CAPSTONE PROJECT
PREDICT RETAIL CUSTOMER BEHAVIOR

Karunakaran Palaniswamy

PGD-BA&I, Mar 2019 Batch

OVERVIEW – PREDICT RETAIL CUSTOMER BEHAVIOR

- Study of customer **buying behavior** is most **important** for retail stores as they can understand the expectation of the **customers**. It helps to understand what makes a **customer** to buy a product.
- Every Retailer needs to assess the customer behavior using the Sales data they have on the below areas:
  - How frequently they visit?
  - How much they spend during every visit?
  - How many times they have visited the store in a particular period?
  - What kind of products they purchase?
- Through this assignment, the below are done:
  - Different Retail Performance KPIs are calculated to understand about Sales, and customers status, etc
  - Customer Segmentation to focus on specific customers to bring them to the store more often
  - Predicting the Life Time value of customers to see what can be done for them
  - Predicting the next purchase day of customers to see what can be done for them

PROJECT MODULES

- The project work is divided in to 5 modules as follows:

1. Basic operations with the dataset
   - Identification and removal of data with NA, conversion of date column (s) to another to support the process, renaming of columns, if required
2. Identification of Retail Performance KPIs - To understand the Customer behaviour
   - Overall revenue, Revenue Growth Rate, Active Customers, Purchase (orders), Average Revenue per Purchase, New/Existing Customer Ratio, and Retention Rate of customers on a monthly basis.
3. Customer Segmentation
   - Identification of customers' recent purchase (Recency) pattern, their frequent (Frequency) trips to stores, and the money (Monetary) they spent. Basically it is RFM.
4. Prediction of Customer's Life Time Value (LTV)
   - To focus on the potential customers who can bring more revenue in the future
5. Prediction of Next Purchase Day
   - To focus further on the planning to maximize the customers' experience and purchase

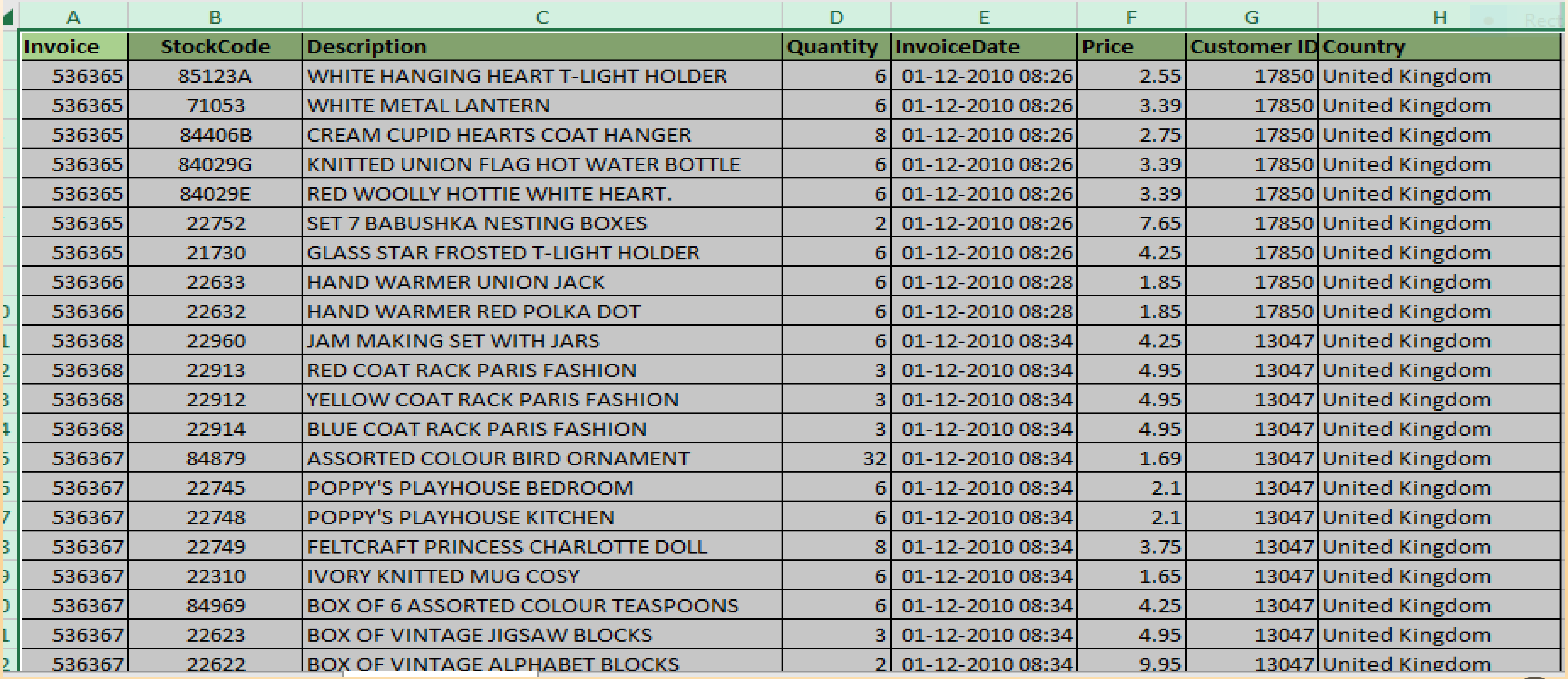
1. BASIC OPERATIONS ON THE DATASET

1. BASIC OPERATIONS

- Dataset used: online_retail_II.xlsx , read data from Year 2010-11 tab.
- Basic operations with the dataset
  - Read the input file
  - Changing the column name of “Customer ID” to “Customer_ID” to avoid any confusion in the coding and execution
  - Display top 5 records and bottom 5.
  - Display the structure of the data to identify numeric, character, text fields and do the necessary conversion
  - Find the NA's in the data and omit them from the analysis
  - Create additional column for the Date field to support the visualization

```
### Change the working directory in the code below before running the code

Input file folder

setwd("C:/Karun/Personal/Amity/Capstone Project")

#Read the input file

Sales_Data <- read_xlsx("online_retail_II.xlsx",sheet = "Year 2010-2011")
```

RETAIL SALES DATA - SNAPSHOT



A	B	C	D	E	F	G	H
Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	01-12-2010 08:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	01-12-2010 08:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UNION JACK	6	01-12-2010 08:28	1.85	17850	United Kingdom
536366	22632	HAND WARMER RED POLKA DOT	6	01-12-2010 08:28	1.85	17850	United Kingdom
536368	22960	JAM MAKING SET WITH JARS	6	01-12-2010 08:34	4.25	13047	United Kingdom
536368	22913	RED COAT RACK PARIS FASHION	3	01-12-2010 08:34	4.95	13047	United Kingdom
536368	22912	YELLOW COAT RACK PARIS FASHION	3	01-12-2010 08:34	4.95	13047	United Kingdom
536368	22914	BLUE COAT RACK PARIS FASHION	3	01-12-2010 08:34	4.95	13047	United Kingdom
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	01-12-2010 08:34	1.69	13047	United Kingdom
536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	01-12-2010 08:34	2.1	13047	United Kingdom
536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	01-12-2010 08:34	2.1	13047	United Kingdom
536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	01-12-2010 08:34	3.75	13047	United Kingdom
536367	22310	IVORY KNITTED MUG COSY	6	01-12-2010 08:34	1.65	13047	United Kingdom
536367	84969	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	01-12-2010 08:34	4.25	13047	United Kingdom
536367	22623	BOX OF VINTAGE JIGSAW BLOCKS	3	01-12-2010 08:34	4.95	13047	United Kingdom
536367	22622	BOX OF VINTAGE ALPHABET BLOCKS	2	01-12-2010 08:34	9.95	13047	United Kingdom

1. BASIC OPERATIONS

```
> str(Sales_Data)   # There are 8 variables overall

Classes ‘tbl_df’, ‘tbl’ and 'data.frame':   541910 obs. of  8 variables:

 $ Invoice     : chr  "536365" "536365" "536365" "536365" ...

 $ StockCode  : chr  "85123A" "71053" "84406B" "84029G" ...

 $ Description: chr  "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM CUPID HEARTS COAT HANGER"
"KNITTED UNION FLAG HOT WATER BOTTLE" ...

 $ Quantity    : num  6 6 8 6 6 2 6 6 6 6 ...

 $ InvoiceDate: POSIXct, format: "2010-12-01 08:26:00" "2010-12-01 08:26:00" "2010-12-01 08:26:00" "2010-12-01 08:26:00" ...

 $ Price       : num  2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 4.25 ...

 $ Customer_ID: num  17850 17850 17850 17850 17850 ...

 $ Country     : chr  "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" ...
```

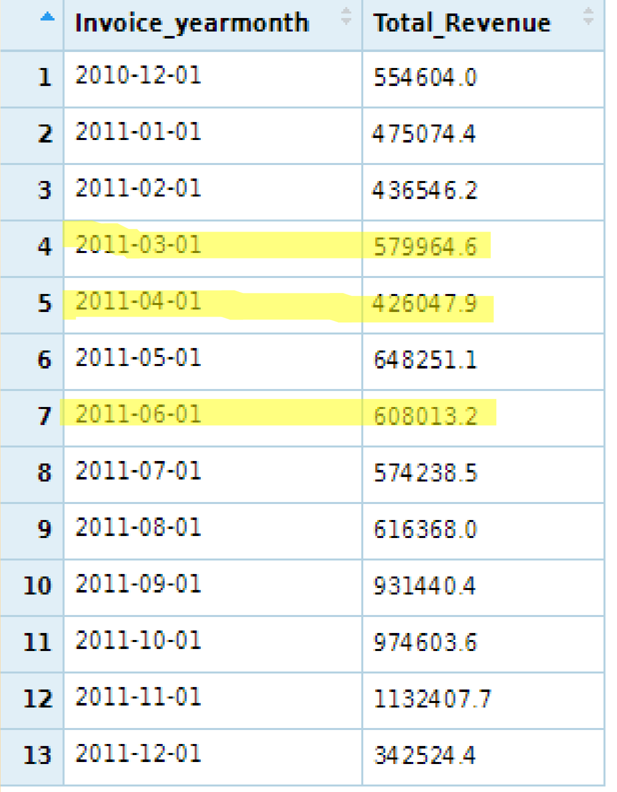
1. BASIC OPERATIONS

```
> head(Sales_Data)
Registered S3 method overwritten by 'cli':   method      from   print.tree tree
# A tibble: 6 x 8
  Invoice StockCode Description                      Quantity InvoiceDate          Price Customer_ID Country
  <chr>   <chr>     <chr>                               <dbl> <dttm>               <dbl>       <dbl> <chr>
1 536365  85123A    WHITE HANGING HEART T-LIGHT HO~         6 2010-12-01 08:26:00  2.55        17850 United King~
2 536365  71053     WHITE METAL LANTERN                     6 2010-12-01 08:26:00  3.39        17850 United King~
3 536365  84406B    CREAM CUPID HEARTS COAT HANGER          8 2010-12-01 08:26:00  2.75        17850 United King~
4 536365  84029G    KNITTED UNION FLAG HOT WATER B~         6 2010-12-01 08:26:00  3.39        17850 United King~
5 536365  84029E    RED WOOLLY HOTTIE WHITE HEART.          6 2010-12-01 08:26:00  3.39        17850 United King~
6 536365  22752     SET 7 BABUSHKA NESTING BOXES            2 2010-12-01 08:26:00  7.65        17850 United King~
```

