Homework 6
Group 16

Sujata Sahu
Likhitha Reddy Gundre

1(a)

(i) **Data Understanding**

Using dlookr package performing data quality diagnosis to generate a data quality report:

The following steps are followed:

1. Diagnosis of categorical variable
2. Diagnosis of numeric variable
3. Diagnosis of outliers
4. Visualisation

1. Diagnosis of the Numeric variable

First to get the numeric columns

TrainNumeric<- select_if(Train, is.numeric) %>% as_tibble()

Diagnose the numeric data and arrange the missing percent in decreasing order

diagnose(TrainNumeric)%>%arrange(desc(missing_percent))

```
> diagnose(TrainNumeric)%>%arrange(desc(missing_percent))
# A tibble: 12 × 6
   variables              types   missing_count missing_percent unique_count unique_rate
   <chr>                  <chr>           <int>           <dbl>        <int>       <dbl>
 1 adwordsClickInfo.page  numeric         68260          97.4              6   0.0000856
 2 bounces                numeric         40719          58.1              2   0.0000285
 3 newVisits              numeric         23944          34.2              2   0.0000285
 4 pageviews              numeric             8           0.0114         155   0.00221
 5 sessionId              numeric             0           0            70071   1
 6 custId                 numeric             0           0            47249   0.674
 7 visitStartTime         numeric             0           0            69951   0.998
 8 visitNumber            numeric             0           0              155   0.00221
 9 timeSinceLastVisit     numeric             0           0            20970   0.299
10 isMobile               numeric             0           0                2   0.0000285
11 isTrueDirect           numeric             0           0                2   0.0000285
12 revenue                numeric             0           0             5850   0.0835
>
```

diagnose_numeric(Train) %>% filter(minus>0|zero>0)

```
> diagnose_numeric(Train) %>% filter(minus>0|zero>0)
# A tibble: 4 × 10
  variables           min    Q1      mean median    Q3      max  zero minus outlier
  <chr>             <dbl> <dbl>     <dbl>  <dbl> <dbl>    <dbl> <int> <int>   <int>
1 timeSinceLastVisit    0     0 256450.        0 10375 30074517 47249     0   15588
2 isMobile              0     0      0.229     0     0        1 53993     0   16078
3 isTrueDirect          0     0      0.400     0     1        1 42026     0       0
4 revenue               0     0     10.2       0     0    15981. 64222     0    5849
```

2. Diagnosis of categorial variables

diagnose_category(Train)

```
> diagnose_category(Train)
# A tibble: 185 × 6
   variables levels         N  freq ratio  rank
   <chr>     <chr>      <int> <int> <dbl> <int>
 1 date      2016-12-05 70071   362 0.517     1
 2 date      2016-11-28 70071   352 0.502     2
 3 date      2016-11-29 70071   349 0.498     3
 4 date      2016-10-04 70071   347 0.495     4
 5 date      2016-12-01 70071   331 0.472     5
 6 date      2016-11-30 70071   324 0.462     6
 7 date      2016-12-20 70071   324 0.462     6
 8 date      2016-11-14 70071   323 0.461     8
 9 date      2016-11-03 70071   320 0.457     9
10 date      2016-11-10 70071   318 0.454    10
# … with 175 more rows
# ℹ Use `print(n = ...)` to see more rows
```

3. Diagnosis of Outliers

diagnose_outlier(Train)

```
> diagnose_outlier(Train)
# A tibble: 12 × 6
   variables            outliers_cnt outliers_ratio outliers_mean with_mean without_mean
   <chr>                       <int>          <dbl>         <dbl>     <dbl>        <dbl>
 1 sessionId                       0        0               NaN    4.71e+12     4.71e+12
 2 custId                          0        0               NaN    4.89e+ 4     4.89e+ 4
 3 visitStartTime                  0        0               NaN    1.49e+ 9     1.49e+ 9
 4 visitNumber                 11300       16.1              12.8  3.15e+ 0     1.29e+ 0
 5 timeSinceLastVisit          15588       22.2         1149369.   2.56e+ 5     9.79e+ 2
 6 isMobile                    16078       22.9               1    2.29e- 1     0
 7 isTrueDirect                    0        0               NaN    4.00e- 1     4.00e- 1
 8 adwordsClickInfo.page           5        0.00714          3.8   1.01e+ 0     1    e+ 0
 9 pageviews                    9182       13.1              28.7  6.30e+ 0     2.93e+ 0
10 bounces                         0        0               NaN    1    e+ 0     1    e+ 0
11 newVisits                       0        0               NaN    1    e+ 0     1    e+ 0
12 revenue                      5849        8.35            122.   1.02e+ 1     0
```

