Duration +
##     ProductRelated_Duration + PageValues + VisitorType + Weekend +
##     SpecialDay, data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.050015 -0.011262 -0.004387  0.005446  0.171030
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.897e-02  3.083e-04  93.952  < 2e-16 ***
## BounceRates                8.771e-01  4.066e-03 215.734  < 2e-16 ***
## Administrative_Duration   -1.227e-05  1.170e-06 -10.491  < 2e-16 ***
## ProductRelated_Duration   -1.989e-06  1.083e-07 -18.373  < 2e-16 ***
## PageValues                -1.477e-04  1.025e-05 -14.405  < 2e-16 ***
## VisitorTypeNew_Visitor    -8.456e-03  5.652e-04 -14.961  < 2e-16 ***
## VisitorTypeOther           7.317e-03  2.355e-03   3.107 0.001895 **
## WeekendTRUE               -1.671e-03  4.465e-04  -3.742 0.000184 ***
## SpecialDay                 6.153e-03  9.582e-04   6.421 1.42e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0187 on 9855 degrees of freedom
## Multiple R-squared:  0.8514, Adjusted R-squared:  0.8513
## F-statistic:  7060 on 8 and 9855 DF,  p-value: < 2.2e-16
```



Going over the changes in brief detail, we see improvements to our R-squared value by including these predictors, at the cost of a drop in our RSE.

Results Comparison / Analysis

It should be relatively clear that the third & final linear regression model created is the best model here. Simply because the model utilizes various other predictive factors to improve on the accuracy of the already accurate first/simple model that preceded it. As noted, the first model created was simple, but also incredibly accurate at its base due to the strong correlation between the exit and bounce column. Logically speaking, we can make a number of assumptions about the user's exit rates given the bounce rate.

Of course, one could argue that the difference in accuracies of the models is almost negligible, therefore the first simple model we created is the best to use. However, I'd argue the difference between the first and last - simple and final - models are notable enough to warrant use of the final model over the simpler one. After all, it'd likely transfer over better outside of the context of our data set.

Metrics Correlation and MSE

This is reserved for predictions and evaluations on metrics correlation and MSE. We can also observe that the Residuals vs Leverage graph has more of an upwards parabola shape past 0.01 leverage.

```
# predictions on test data
pred_simple <- predict(lm_simple, newdata=test)
pred_multiple <- predict(lm_multiple, newdata=test)
pred_final <- predict(lm_final, newdata=test)
```

```
## METRICS:

## Simple Linear Regression Model:
## Correlation:  0.91375270859772
## MSE:  0.000397087371285659
## RSE:  0.0199270512441169

## Multiple Linear Regression Model:
## Correlation:  0.919064943284256
## MSE:  0.000373696568921503
## RSE:  0.0193312329902028

## Final Linear Regression Model:
## Correlation:  0.923506439736259
## MSE:  0.000353894969157439
## RSE:  0.0188120963520135
```

Now that we have all of the metrics, let's compare between the three models!

We can see pretty clearly that we make improvements to our correlation values with each linear model. The correlation was already high to begin with in our linear regression model, as there is a strong base correlation between our exit and bounce rate. However, it's made clear through the additions of other relevant data as predictors that the model can notably improve its prediction accuracy.

The same explanation can be applied to our MSE & RSE values. As there is a notable decrease in the mean error of the model.