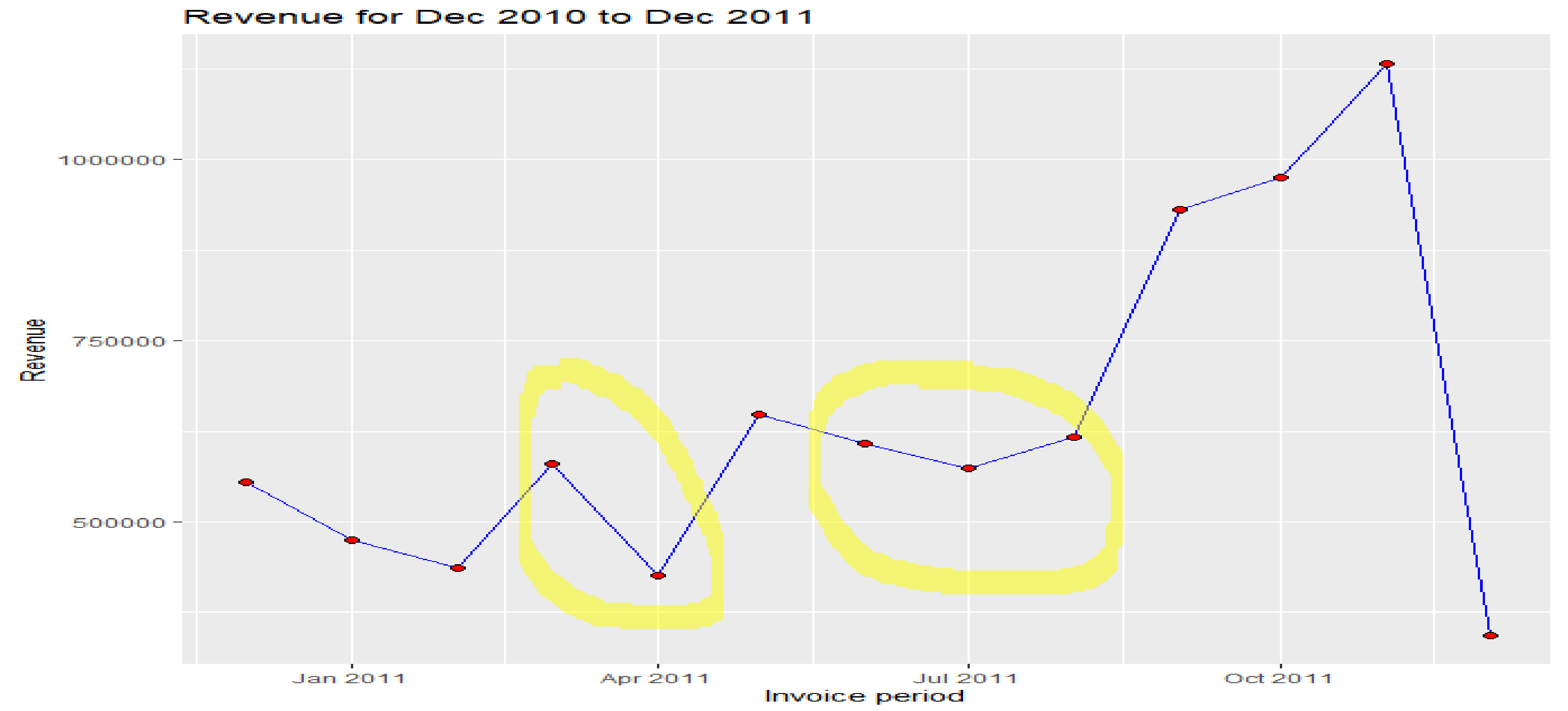
II. RETAIL PERFORMANCE KPIS

11. RETAIL PERFORMANCE KPIS

- **Overall Revenue**



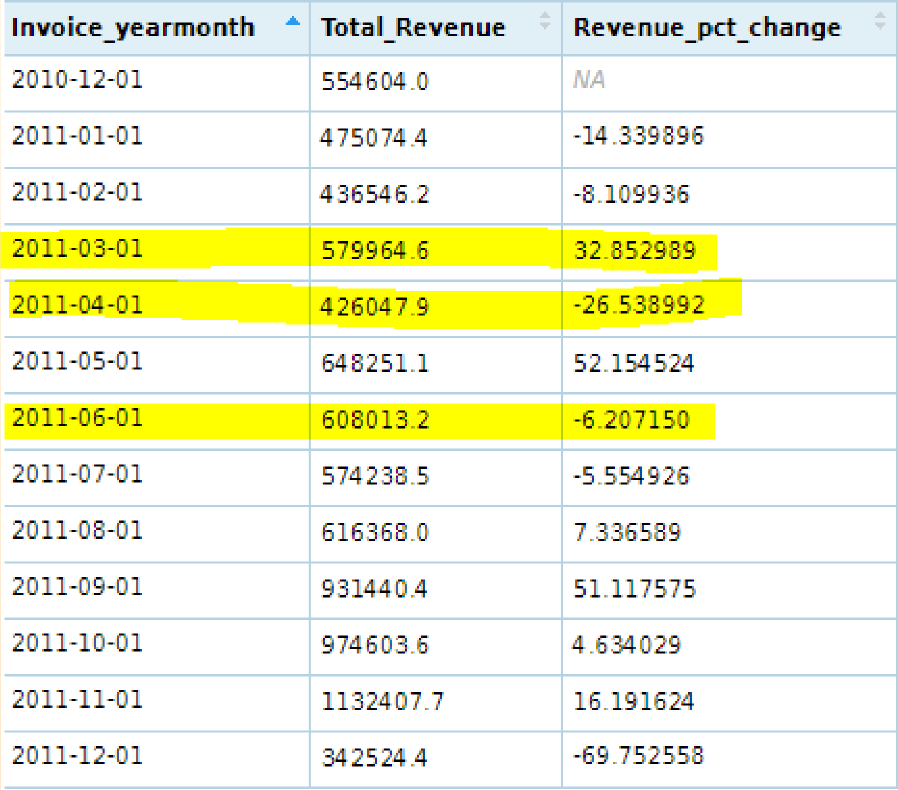
	Invoice_yearmonth	Total_Revenue
1	2010-12-01	554604.0
2	2011-01-01	475074.4
3	2011-02-01	436546.2
4	2011-03-01	579964.6
5	2011-04-01	426047.9
6	2011-05-01	648251.1
7	2011-06-01	608013.2
8	2011-07-01	574238.5
9	2011-08-01	616368.0
10	2011-09-01	931440.4
11	2011-10-01	974603.6
12	2011-11-01	1132407.7
13	2011-12-01	342524.4



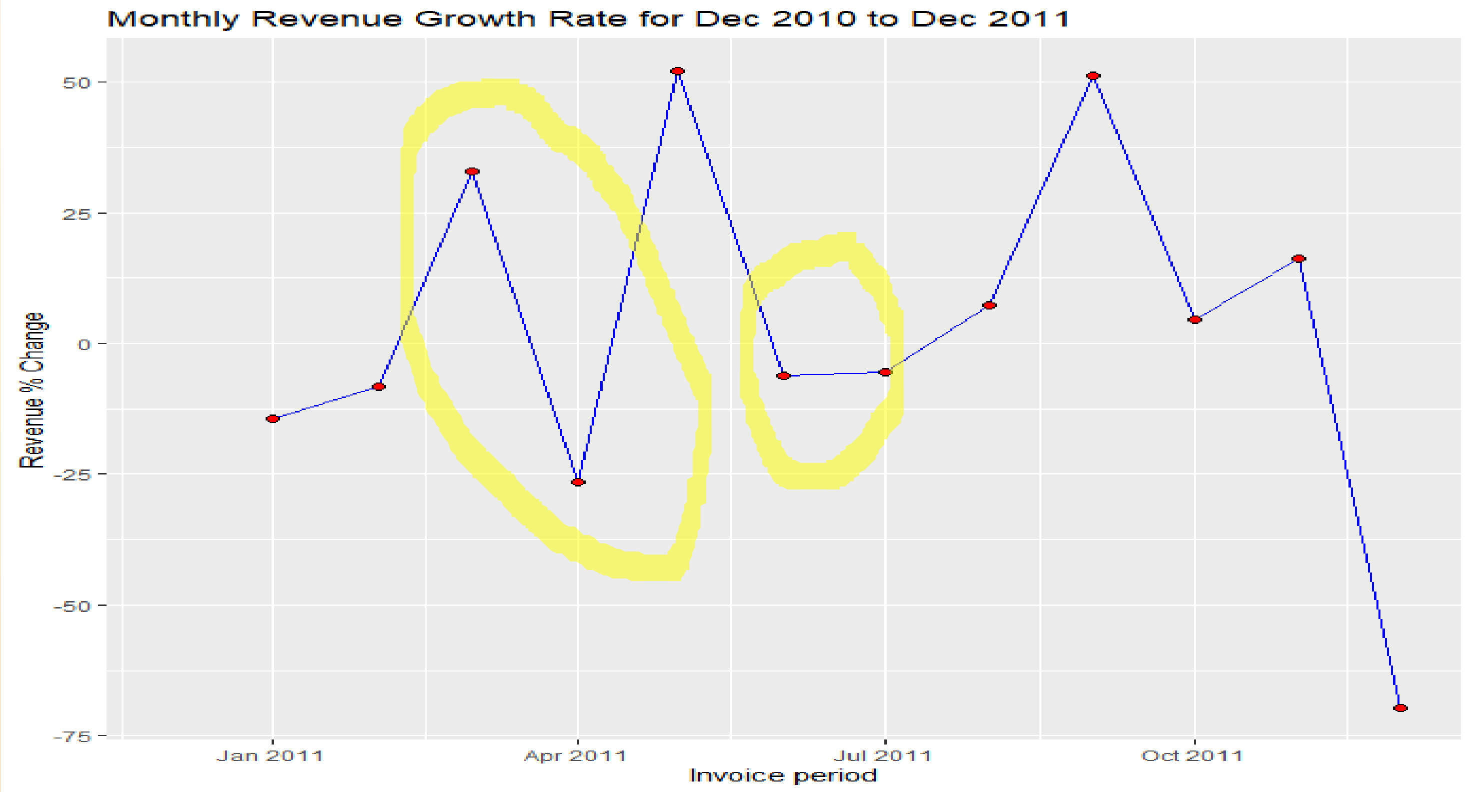
Inference: The above shows there is a good growth over the period but there is a drop in Apr 2011 and also in June

II. RETAIL PERFORMANCE KPIS

- **Monthly Revenue Growth Rate**



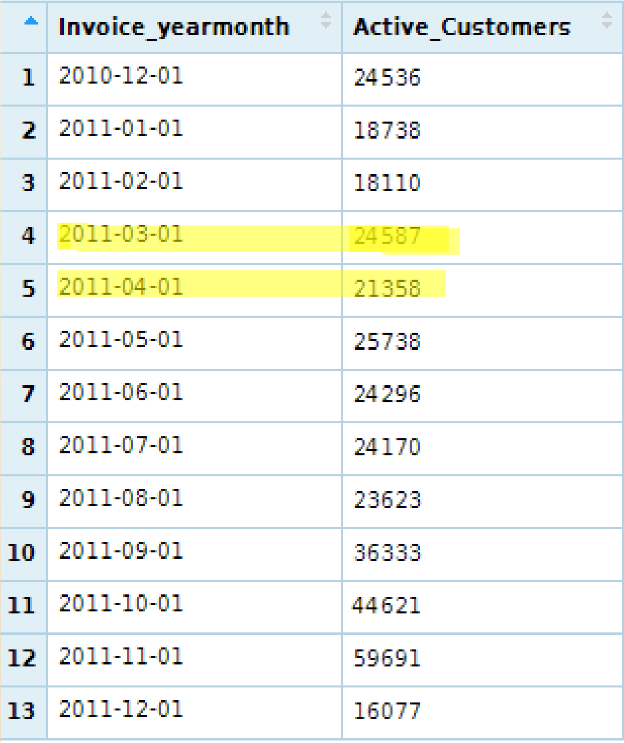
Invoice_yearmonth	Total_Revenue	Revenue_pct_change
2010-12-01	554604.0	NA
2011-01-01	475074.4	-14.339896
2011-02-01	436546.2	-8.109936
2011-03-01	579964.6	32.852989
2011-04-01	426047.9	-26.538992
2011-05-01	648251.1	52.154524
2011-06-01	608013.2	-6.207150
2011-07-01	574238.5	-5.554926
2011-08-01	616368.0	7.336589
2011-09-01	931440.4	51.117575
2011-10-01	974603.6	4.634029
2011-11-01	1132407.7	16.191624
2011-12-01	342524.4	-69.752558



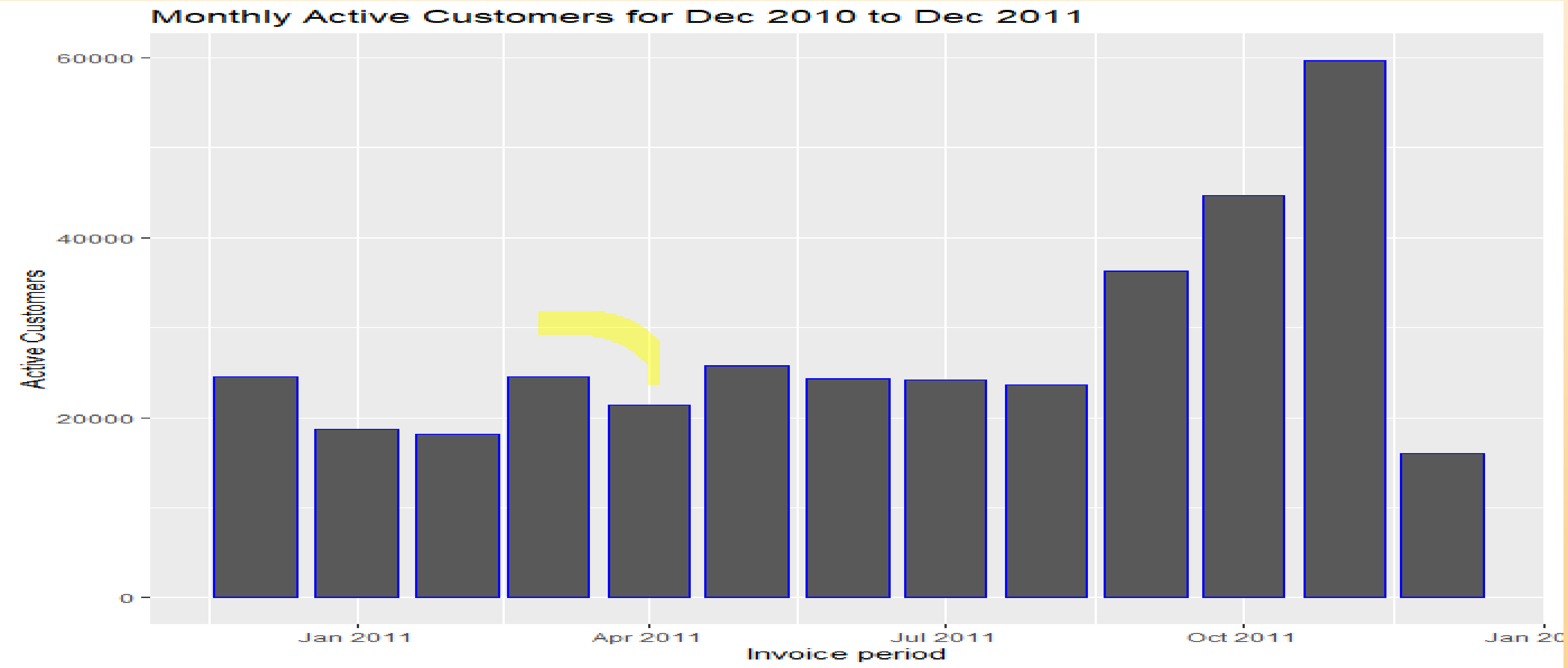
Inference: The above shows there is a good growth over the period but there is a drop in Apr 2011 and also in June

11. RETAIL PERFORMANCE KPIS

- **Monthly Active Customers –** Use the data from country "United Kingdom". Focusing one country data will help deeply analysing and predicting issues.