To find the outliers of numeric variable

```
> diagnose_outlier(Train)%>%filter(outliers_cnt>0)
# A tibble: 6 × 6
  variables              outliers_cnt outliers_ratio outliers_mean  with_mean without_mean
  <chr>                         <int>          <dbl>         <dbl>      <dbl>        <dbl>
1 visitNumber                   11300        16.1             12.8       3.15         1.29
2 timeSinceLastVisit            15588        22.2        1149369.   256450.         979.
3 isMobile                      16078        22.9              1         0.229        0
4 adwordsClickInfo.page             5         0.00714          3.8       1.01         1
5 pageviews                      9182        13.1             28.7       6.30         2.93
6 revenue                        5849         8.35           122.       10.2          0
```

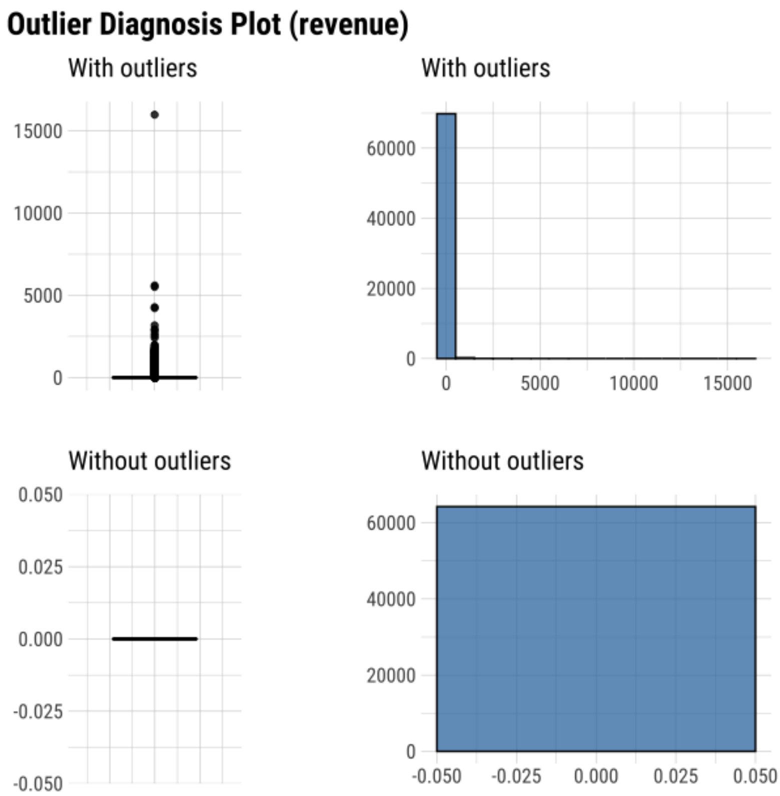
To find the numeric variable with an outlier ratio of 5% or more and then returns the result of dividing mean of outliers by overall mean in descending order.

```
> diagnose_outlier(Train) %>%
+   filter(outliers_ratio > 5) %>%
+   mutate(rate = outliers_mean / with_mean) %>%
+   arrange(desc(rate)) %>%
+   select(-outliers_cnt)
# A tibble: 5 × 6
  variables          outliers_ratio outliers_mean  with_mean without_mean  rate
  <chr>                       <dbl>         <dbl>      <dbl>        <dbl> <dbl>
1 revenue                      8.35          122.       10.2            0  12.0
2 pageviews                   13.1            28.7       6.30         2.93  4.55
3 timeSinceLastVisit          22.2       1149369.   256450.          979.   4.48
4 isMobile                    22.9             1         0.229          0   4.36
5 visitNumber                 16.1            12.8       3.15         1.29  4.07
```