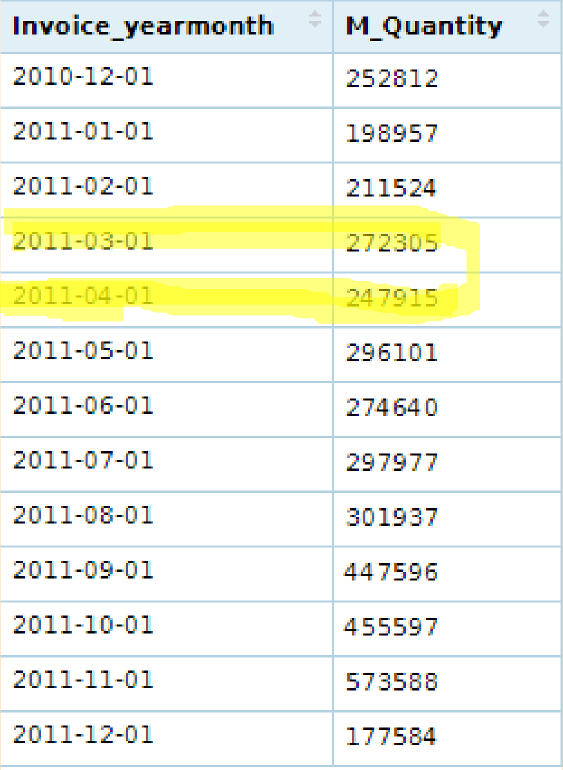
	Invoice_yearmonth	Active_Customers
1	2010-12-01	24536
2	2011-01-01	18738
3	2011-02-01	18110
4	2011-03-01	24587
5	2011-04-01	21358
6	2011-05-01	25738
7	2011-06-01	24296
8	2011-07-01	24170
9	2011-08-01	23623
10	2011-09-01	36333
11	2011-10-01	44621
12	2011-11-01	59691
13	2011-12-01	16077



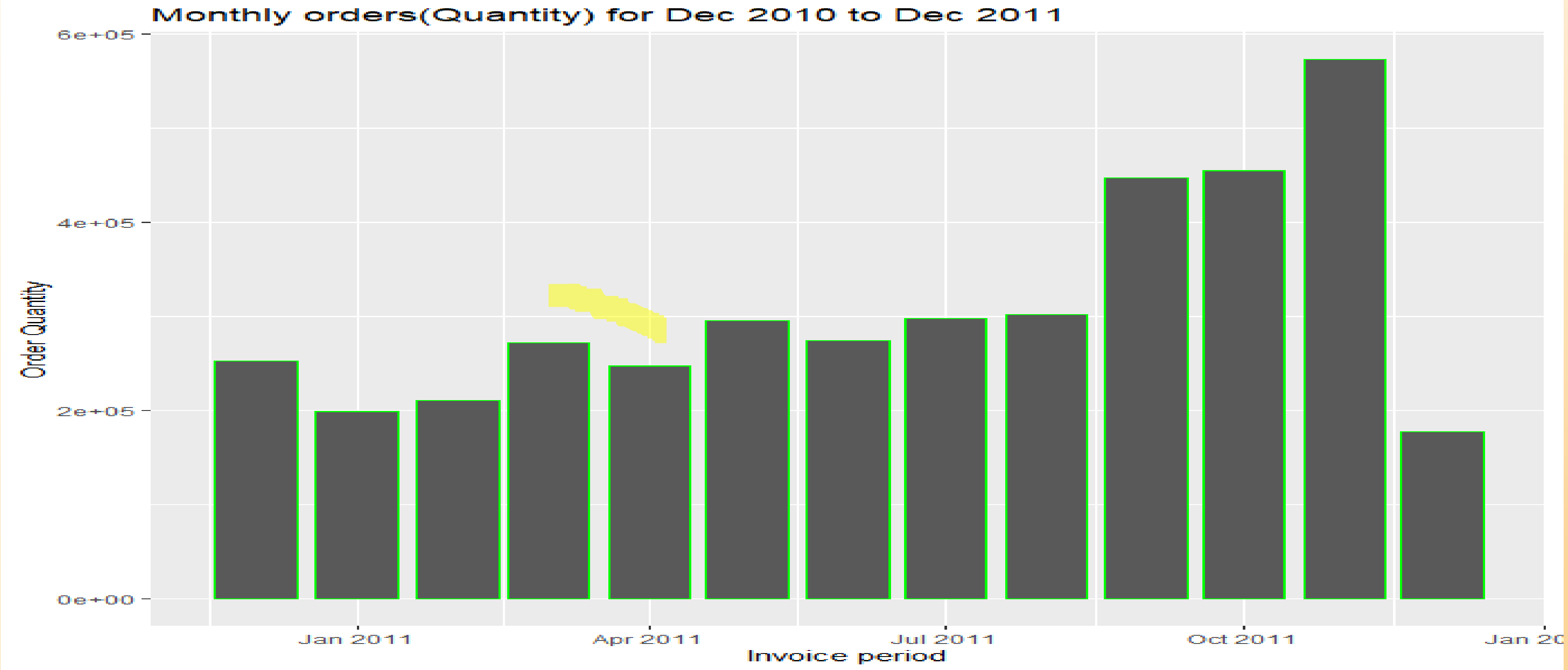
Inference: In Apr 2011, no. of customers fell 13% from 24,587 to 21,358

II. RETAIL PERFORMANCE KPIS

- **Monthly Orders (or Purchase Quantity)**



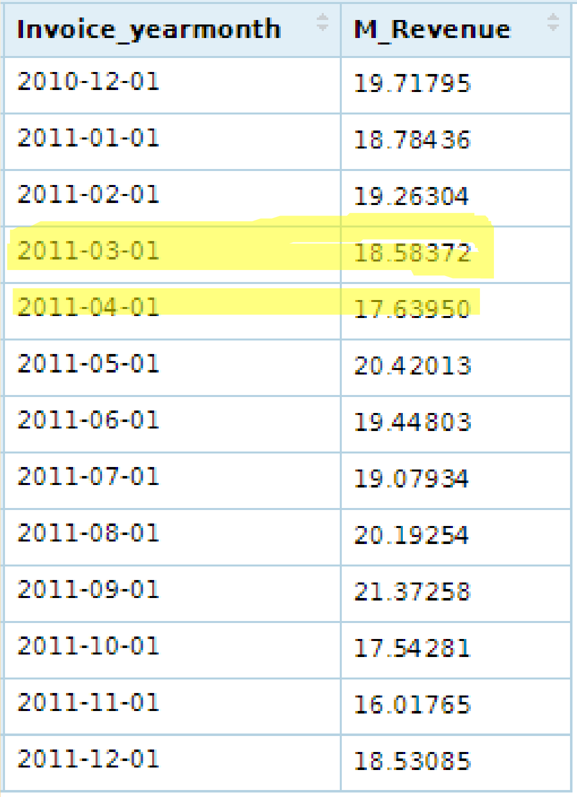
Invoice_yearmonth	M_Quantity
2010-12-01	252812
2011-01-01	198957
2011-02-01	211524
2011-03-01	272305
2011-04-01	247915
2011-05-01	296101
2011-06-01	274640
2011-07-01	297977
2011-08-01	301937
2011-09-01	447596
2011-10-01	455597
2011-11-01	573588
2011-12-01	177584



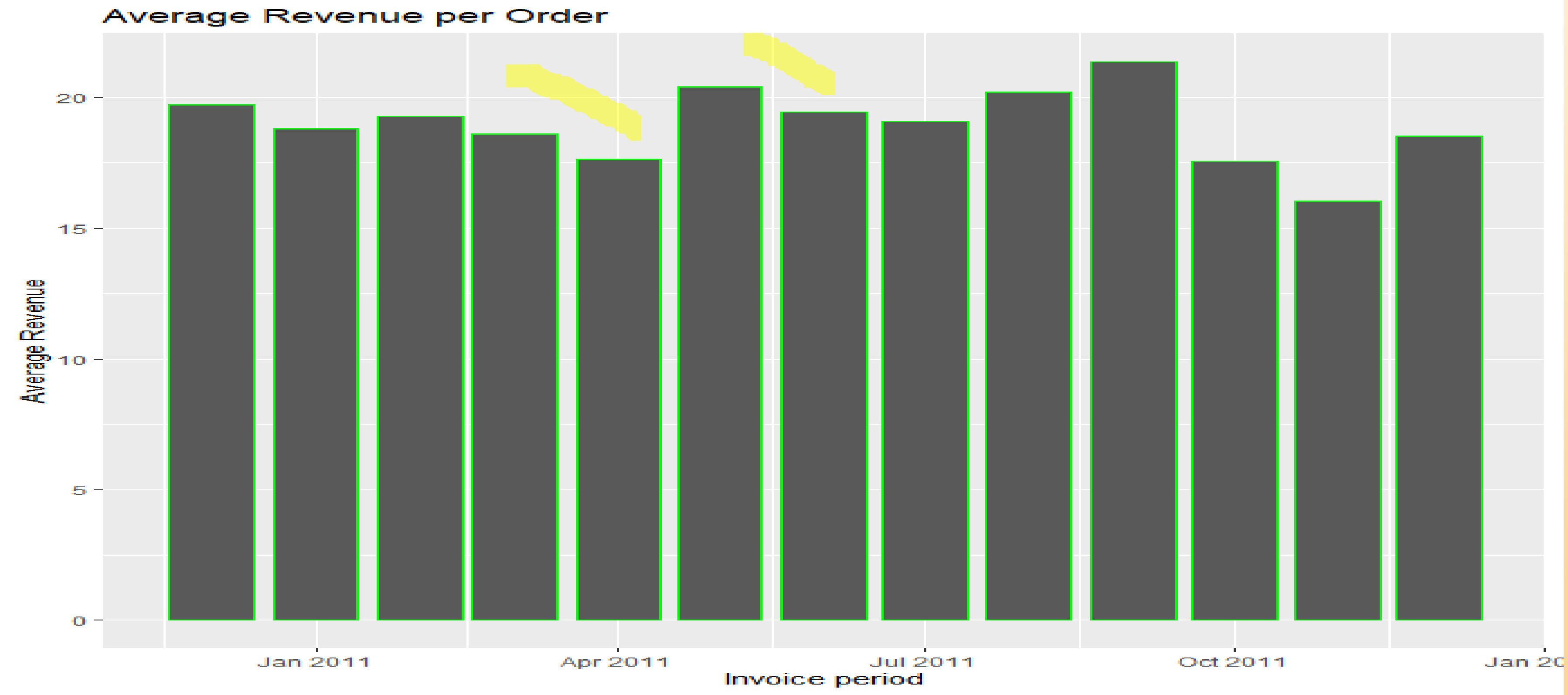
Inference: Between March and April 2011, the total quantity has come down from 272,305 to 247,915, that is almost 9%. This could be the impact of decrease in Active Customer Count

11. RETAIL PERFORMANCE KPIS

- **Average Revenue per Order – What is the average total amount of spent during every purchase?**



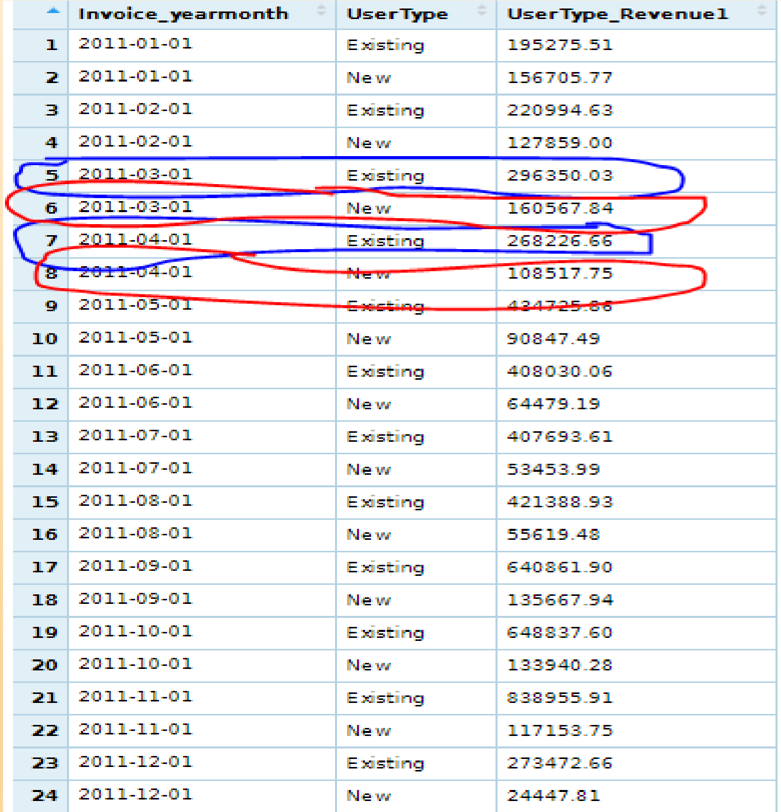
Invoice_yearmonth	M_Revenue
2010-12-01	19.71795
2011-01-01	18.78436
2011-02-01	19.26304
2011-03-01	18.58372
2011-04-01	17.63950
2011-05-01	20.42013
2011-06-01	19.44803
2011-07-01	19.07934
2011-08-01	20.19254
2011-09-01	21.37258
2011-10-01	17.54281
2011-11-01	16.01765
2011-12-01	18.53085



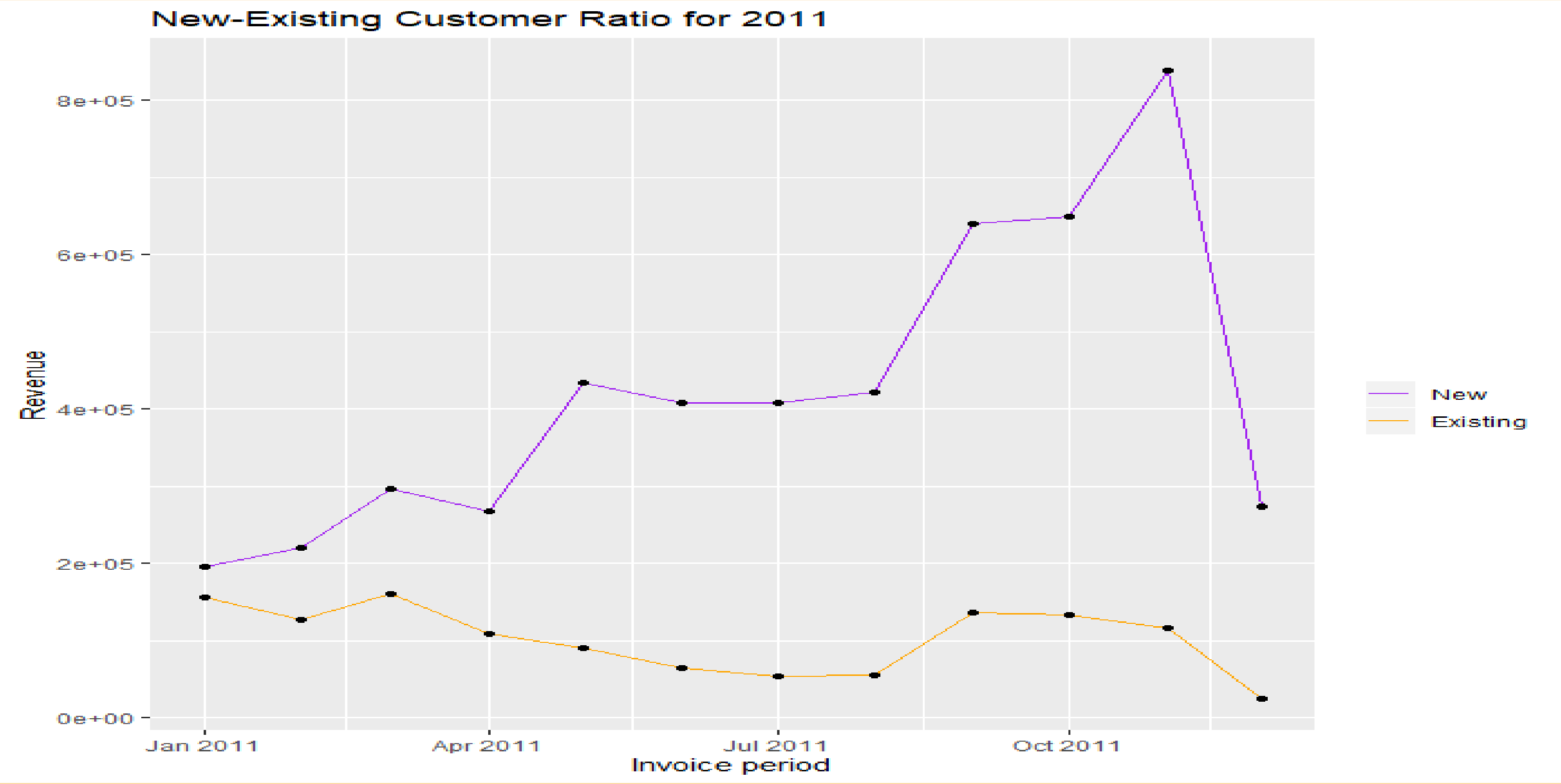
Inference: In the above, the average revenue has come down from 18.59 to 17.64 i.e 5.1%. Here we are seeing slowdown in the overall retail.

11. RETAIL PERFORMANCE KPIS

- **New/Existing Customer Ratio –** What is the trend of New and Existing customer? Since we have only one month data from 2010, remove it and keep only 2011 data.



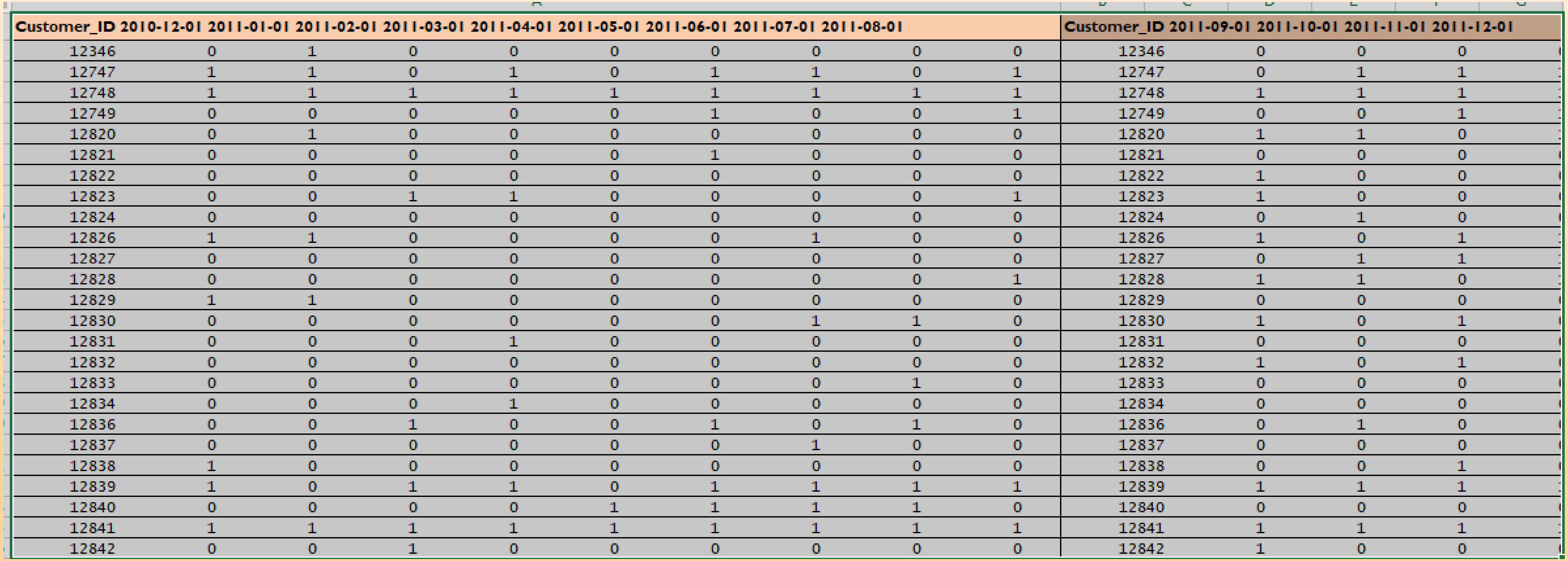
	Invoice_yearmonth	UserType	UserType_Revenue1
1	2011-01-01	Existing	195275.51
2	2011-01-01	New	156705.77
3	2011-02-01	Existing	220994.63
4	2011-02-01	New	127859.00
5	2011-03-01	Existing	296350.03
6	2011-03-01	New	160567.84
7	2011-04-01	Existing	268226.66
8	2011-04-01	New	108517.75
9	2011-05-01	Existing	434725.86
10	2011-05-01	New	90847.49
11	2011-06-01	Existing	408030.06
12	2011-06-01	New	64479.19
13	2011-07-01	Existing	407693.61
14	2011-07-01	New	53453.99
15	2011-08-01	Existing	421388.93
16	2011-08-01	New	55619.48
17	2011-09-01	Existing	640861.90
18	2011-09-01	New	135667.94
19	2011-10-01	Existing	648837.60
20	2011-10-01	New	133940.28
21	2011-11-01	Existing	838955.91
22	2011-11-01	New	117153.75
23	2011-12-01	Existing	273472.66
24	2011-12-01	New	24447.81



Inference: Both New and Existing customers totals are showing negative trend in Apr 2011

11. RETAIL PERFORMANCE KPIS

- **Monthly Retention Rate–** Indicates the status of the customer's membership with the store and it helps to understand their satisfaction.



Customer_ID	2010-12-01	2011-01-01	2011-02-01	2011-03-01	2011-04-01	2011-05-01	2011-06-01	2011-07-01	2011-08-01	2011-09-01	2011-10-01	2011-11-01	2011-12-01
12346	0	1	0	0	0	0	0	0	0	0	0	0	0
12747	1	1	0	1	0	1	1	0	1	0	1	1	1
12748	1	1	1	1	1	1	1	1	1	1	1	1	1
12749	0	0	0	0	0	1	0	0	1	0	0	1	1
12820	0	1	0	0	0	0	0	0	0	1	1	0	1
12821	0	0	0	0	0	1	0	0	0	0	0	0	0
12822	0	0	0	0	0	0	0	0	0	1	0	0	0
12823	0	0	1	1	0	0	0	0	1	1	0	0	0
12824	0	0	0	0	0	0	0	0	0	0	1	0	0
12826	1	1	0	0	0	0	1	0	0	1	0	1	1
12827	0	0	0	0	0	0	0	0	0	0	1	1	1
12828	0	0	0	0	0	0	0	0	1	1	1	0	1
12829	1	1	0	0	0	0	0	0	0	0	0	0	0
12830	0	0	0	0	0	0	1	1	0	1	0	1	0
12831	0	0	0	1	0	0	0	0	0	0	0	0	0
12832	0	0	0	0	0	0	0	0	0	1	0	1	0
12833	0	0	0	0	0	0	0	1	0	0	0	0	0
12834	0	0	0	1	0	0	0	0	0	0	0	0	0
12836	0	0	1	0	0	1	0	1	0	0	1	0	0
12837	0	0	0	0	0	0	1	0	0	0	0	0	0
12838	1	0	0	0	0	0	0	0	0	0	0	1	0
12839	1	0	1	1	0	1	1	1	1	1	1	1	1
12840	0	0	0	0	1	1	1	1	0	0	0	0	0
12841	1	1	1	1	1	1	1	1	1	1	1	1	1
12842	0	0	1	0	0	0	0	0	0	1	0	0	0

Inference: The R language doesn't have any command to display the above data in a proper way, hence formatted it.
In above cells, 0 indicates, the customer is not a member of the retail store. 1 indicates otherwise.

III - CUSTOMER SEGMENTATION

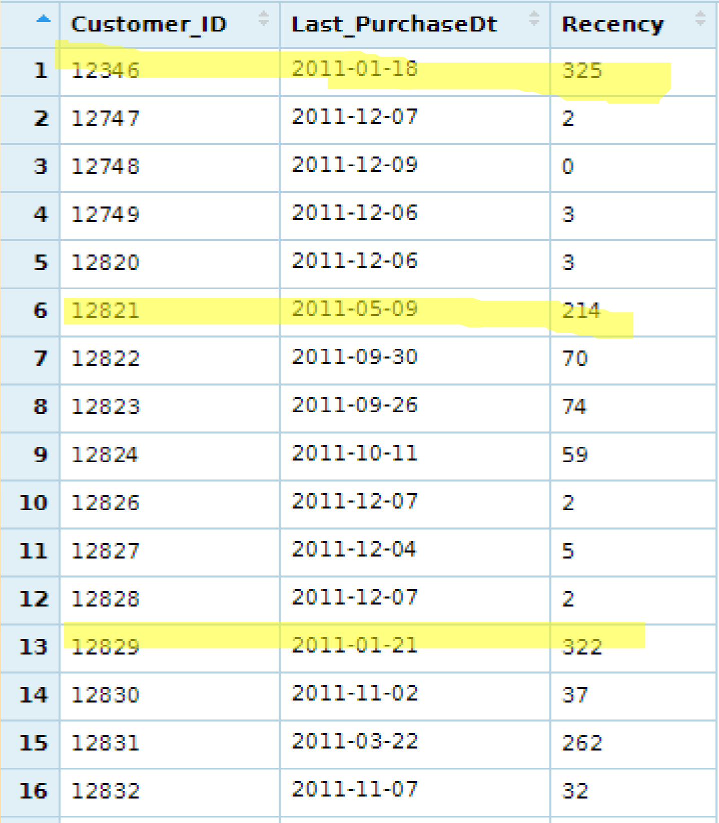
III - CUSTOMER SEGMENTATION

- **Why Segmentation is necessary?**
  - Better matching of customer needs
  - Enhanced profits for business
  - Better opportunities for growth
  - Retain more customers
  - Target marketing communications
  - Gain share of the market segment
- **Recency Frequency Monetary (RFM) –** By determining RFM, we can segment customers into the below categories.

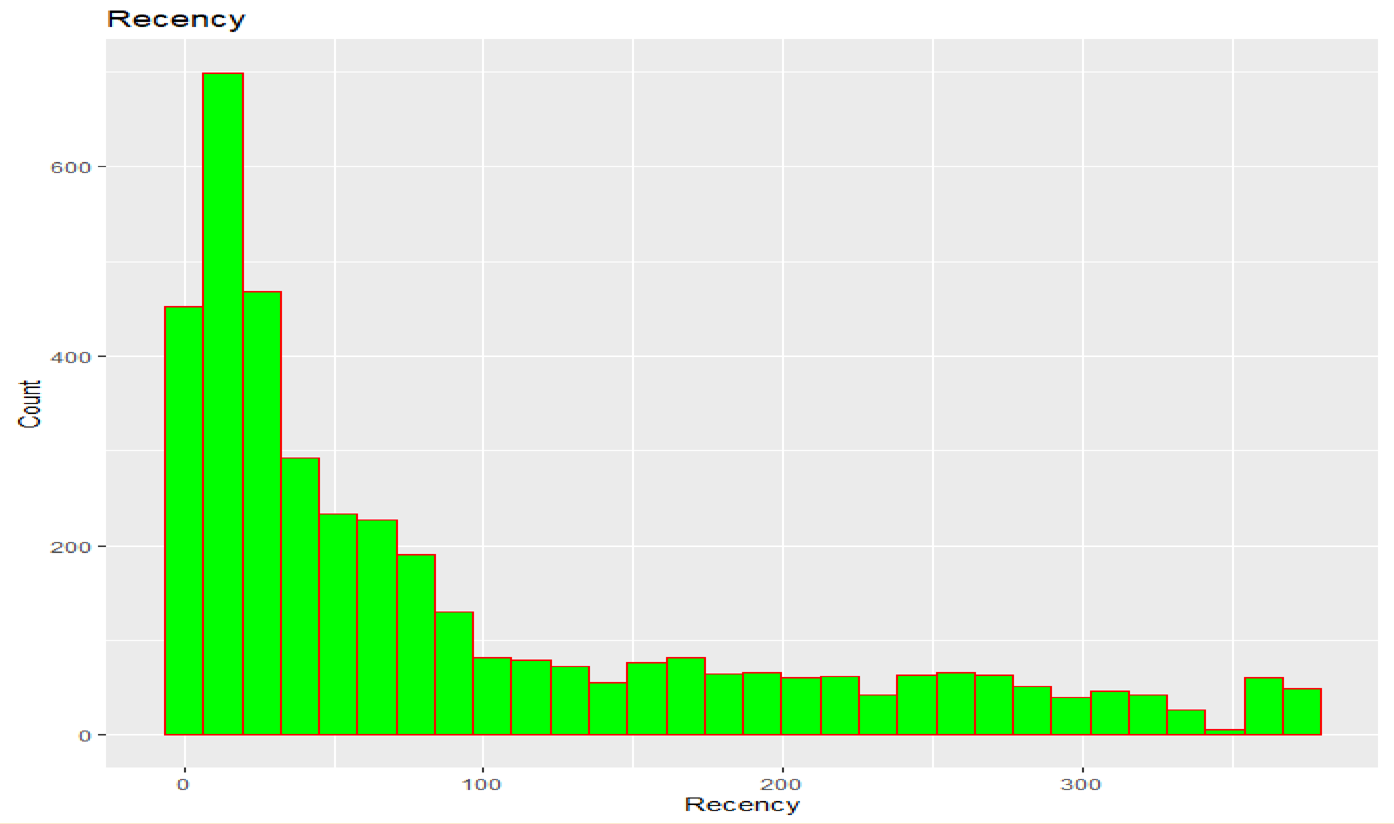
1. **<u>Low Value:</u>**
   - Customers who are less active than others, not very frequent buyer/visitor and generates very low – zero.
   - May be negative revenue.
2. **<u>Mid Value:</u>**
   - In the middle of everything. Often using our platform (but not as much as our High Values), fairly frequent and generates moderate revenue.
3. **<u>High Value:</u>**
   - The group we don't want to lose. High Revenue, Frequency and low Inactivity.