4. Visualisation

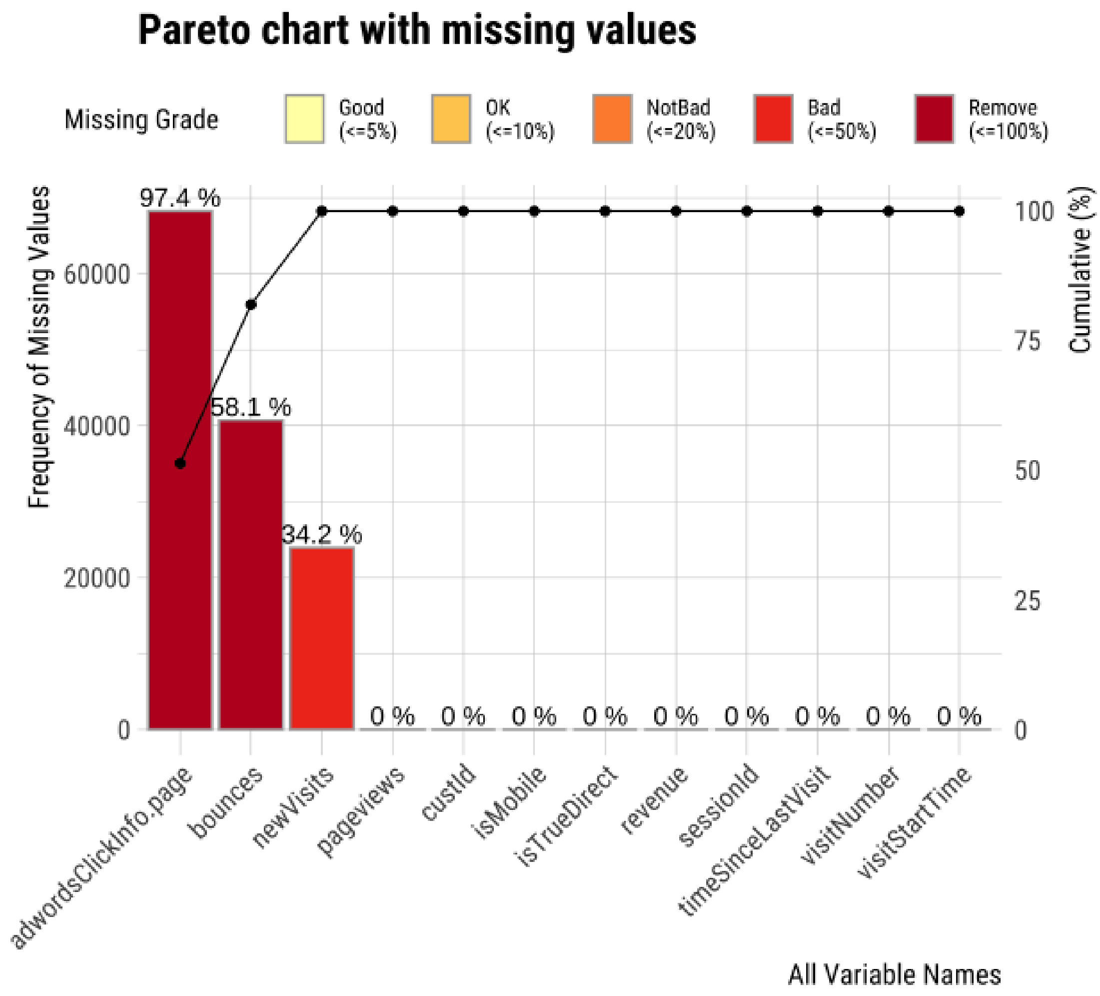
   a. Outlier of revenue

Train %>%
  plot_outlier(revenue)



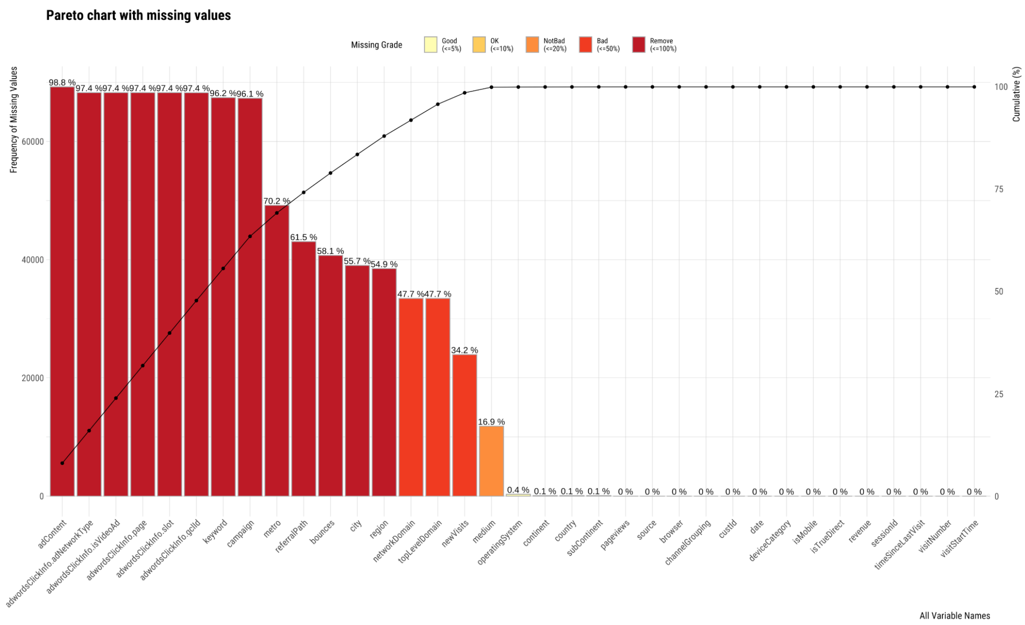
Plot of missing value

```
TrainNumeric %>%
  plot_na_pareto()
```



Overall missing data

```
Train %>%
  plot_na_pareto()
```



Relevance:
Outliers plays a major role in affecting data analysis.Checking the outlier plot of revenue we can see we have one outliers which will really affect the data. Removing the outliers will improve the data quality and help to analyse the data accurately
Missing data is another important factor which needs to be taken into consideration. Without data, analysis can't be made. It's always better to remove the variables with more than 75% of missing data. From the graph we can see adcount contributes the highest missing data value.

(ii).

Data Preparation

```
PreprocessTrainData<-Train %>%
  mutate(date=ymd(date)) %>%
  mutate(country = fct_lump(fct_explicit_na(country), n = 11)) %>%
  mutate(medium = fct_lump(fct_explicit_na(medium), n = 5)) %>%
  mutate(browser = fct_lump(fct_explicit_na(browser), n = 4)) %>%
  mutate(operatingSystem = fct_lump(fct_explicit_na(operatingSystem), n = 2))
%>%
  group_by(custId) %>%
  summarize(
    channelGrouping = max(ifelse(is.na(channelGrouping) == TRUE, -9999,
channelGrouping)),
    maxVisitNum = max(visitNumber, na.rm = TRUE),
```

```
    browser = first(browser),
    operatingSystem = first(operatingSystem),
    country = first(country),
    medium = first(medium),
    isTrueDirect = mean(ifelse(is.na(isTrueDirect) == TRUE, 0, 1)),
    bounce_sessions = sum(ifelse(is.na(bounces) == TRUE, 0, 1)),
    pageviews_sum = sum(pageviews, na.rm = TRUE),
    pageviews_mean = mean(ifelse(is.na(pageviews), 0, pageviews)),
    pageviews_min = min(ifelse(is.na(pageviews), 0, pageviews)),
    pageviews_max = max(ifelse(is.na(pageviews), 0, pageviews)),
    pageviews_median = median(ifelse(is.na(pageviews), 0, pageviews)),

  )

targetRevenue<-Train %>%
  group_by(custId) %>%
  summarize(
    custRevenue = sum(revenue)
  ) %>%
  mutate(logSumRevenue = log(custRevenue+1)) %>%
dplyr::select(-custRevenue)
```

Missing value:In the data preparation process the first action we took was checking missing values and we removed the variables with more than 75 % of missing data and the rest numeric variables and non numeric variables are imputed and some values are just replaced with binary digits.
Reason: To predict we need data but if 75% is missing then the prediction may not be correct so for the accurate prediction it's better to remove those variables

Calculated the target revenue value: As suggested in the question we calculated the target revenue value using the formula and after imputing we summarised all the categorical variables as well as numeric variables ,and grouped them using the common column custID.
Reason: We calculated the target revenue and summarised in order to predict each customer will spend in total across all visits so so that's why we aggregated the revenue and found out the target revenue