III - CUSTOMER SEGMENTATION

- **RECENCY -** find out most recent purchase date and find out the recency in number of days for every customer



	Customer_ID	Last_PurchaseDt	Recency
1	12346	2011-01-18	325
2	12747	2011-12-07	2
3	12748	2011-12-09	0
4	12749	2011-12-06	3
5	12820	2011-12-06	3
6	12821	2011-05-09	214
7	12822	2011-09-30	70
8	12823	2011-09-26	74
9	12824	2011-10-11	59
10	12826	2011-12-07	2
11	12827	2011-12-04	5
12	12828	2011-12-07	2
13	12829	2011-01-21	322
14	12830	2011-11-02	37
15	12831	2011-03-22	262
16	12832	2011-11-07	32

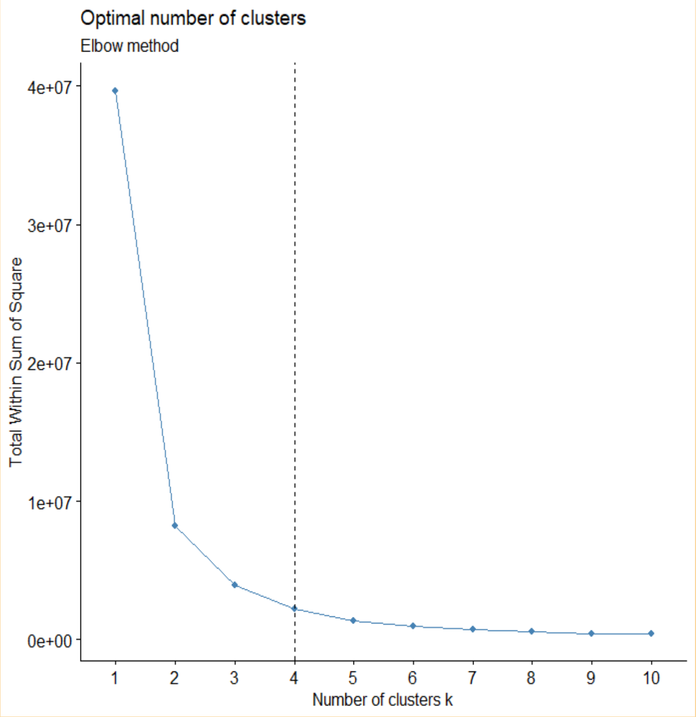


Inference: The table shows there are many customers never visited the store for very long time.

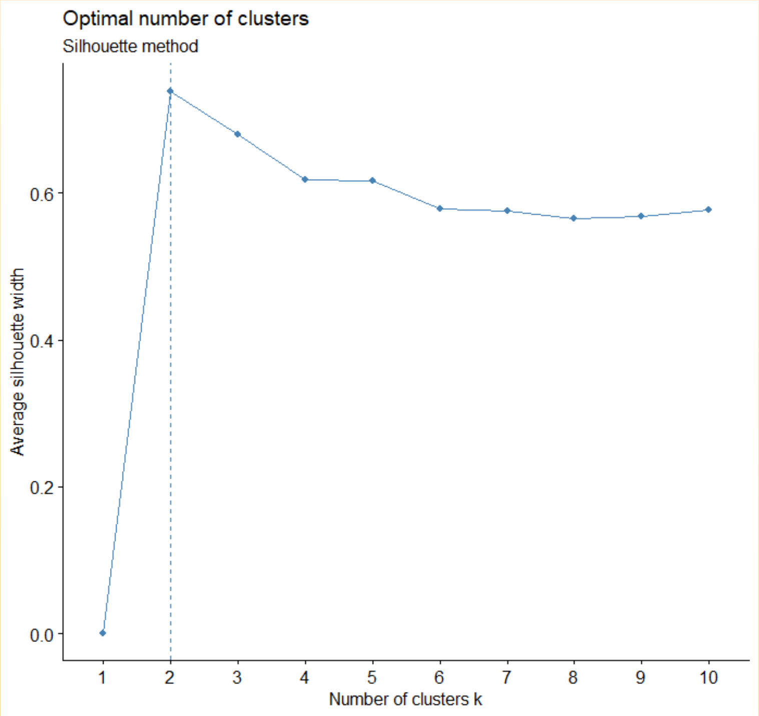
III - CUSTOMER SEGMENTATION

- Cluster all the segments using **k-means** technique. To identify optimum number of clusters to do our process, used the below methods.

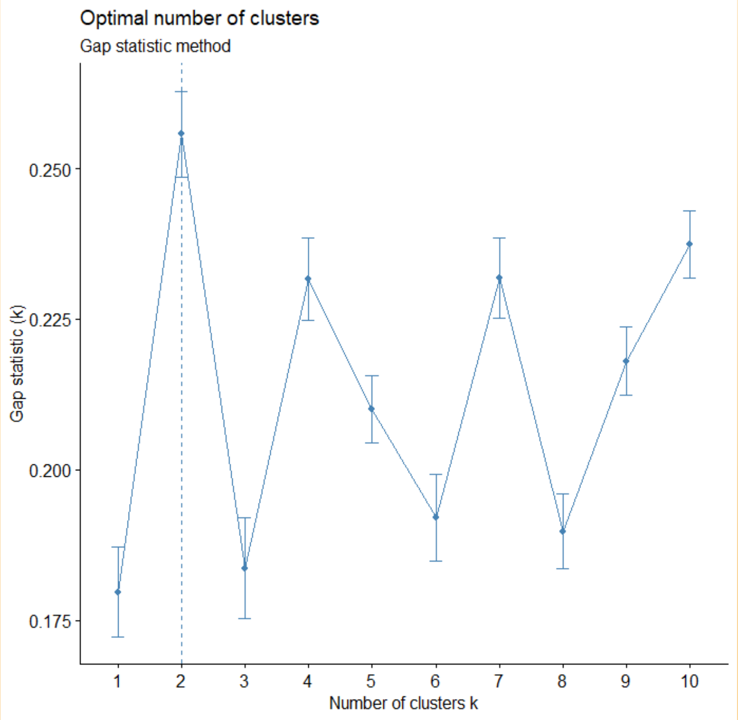
Elbow Method



Silhouette



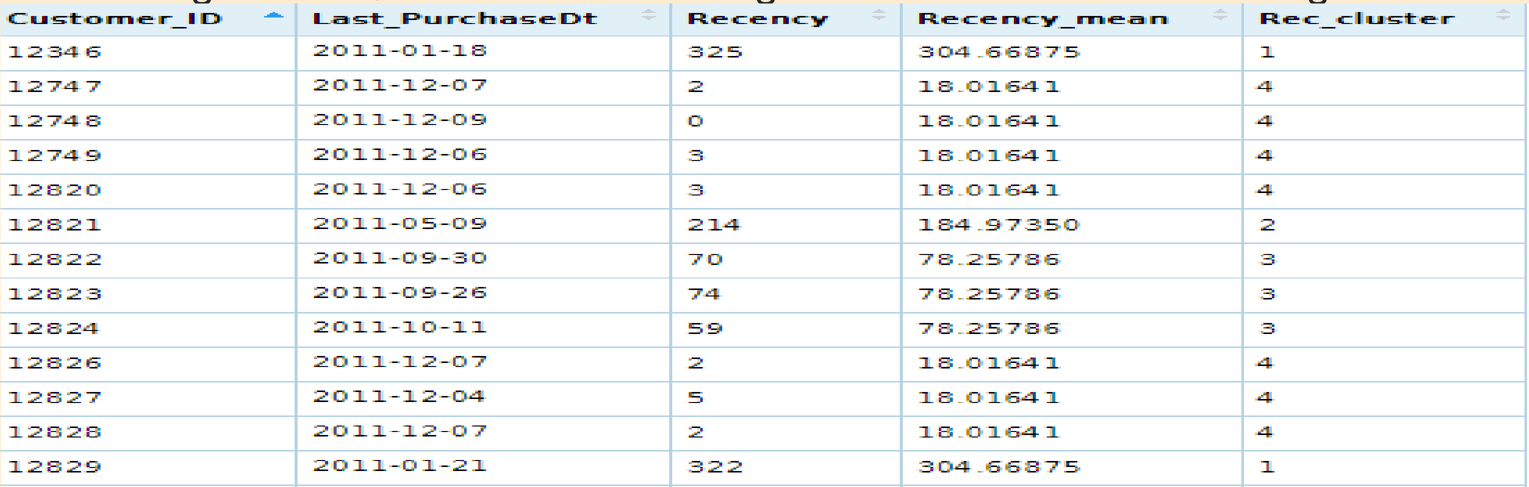
Gap statistic - nboot



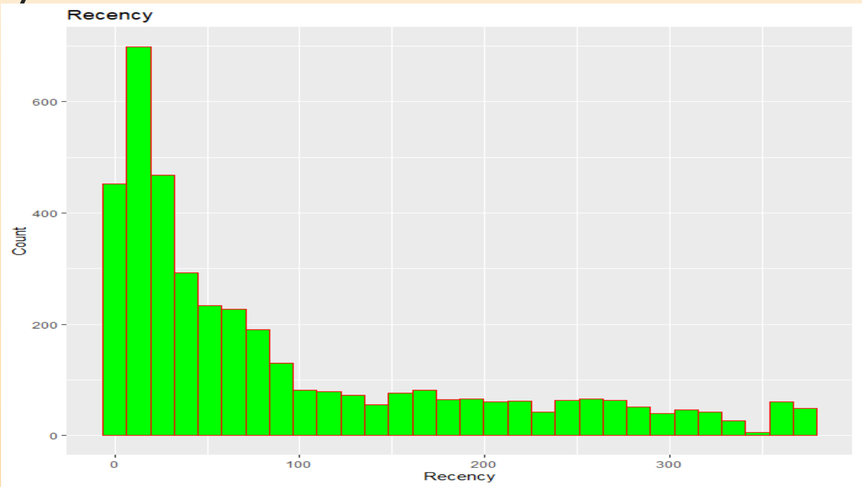
Inference: Even though Silhoutte and Gap statistic gives no. of clusters 2, it is better to go with 4 .

III - CUSTOMER SEGMENTATION

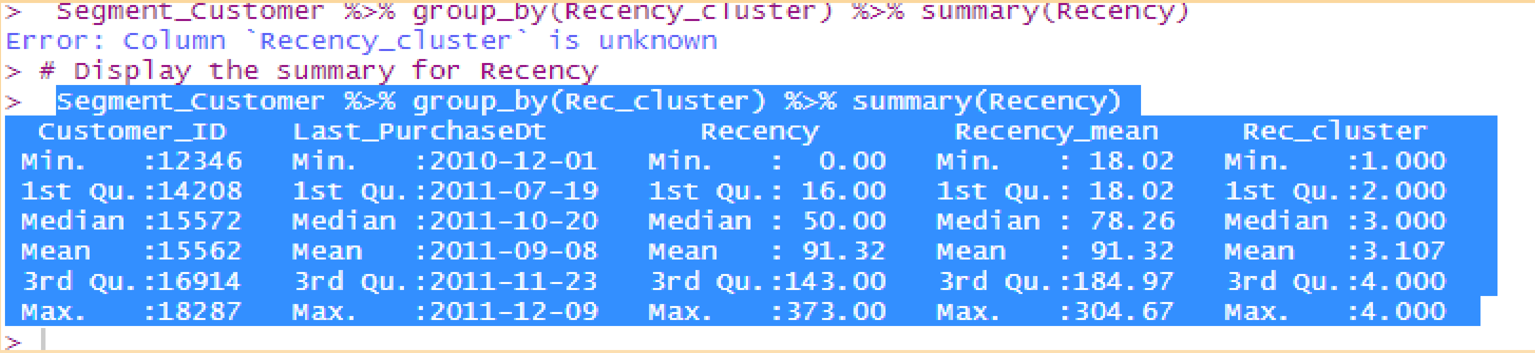
- **Recency**
  - Using K-means, each customer is assigned a cluster number and also got the recency mean.



Customer_ID	Last_PurchaseDt	Recency	Recency_mean	Rec_cluster
12346	2011-01-18	325	304.66875	1
12747	2011-12-07	2	18.01641	4
12748	2011-12-09	0	18.01641	4
12749	2011-12-06	3	18.01641	4
12820	2011-12-06	3	18.01641	4
12821	2011-05-09	214	184.97350	2
12822	2011-09-30	70	78.25786	3
12823	2011-09-26	74	78.25786	3
12824	2011-10-11	59	78.25786	3
12826	2011-12-07	2	18.01641	4
12827	2011-12-04	5	18.01641	4
12828	2011-12-07	2	18.01641	4
12829	2011-01-21	322	304.66875	1



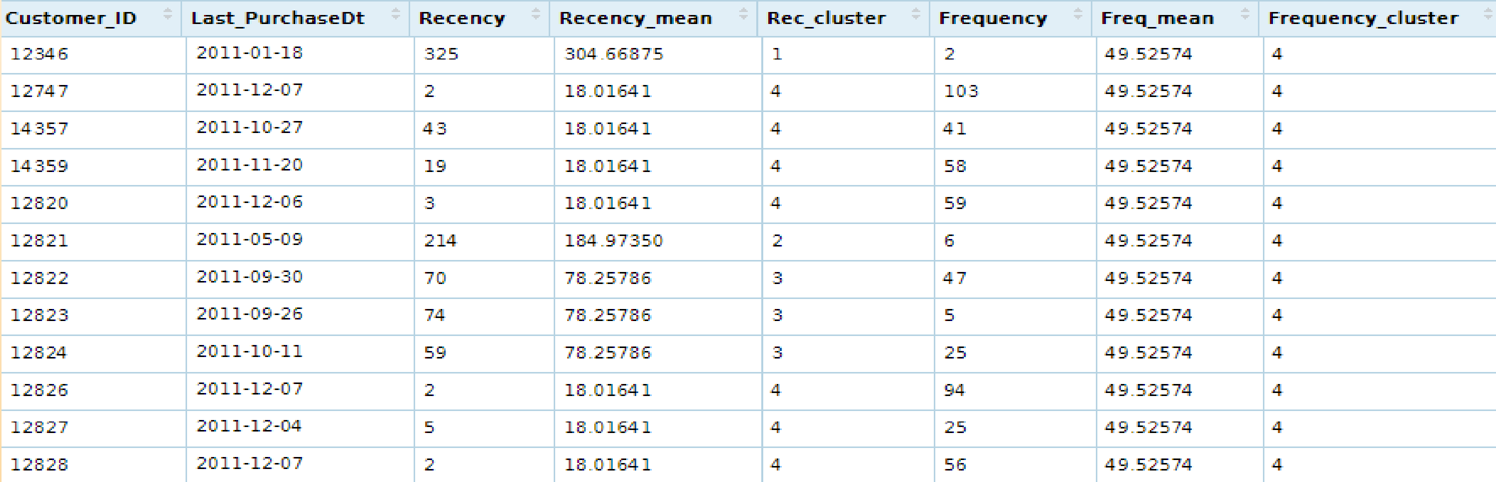
  - Get the range of different columns using summary command