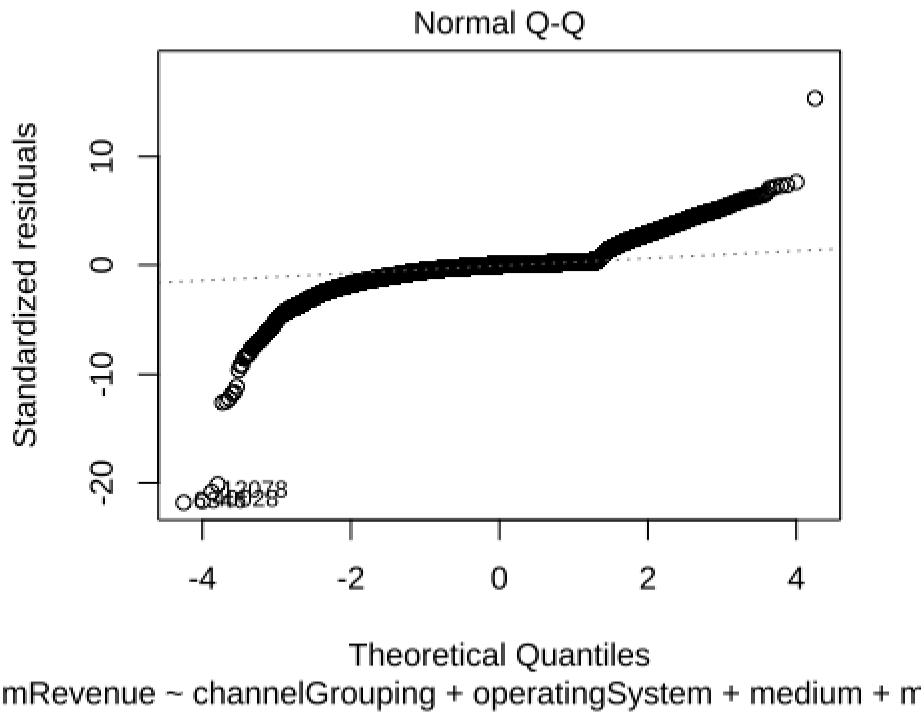
(iii) **Modelling**

Resampling approach: We used 10 fold cross validation

(a) OLS MODEL

```
Linearols<-lm(logSumRevenue ~
        channelGrouping + operatingSystem+medium+
        maxVisitNum+ browser +
        country +
        bounce_sessions + bounce_sessions*pageviews_sum +
        pageviews_sum +pageviews_mean + pageviews_min +
        pageviews_median,
       data = trainTransformed)
```

```
Coefficients:
                               Estimate Std. Error t value           Pr(>|t|)    
(Intercept)                   -1.399534   0.874212   -1.60              0.109    
channelGroupingAffiliates      1.233266   0.874365    1.41              0.158    
channelGroupingDirect          0.263148   0.861867    0.31              0.760    
channelGroupingDisplay        -0.311925   0.869502   -0.36              0.720    
channelGroupingOrganic Search  0.446754   0.862773    0.52              0.605    
channelGroupingPaid Search     0.319260   0.864468    0.37              0.712    
channelGroupingReferral        0.993173   0.862601    1.15              0.250    
channelGroupingSocial          0.596773   0.862724    0.69              0.489    
operatingSystemWindows        -0.188230   0.011666  -16.13 < 0.0000000000000002 ***
operatingSystemOther          -0.221036   0.010967  -20.15 < 0.0000000000000002 ***
mediumcpc                      0.982294   0.155667    6.31    0.000000000281099 ***
mediumcpm                      1.796682   0.178242   10.08 < 0.0000000000000002 ***
mediumorganic                  0.847242   0.143016    5.92    0.000000003161328 ***
mediumreferral                 0.769475   0.140748    5.47    0.000000045996459 ***
medium(Missing)                1.098445   0.144046    7.63    0.000000000000025 ***
maxVisitNum                    0.075425   0.002924   25.79 < 0.0000000000000002 ***
browserFirefox                -0.025386   0.020442   -1.24              0.214    
browserInternet Explorer       0.011547   0.026771    0.43              0.666    
browserSafari                 -0.118355   0.011896   -9.95 < 0.0000000000000002 ***
browserOther                  -0.016921   0.018640   -0.91              0.364    
countryCanada                 -0.060321   0.036814   -1.64              0.101    
countryFrance                 -0.054154   0.041536   -1.30              0.192    
countryGermany                -0.009453   0.039166   -0.24              0.809    
countryIndia                  -0.004528   0.031752   -0.14              0.887    
countryJapan                  -0.096535   0.039068   -2.47              0.013 *  
countryThailand               -0.001705   0.036977   -0.05              0.963    
countryTurkey                  0.020315   0.036987    0.55              0.583    
countryUnited Kingdom          0.019381   0.033631    0.58              0.564    
countryUnited States           0.268924   0.028403    9.47 < 0.0000000000000002 ***
countryVietnam                 0.015324   0.035426    0.43              0.665    
countryOther                  -0.020812   0.027768   -0.75              0.454    
bounce_sessions                0.009809   0.005092    1.93              0.054 .  
pageviews_sum                  0.009012   0.000323   27.88 < 0.0000000000000002 ***
pageviews_mean                 0.116258   0.002959   39.29 < 0.0000000000000002 ***
pageviews_min                 -0.065739   0.001550  -42.41 < 0.0000000000000002 ***
pageviews_median              -0.000271   0.002766   -0.10              0.922    
bounce_sessions:pageviews_sum -0.000444   0.000019  -23.35 < 0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.862 on 47212 degrees of freedom
Multiple R-squared:  0.567,	Adjusted R-squared:  0.567 
F-statistic: 1.72e+03 on 36 and 47212 DF,  p-value: <0.0000000000000002
```