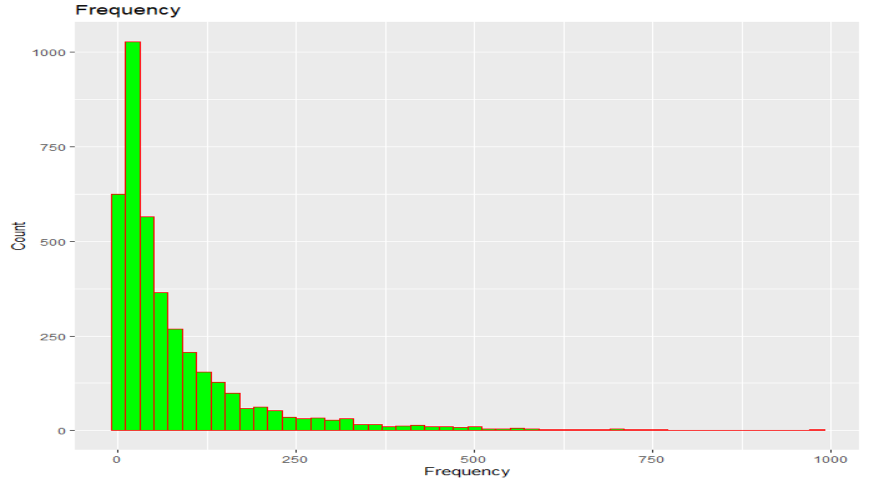
```
>  Segment_Customer %>% group_by(Recency_cluster) %>% summary(Recency)
Error: Column `Recency_cluster` is unknown
> # Display the summary for Recency
>  Segment_Customer %>% group_by(Rec_cluster) %>% summary(Recency)
  Customer_ID     Last_PurchaseDt        Recency         Recency_mean      Rec_cluster
 Min.   :12346   Min.   :2010-12-01   Min.   :  0.00   Min.   : 18.02   Min.   :1.000
 1st Qu.:14208   1st Qu.:2011-07-19   1st Qu.: 16.00   1st Qu.: 18.02   1st Qu.:2.000
 Median :15572   Median :2011-10-20   Median : 50.00   Median : 78.26   Median :3.000
 Mean   :15562   Mean   :2011-09-08   Mean   : 91.32   Mean   : 91.32   Mean   :3.107
 3rd Qu.:16914   3rd Qu.:2011-11-23   3rd Qu.:143.00   3rd Qu.:184.97   3rd Qu.:4.000
 Max.   :18287   Max.   :2011-12-09   Max.   :373.00   Max.   :304.67   Max.   :4.000
> 
```

III - CUSTOMER SEGMENTATION

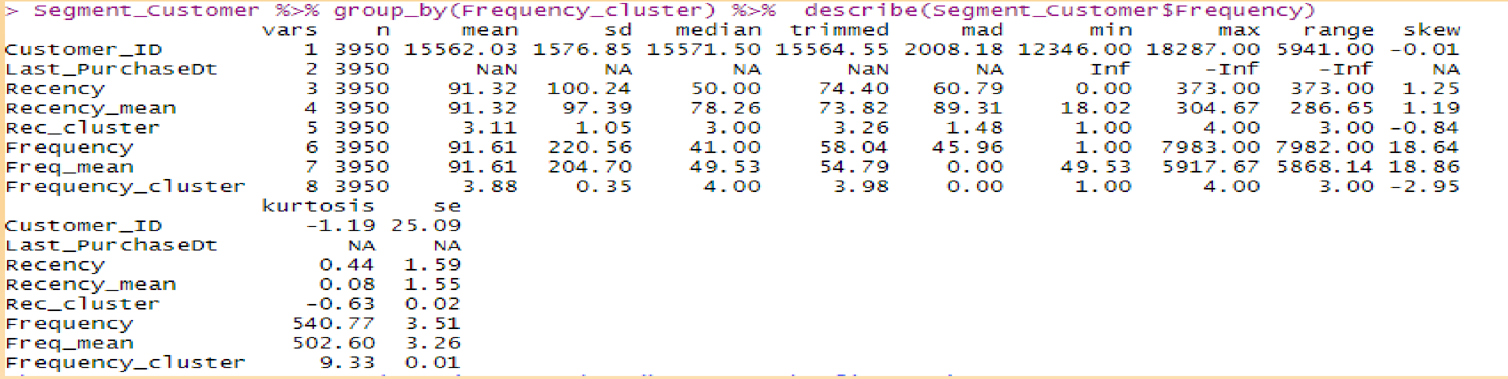
- **Frequency**
  - Using K-means, each customer is assigned a cluster number and also got the frequency and frequency mean.



Customer_ID	Last_PurchaseDt	Recency	Recency_mean	Rec_cluster	Frequency	Freq_mean	Frequency_cluster
12346	2011-01-18	325	304.66875	1	2	49.52574	4
12747	2011-12-07	2	18.01641	4	103	49.52574	4
14357	2011-10-27	43	18.01641	4	41	49.52574	4
14359	2011-11-20	19	18.01641	4	58	49.52574	4
12820	2011-12-06	3	18.01641	4	59	49.52574	4
12821	2011-05-09	214	184.97350	2	6	49.52574	4
12822	2011-09-30	70	78.25786	3	47	49.52574	4
12823	2011-09-26	74	78.25786	3	5	49.52574	4
12824	2011-10-11	59	78.25786	3	25	49.52574	4
12826	2011-12-07	2	18.01641	4	94	49.52574	4
12827	2011-12-04	5	18.01641	4	25	49.52574	4
12828	2011-12-07	2	18.01641	4	56	49.52574	4



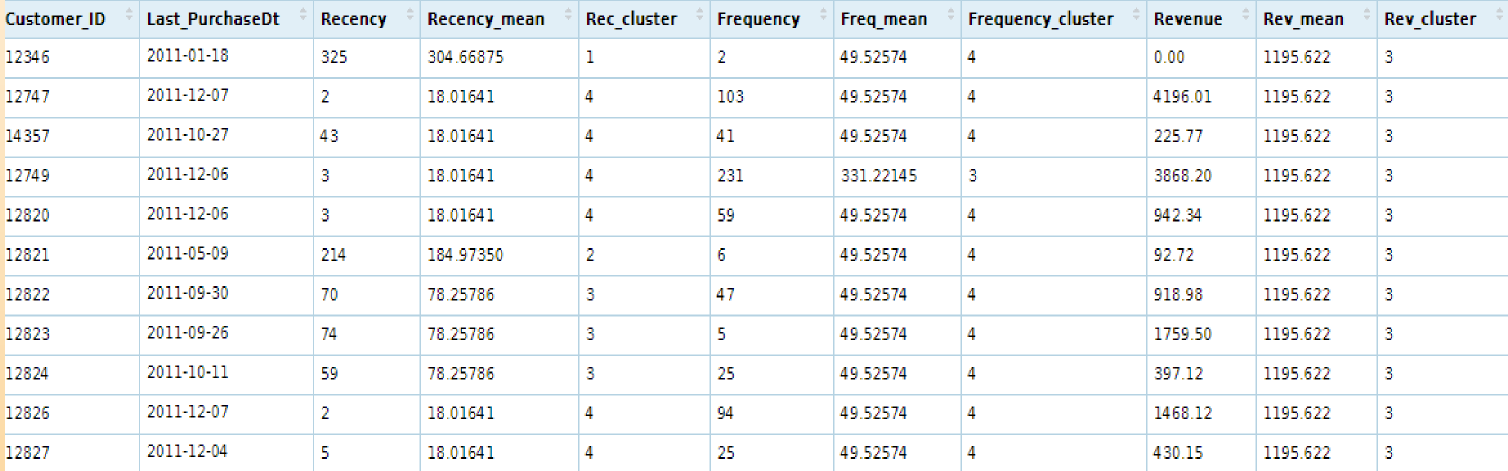
  - Get the range of different columns using summary command



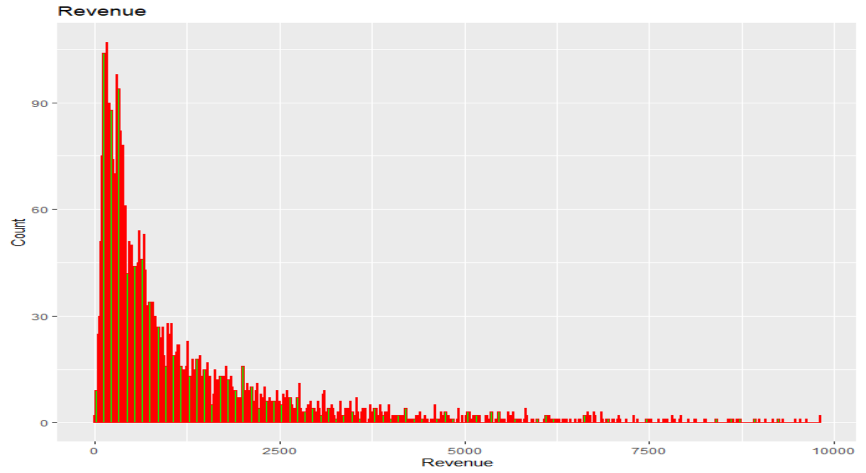
```
> Segment_Customer %>% group_by(Frequency_cluster) %>%  describe(Segment_Customer$Frequency)
                  vars    n     mean      sd   median  trimmed     mad      min      max   range  skew
Customer_ID          1 3950 15562.03 1576.85 15571.50 15564.55 2008.18 12346.00 18287.00 5941.00 -0.01
Last_PurchaseDt      2 3950      NaN      NA       NA      NaN      NA      Inf     -Inf    -Inf    NA
Recency              3 3950    91.32  100.24    50.00    74.40   60.79     0.00   373.00  373.00  1.25
Recency_mean         4 3950    91.32   97.39    78.26    73.82   89.31    18.02   304.67  286.65  1.19
Rec_cluster          5 3950     3.11    1.05     3.00     3.26    1.48     1.00     4.00    3.00 -0.84
Frequency            6 3950    91.61  220.56    41.00    58.04   45.96     1.00  7983.00 7982.00 18.64
Freq_mean            7 3950    91.61  204.70    49.53    54.79    0.00    49.53  5917.67 5868.14 18.86
Frequency_cluster    8 3950     3.88    0.35     4.00     3.98    0.00     1.00     4.00    3.00 -2.95
                  kurtosis    se
Customer_ID          -1.19 25.09
Last_PurchaseDt         NA    NA
Recency               0.44  1.59
Recency_mean          0.08  1.55
Rec_cluster          -0.63  0.02
Frequency           540.77  3.51
Freq_mean           502.60  3.26
Frequency_cluster     9.33  0.01
```

III - CUSTOMER SEGMENTATION

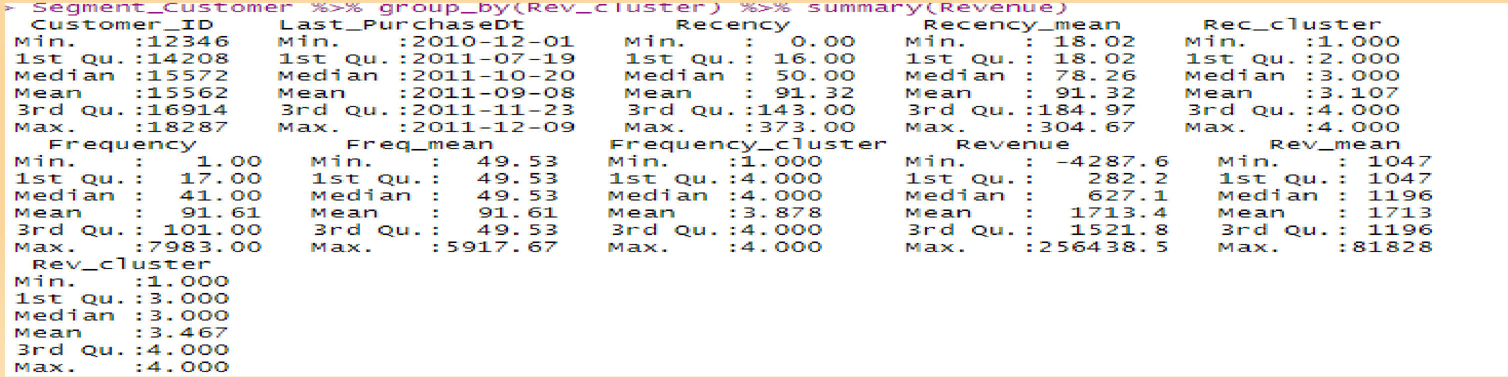
- **Monetary (Revenue)**
  - Using K-means, each customer is assigned a cluster number and also got the revenue and revenue mean



Customer_ID	Last_PurchaseDt	Recency	Recency_mean	Rec_cluster	Frequency	Freq_mean	Frequency_cluster	Revenue	Rev_mean	Rev_cluster
12346	2011-01-18	325	304.66875	1	2	49.52574	4	0.00	1195.622	3
12747	2011-12-07	2	18.01641	4	103	49.52574	4	4196.01	1195.622	3
14357	2011-10-27	43	18.01641	4	41	49.52574	4	225.77	1195.622	3
12749	2011-12-06	3	18.01641	4	231	331.22145	3	3868.20	1195.622	3
12820	2011-12-06	3	18.01641	4	59	49.52574	4	942.34	1195.622	3
12821	2011-05-09	214	184.97350	2	6	49.52574	4	92.72	1195.622	3
12822	2011-09-30	70	78.25786	3	47	49.52574	4	918.98	1195.622	3
12823	2011-09-26	74	78.25786	3	5	49.52574	4	1759.50	1195.622	3
12824	2011-10-11	59	78.25786	3	25	49.52574	4	397.12	1195.622	3
12826	2011-12-07	2	18.01641	4	94	49.52574	4	1468.12	1195.622	3
12827	2011-12-04	5	18.01641	4	25	49.52574	4	430.15	1195.622	3