(b) MARS MODEL

```
marsFit <- earth(logSumRevenue ~
                 channelGrouping + operatingSystem+medium+
                 maxVisitNum+ browser +
                 country +
                 bounce_sessions + bounce_sessions*pageviews_sum +
                   pageviews_sum +pageviews_mean + pageviews_min +
                     pageviews_median,
               data = trainTransformed,
               degree=2,nk=49,pmethod="cv",nfold=10,ncross=10)
```

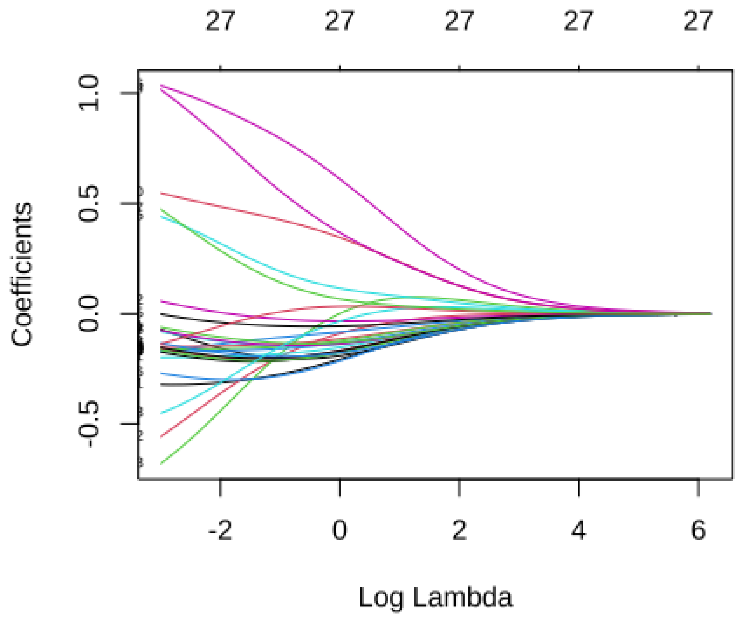
```
> summary(marsFit)
Call: earth(formula=logSumRevenue~channelGrouping+browser+country+medi...),
            data=trainTransformed, pmethod="cv", degree=2, nfold=5, ncross=5, nk=49)

                                                      coefficients
(Intercept)                                                 0.0248
countryCanada                                               0.2698
countryUnited States                                        0.9922
channelGroupingAffiliates * countryUnited States          -0.8949
channelGroupingDirect * countryUnited States              -0.7654
channelGroupingOrganic Search * countryUnited States      -0.3018
channelGroupingReferral * countryUnited States             1.7181
browserSafari * countryUnited States                      -0.4129
browserOther * countryUnited States                       -0.4623
countryUnited States * mediumreferral                     -0.6844
countryUnited States * medium(Missing)                     0.5963

Selected 11 of 12 terms, and 10 of 27 predictors (pmethod="cv")
Termination condition: RSq changed by less than 0.001 at 12 terms
Importance: channelGroupingReferral, countryUnited States, mediumreferral, ...
Number of terms at each degree of interaction: 1 2 8
GRSq 0.259  RSq 0.259  mean.oof.RSq 0.258 (sd 0.00951)

pmethod="backward" would have selected:
    12 terms 10 preds,  GRSq 0.259  RSq 0.26  mean.oof.RSq 0.257
```