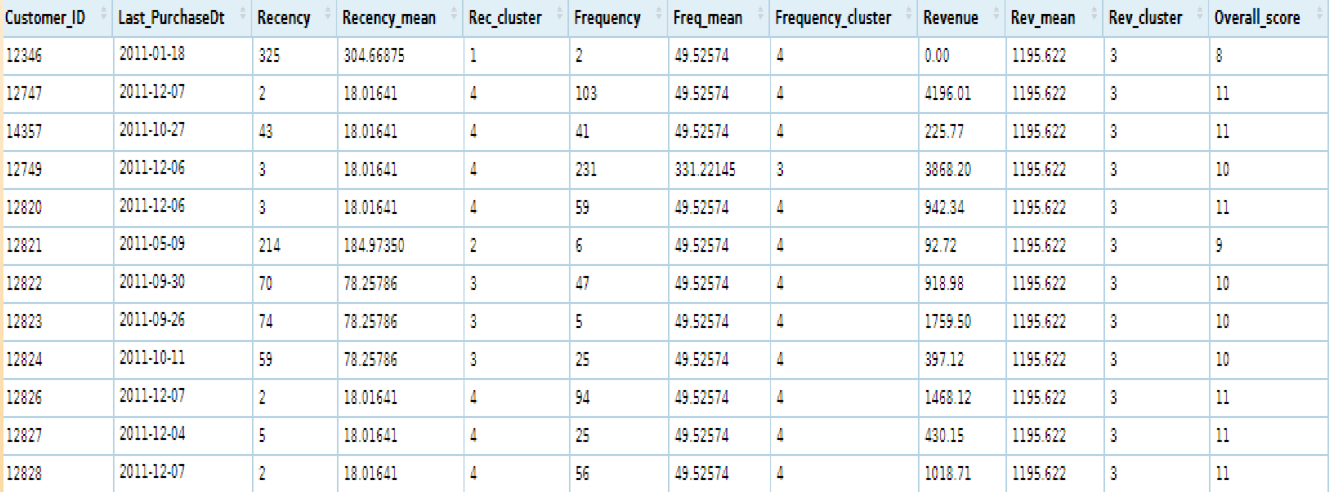
  - Get the range of different columns using summary command



```
> Segment_Customer %>% group_by(Rev_cluster) %>% summary(Revenue)
  Customer_ID     Last_PurchaseDt            Recency          Recency_mean       Rec_cluster
 Min.   :12346   Min.   :2010-12-01   Min.   :  0.00   Min.   : 18.02   Min.   :1.000
 1st Qu.:14208   1st Qu.:2011-07-19   1st Qu.: 16.00   1st Qu.: 18.02   1st Qu.:2.000
 Median :15572   Median :2011-10-20   Median : 50.00   Median : 78.26   Median :3.000
 Mean   :15562   Mean   :2011-09-08   Mean   : 91.32   Mean   : 91.32   Mean   :3.107
 3rd Qu.:16914   3rd Qu.:2011-11-23   3rd Qu.:143.00   3rd Qu.:184.97   3rd Qu.:4.000
 Max.   :18287   Max.   :2011-12-09   Max.   :373.00   Max.   :304.67   Max.   :4.000
   Frequency           Freq_mean        Frequency_cluster    Revenue              Rev_mean
 Min.   :   1.00   Min.   :  49.53   Min.   :1.000     Min.   : -4287.6   Min.   : 1047
 1st Qu.:  17.00   1st Qu.:  49.53   1st Qu.:4.000     1st Qu.:   282.2   1st Qu.: 1047
 Median :  41.00   Median :  49.53   Median :4.000     Median :   627.1   Median : 1196
 Mean   :  91.61   Mean   :  91.61   Mean   :3.878     Mean   :  1713.4   Mean   : 1713
 3rd Qu.: 101.00   3rd Qu.:  49.53   3rd Qu.:4.000     3rd Qu.:  1521.8   3rd Qu.: 1196
 Max.   :7983.00   Max.   :5917.67   Max.   :4.000     Max.   :256438.5   Max.   :81828
  Rev_cluster
 Min.   :1.000
 1st Qu.:3.000
 Median :3.000
 Mean   :3.467
 3rd Qu.:4.000
 Max.   :4.000
```

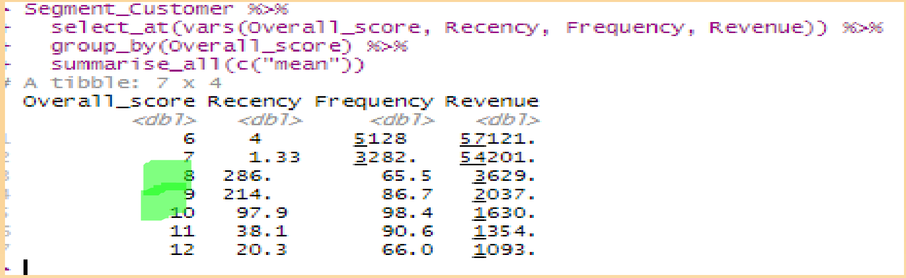
III - CUSTOMER SEGMENTATION

- **Overall score**
  - Using the cluster number of Recency, Monetary, and Frequency



Customer_ID	Last_PurchaseDt	Recency	Recency_mean	Rec_cluster	Frequency	Freq_mean	Frequency_cluster	Revenue	Rev_mean	Rev_cluster	Overall_score
12346	2011-01-18	325	304.66875	1	2	49.52574	4	0.00	1195.622	3	8
12747	2011-12-07	2	18.01641	4	103	49.52574	4	4196.01	1195.622	3	11
14357	2011-10-27	43	18.01641	4	41	49.52574	4	225.77	1195.622	3	11
12749	2011-12-06	3	18.01641	4	231	331.22145	3	3868.20	1195.622	3	10
12820	2011-12-06	3	18.01641	4	59	49.52574	4	942.34	1195.622	3	11
12821	2011-05-09	214	184.97350	2	6	49.52574	4	92.72	1195.622	3	9
12822	2011-09-30	70	78.25786	3	47	49.52574	4	918.98	1195.622	3	10
12823	2011-09-26	74	78.25786	3	5	49.52574	4	1759.50	1195.622	3	10
12824	2011-10-11	59	78.25786	3	25	49.52574	4	397.12	1195.622	3	10
12826	2011-12-07	2	18.01641	4	94	49.52574	4	1468.12	1195.622	3	11
12827	2011-12-04	5	18.01641	4	25	49.52574	4	430.15	1195.622	3	11
12828	2011-12-07	2	18.01641	4	56	49.52574	4	1018.71	1195.622	3	11

  - The below shows how the segment can be arrived using the overall score.

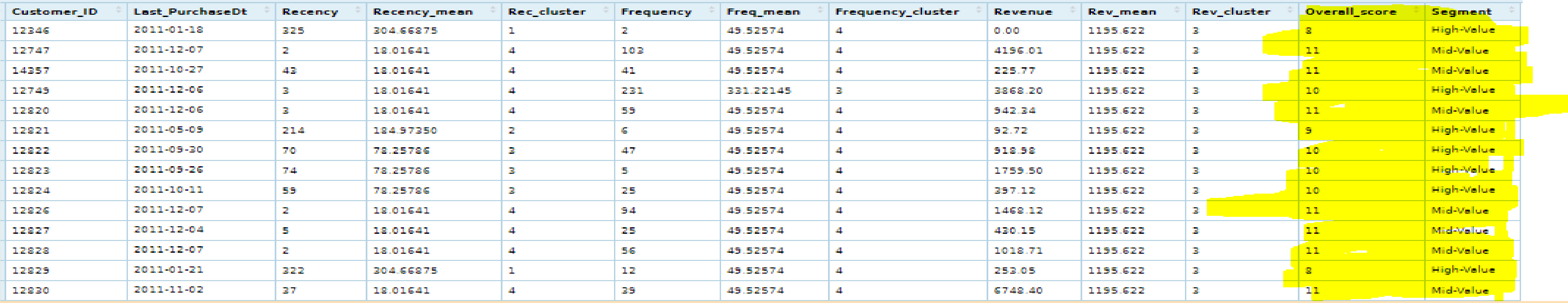


```
 Segment_Customer %>%
   select_at(vars(Overall_score, Recency, Frequency, Revenue)) %>%
   group_by(Overall_score) %>%
   summarise_all(c("mean"))
 A tibble: 7 x 4
 Overall_score Recency Frequency Revenue
         <dbl>   <dbl>     <dbl>   <dbl>
             6    4         5128   57121.
             7    1.33      3282.  54201.
             8  286.          65.5  3629.
             9  214.          86.7  2037.
            10   97.9         98.4  1630.
            11   38.1         90.6  1354.
            12   20.3         66.0  1093.
```

Couldn't find a better R statement to display the below in Overall score and Recency order.
EVery time it is giving different results

The scoring above clearly shows us that customers with score 8, 9&10 are our best customers whereas 6, 7 and 12 are the worst.
#To keep things simple, better we name these scores:
# 6, 7, and 12: Low Value
# 8, 9 and 10 : High Value
# 11 & 12 : Mid Value

III - CUSTOMER SEGMENTATION



Customer_ID	Last_PurchaseDt	Recency	Recency_mean	Rec_cluster	Frequency	Freq_mean	Frequency_cluster	Revenue	Rev_mean	Rev_cluster	Overall_score	Segment
12346	2011-01-18	325	304.66875	1	2	49.52574	4	0.00	1195.622	3	8	High-Value
12747	2011-12-07	2	18.01641	4	103	49.52574	4	4196.01	1195.622	3	11	Mid-Value
14357	2011-10-27	43	18.01641	4	41	49.52574	4	225.77	1195.622	3	11	Mid-Value
12749	2011-12-06	3	18.01641	4	231	331.22145	3	3868.20	1195.622	3	10	High-Value
12820	2011-12-06	3	18.01641	4	59	49.52574	4	942.34	1195.622	3	11	Mid-Value
12821	2011-05-09	214	184.97350	2	6	49.52574	4	92.72	1195.622	3	9	High-Value
12822	2011-09-30	70	78.25786	3	47	49.52574	4	918.98	1195.622	3	10	High-Value
12823	2011-09-26	74	78.25786	3	5	49.52574	4	1759.50	1195.622	3	10	High-Value
12824	2011-10-11	59	78.25786	3	25	49.52574	4	397.12	1195.622	3	10	High-Value
12826	2011-12-07	2	18.01641	4	94	49.52574	4	1468.12	1195.622	3	11	Mid-Value
12827	2011-12-04	5	18.01641	4	25	49.52574	4	430.15	1195.622	3	11	Mid-Value
12828	2011-12-07	2	18.01641	4	56	49.52574	4	1018.71	1195.622	3	11	Mid-Value
12829	2011-01-21	322	304.66875	1	12	49.52574	4	253.05	1195.622	3	8	High-Value
12830	2011-11-02	37	18.01641	4	39	49.52574	4	6748.40	1195.622	3	11	Mid-Value

- This segmentation helps us to define action plan for each customer based on his/her segment group. Re-iterating the below.

1. **<u>Low Value:</u>**
   - Customers who are less active than others, not very frequent buyer/visitor and generates very low – zero.
   - May be negative revenue.
2. **<u>Mid Value:</u>**
   - In the middle of everything. Often using our platform (but not as much as our High Values), fairly frequent and generates moderate revenue.
3. **<u>High Value:</u>**
   - The group we don't want to lose. High Revenue, Frequency and low Inactivity.

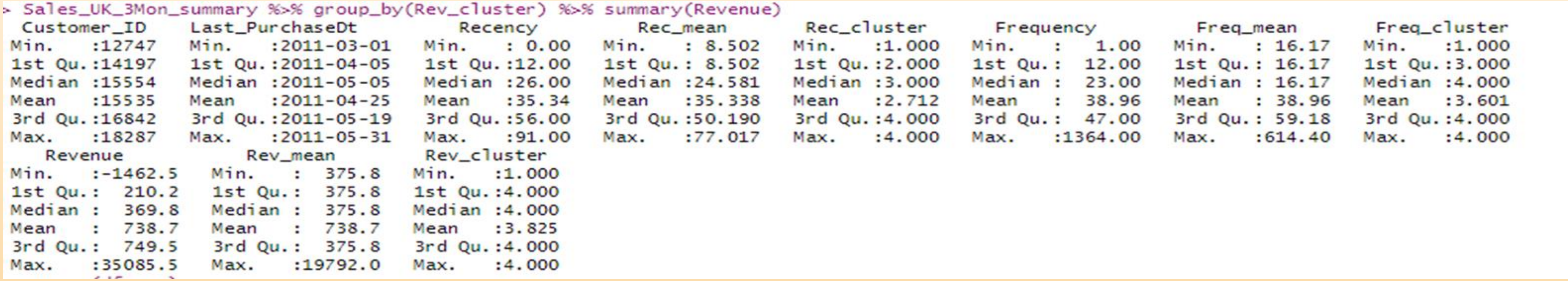
IV - PREDICTION OF CUSTOMER'S LIFE TIME VALUE (LTV)

IV - LIFE TIME VALUE (LTV) PREDICTION

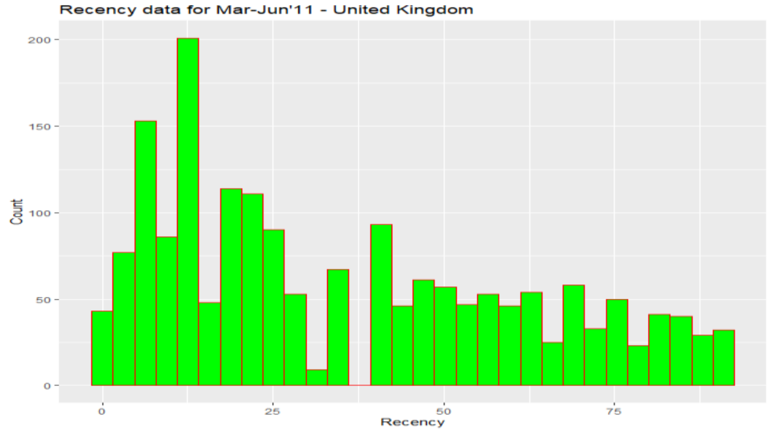
- **Data Preparation**
  - **To implement it correctly, we need to split our dataset.**
    - We will take 3 months (Mar-Apr-May) of data, calculate RFM and use it for predicting next 6 (Jun-Nov) months. So we need to create two data frames first and append RFM scores to them.
      - Sales_UK_3Mon <- Sales_UK_Data %>% subset(InvoiceDate1 >= "2011-03-01" & InvoiceDate1 < "2011-06-01")
      - Sales_UK_6Mon <- Sales_UK_Data %>% subset(InvoiceDate1 >= "2011-06-01" & InvoiceDate1 < "2011-12-01")
    - Using 3 months data, calculate Recency, Frequency, and Monetary like before. Apply K-means on each to respective mean, and cluster number. Finally calculate the overall score to segment the customer
    - Using 6 months data, calculate the Revenue.
    - Merge the RFM dataset created using 3 months data and the Revenue dataset created using 6 months data
    - Apply K-means on the 6 months data Revenue to identify the LTV cluster
- **Correlation**
- **Machine Learning Techniques – Using LTV cluster apply the following:**
  - Gradient Boosting
  - Linear Regression an Polynomial (up to degree 3)
  - Naïve Bayes
  - LOOCV – Cross Validation
  - K Fold Model and Bootstrap