(c) RIDGE REGRESSION MODEL

```
set.seed(1234)
rR <- train(logSumRevenue ~
            channelGrouping + operatingSystem+medium+
            maxVisitNum+ browser +
            country +
            bounce_sessions + bounce_sessions*pageviews_sum +
            pageviews_sum +pageviews_mean + pageviews_min +
            pageviews_median,
          data = trainTransformed, method = 'glmnet',
            tuneGrid = expand.grid(alpha = 0, lambda = 0.0001), trControl =
custom)
```

	alpha	lambda	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
1	0	1e-05	0.8761790	0.5551877	0.4652064	0.02996307	0.03007052	0.009671122
2	0	1e-04	0.8761790	0.5551877	0.4652064	0.02996307	0.03007052	0.009671122
3	0	1e-03	0.8761790	0.5551877	0.4652064	0.02996307	0.03007052	0.009671122
4	0	1e-02	0.8761790	0.5551877	0.4652064	0.02996307	0.03007052	0.009671122
5	1	1e-05	0.8703425	0.5608932	0.4555396	0.03036505	0.02951952	0.009385059
6	1	1e-04	0.8703425	0.5608932	0.4555396	0.03036505	0.02951952	0.009385059
7	1	1e-03	0.8700971	0.5611146	0.4555719	0.03029608	0.02951690	0.009418793
8	1	1e-02	0.8718133	0.5595532	0.4569425	0.02977725	0.03022431	0.009466733

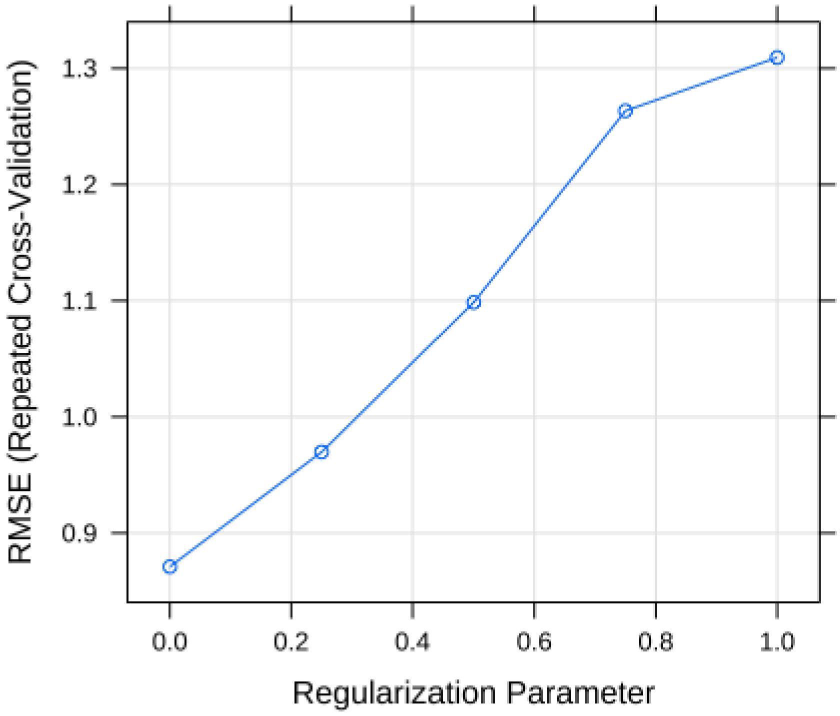
>



(d) LASSO REGRESSION MODEL

```
set.seed(1234)
lassoModel <- train(logSumRevenue ~
                channelGrouping + operatingSystem+medium+
                log(maxVisitNum+1) + browser +
                country+
                log(bounce_sessions+1) + bounce_sessions*pageviews_sum +(
                  log(pageviews_sum+1) + log(pageviews_mean+1) +
pageviews_min +
                    pageviews_median),
              data = trainTransformed, method = 'glmnet',
              tuneGrid = expand.grid(alpha = 1, lambda =
seq(0.0001,1,length=5)), trControl = custom)
```

```
> lassoModel$results
  alpha lambda  RMSE Rsquared   MAE RMSESD RsquaredSD   MAESD
1     1 0.0001 0.871    0.560 0.456 0.0380     0.0362 0.00951
2     1 0.2501 0.970    0.508 0.519 0.0238     0.0481 0.00938
3     1 0.5000 1.099    0.481 0.636 0.0232     0.0528 0.01092
4     1 0.7500 1.263    0.415 0.749 0.0285     0.0620 0.01294
5     1 1.0000 1.309      NaN 0.776 0.0269         NA 0.01159
```

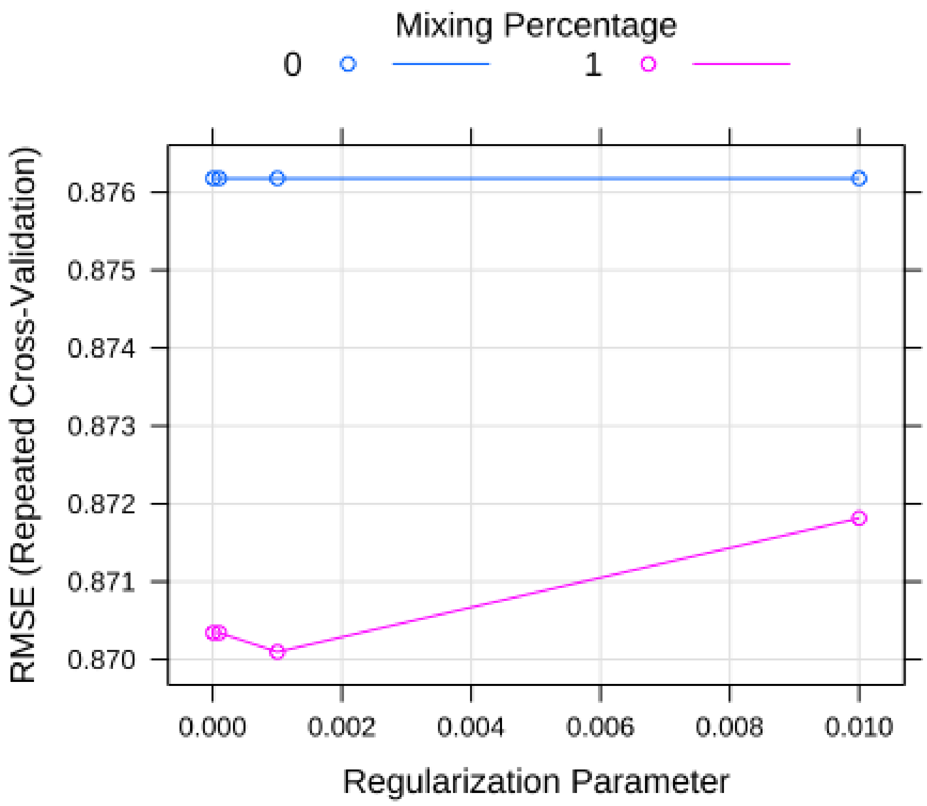


Model	Method	Package	Hyperparameter	value	CV RMSE	R^2
OLS	Ols	lm	NA	NA	0.9	0.582
Lasso	glmnet	glmnet	fraction	0.0001	0.878	0.415
RIDGE	glmnet	glmnet	fraction	0.0001	0.876	0.556
Mars	cv	earth	degree	2	0.89	0.584

(iv)
Modelling Approach- Ridge Regression worked best for us. We used lambda as the tuning parameter.We got the lowest RMSE value.The approach we used is we took 10 fold resampling.This model is used as a technique which is specialised to analyse multiple regression data which is multicollinearity in nature. We use Ridge Regression to create a parsimonious model .The below code is used to find the relationship between RMSE and Regularisation Parameter and selecting the best alpha value and Lamda value.
Variables used- channelGrouping, operatingSystem,medium, maxVisitNum,browser, country, pageviews

```
glmnet

47249 samples
   11 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 42524, 42525, 42524, 42524, 42524, 42524, ...
Resampling results:

  RMSE       Rsquared   MAE
  0.8760678  0.5560196  0.4652521

Tuning parameter 'alpha' was held constant at a value of 0
Tuning parameter 'lambda'
 was held constant at a value of 1e-04
>
```

```
set.seed(1234)
rR <- train(logSumRevenue ~
          channelGrouping + operatingSystem+medium+
          maxVisitNum+ browser +
          country +
          bounce_sessions + bounce_sessions*pageviews_sum +
          pageviews_sum +pageviews_mean + pageviews_min +
          pageviews_median,
        data = trainTransformed, method = 'glmnet',
          tuneGrid = expand.grid(alpha = 0:1, lambda = c(0.0001, 0.001,
0.01)), trControl = custom)
rR$finalModel
rR$results
plot(rR)
```



The above graph shows the relationship between RMSE(Repeated Cross-Validation) and Regularisation Parameter.