IV - LIFE TIME VALUE (LTV) PREDICTION

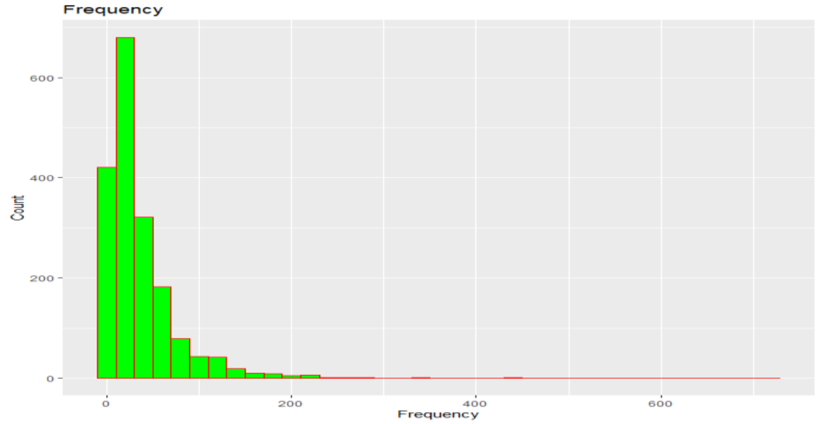
- Using 3 months data, calculate Recency, Frequency, and Monetary like before. Apply K-means on each to respective mean, and cluster number. Finally calculate the overall score to segment the customer



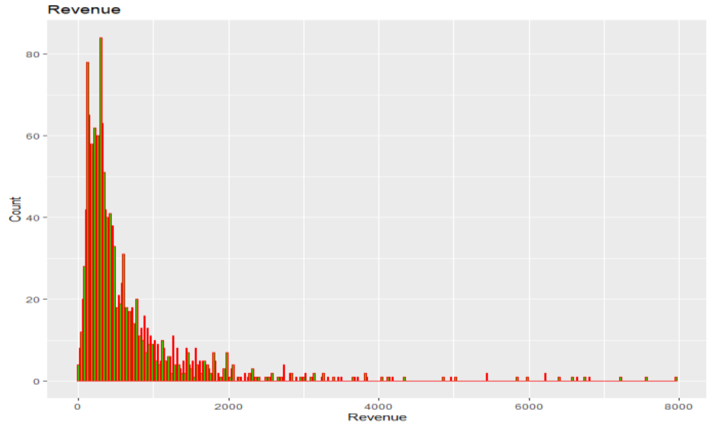
```
> Sales_UK_3Mon_summary %>% group_by(Rev_cluster) %>% summary(Revenue)
  Customer_ID     Last_PurchaseDt          Recency          Rec_mean        Rec_cluster       Frequency         Freq_mean        Freq_cluster
 Min.   :12747   Min.   :2011-03-01   Min.   : 0.00   Min.   : 8.502   Min.   :1.000   Min.   :   1.00   Min.   : 16.17   Min.   :1.000
 1st Qu.:14197   1st Qu.:2011-04-05   1st Qu.:12.00   1st Qu.: 8.502   1st Qu.:2.000   1st Qu.:  12.00   1st Qu.: 16.17   1st Qu.:3.000
 Median :15554   Median :2011-05-05   Median :26.00   Median :24.581   Median :3.000   Median :  23.00   Median : 16.17   Median :4.000
 Mean   :15535   Mean   :2011-04-25   Mean   :35.34   Mean   :35.338   Mean   :2.712   Mean   :  38.96   Mean   : 38.96   Mean   :3.601
 3rd Qu.:16842   3rd Qu.:2011-05-19   3rd Qu.:56.00   3rd Qu.:50.190   3rd Qu.:4.000   3rd Qu.:  47.00   3rd Qu.: 59.18   3rd Qu.:4.000
 Max.   :18287   Max.   :2011-05-31   Max.   :91.00   Max.   :77.017   Max.   :4.000   Max.   :1364.00   Max.   :614.40   Max.   :4.000
    Revenue            Rev_mean         Rev_cluster
 Min.   :-1462.5   Min.   :  375.8   Min.   :1.000
 1st Qu.:  210.2   1st Qu.:  375.8   1st Qu.:4.000
 Median :  369.8   Median :  375.8   Median :4.000
 Mean   :  738.7   Mean   :  738.7   Mean   :3.825
 3rd Qu.:  749.5   3rd Qu.:  375.8   3rd Qu.:4.000
 Max.   :35085.5   Max.   :19792.0   Max.   :4.000
```



Recency data for Mar-Jun'11 - United Kingdom



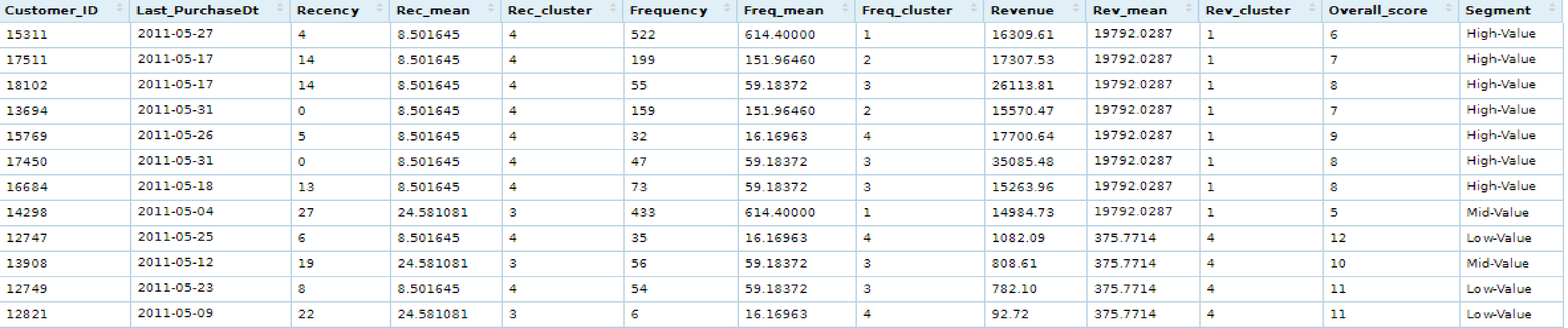
Frequency



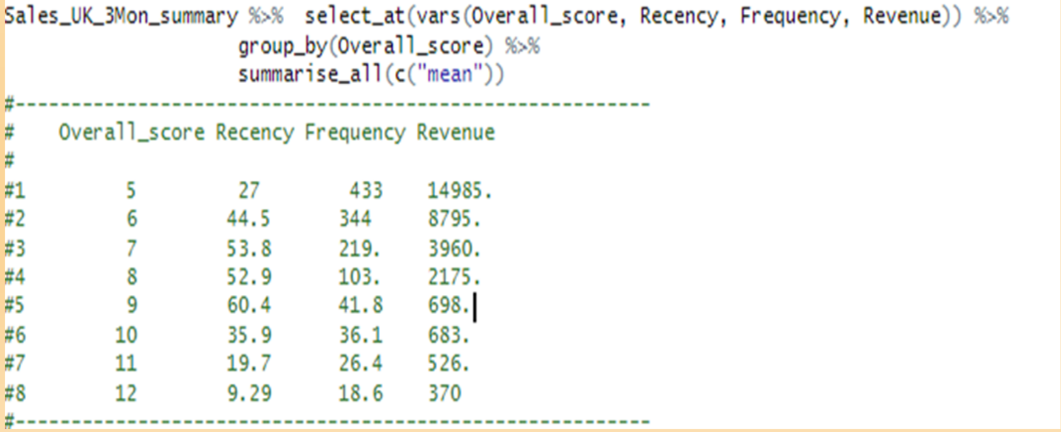
Revenue

IV - LIFE TIME VALUE (LTV) PREDICTION

- Using 3 months data, calculate Recency, Frequency, and Monetary like before. Apply K-means on each to respective mean, and cluster number. Finally calculate the overall score to segment the customer



Customer_ID	Last_PurchaseDt	Recency	Rec_mean	Rec_cluster	Frequency	Freq_mean	Freq_cluster	Revenue	Rev_mean	Rev_cluster	Overall_score	Segment
15311	2011-05-27	4	8.501645	4	522	614.40000	1	16309.61	19792.0287	1	6	High-Value
17511	2011-05-17	14	8.501645	4	199	151.96460	2	17307.53	19792.0287	1	7	High-Value
18102	2011-05-17	14	8.501645	4	55	59.18372	3	26113.81	19792.0287	1	8	High-Value
13694	2011-05-31	0	8.501645	4	159	151.96460	2	15570.47	19792.0287	1	7	High-Value
15769	2011-05-26	5	8.501645	4	32	16.16963	4	17700.64	19792.0287	1	9	High-Value
17450	2011-05-31	0	8.501645	4	47	59.18372	3	35085.48	19792.0287	1	8	High-Value
16684	2011-05-18	13	8.501645	4	73	59.18372	3	15263.96	19792.0287	1	8	High-Value
14298	2011-05-04	27	24.581081	3	433	614.40000	1	14984.73	19792.0287	1	5	Mid-Value
12747	2011-05-25	6	8.501645	4	35	16.16963	4	1082.09	375.7714	4	12	Low-Value
13908	2011-05-12	19	24.581081	3	56	59.18372	3	808.61	375.7714	4	10	Mid-Value
12749	2011-05-23	8	8.501645	4	54	59.18372	3	782.10	375.7714	4	11	Low-Value
12821	2011-05-09	22	24.581081	3	6	16.16963	4	92.72	375.7714	4	11	Low-Value

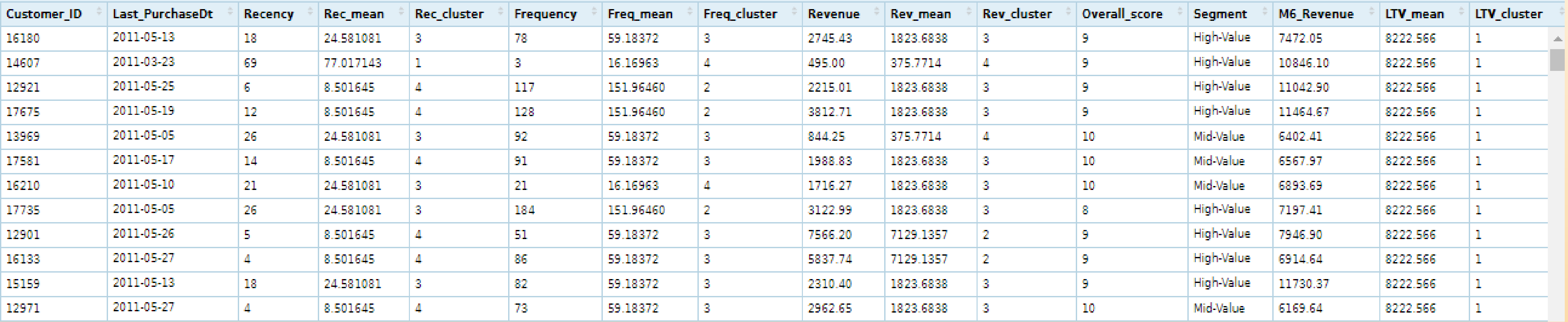


```
Sales_UK_3Mon_summary %>%  select_at(vars(Overall_score, Recency, Frequency, Revenue)) %>%
                           group_by(Overall_score) %>%
                           summarise_all(c("mean"))
#-------------------------------------------------------------
#    Overall_score Recency Frequency Revenue
#
#1         5         27        433    14985.
#2         6        44.5       344    8795.
#3         7        53.8       219.   3960.
#4         8        52.9       103.   2175.
#5         9        60.4       41.8   698.
#6        10        35.9       36.1   683.
#7        11        19.7       26.4   526.
#8        12        9.29       18.6   370
#-------------------------------------------------------------
```

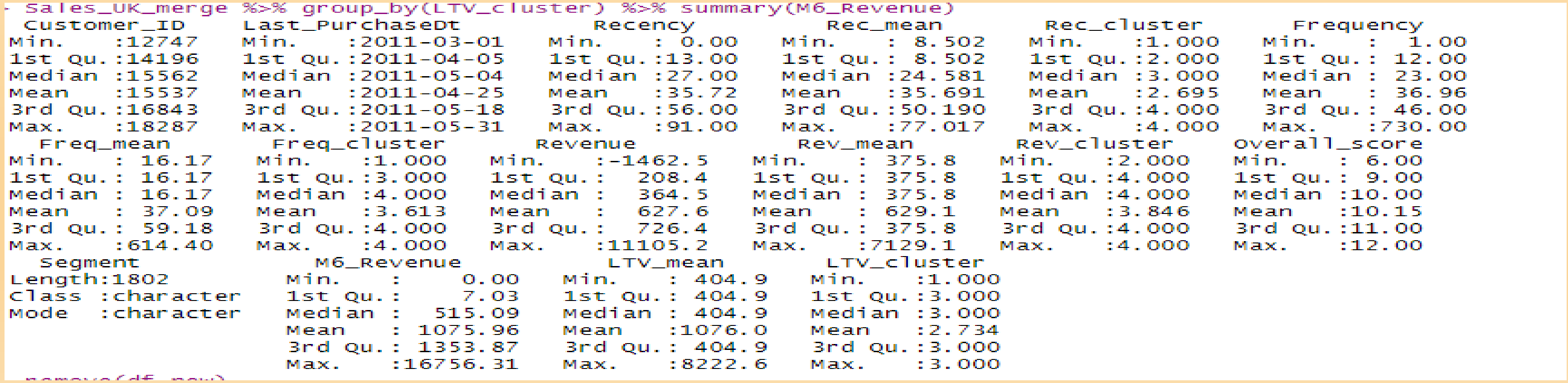
```
Sales_UK_3Mon_summary$Segment <- 'Low-Value'
Sales_UK_3Mon_summary$Segment[between(Sales_UK_3Mon_summary$Overall_score,6,9)] <- 'High-Value'
Sales_UK_3Mon_summary$Segment[Sales_UK_3Mon_summary$Overall_score == 10 | +
                             Sales_UK_3Mon_summary$Overall_score == 5] <- 'Mid-Value'
```

IV - LIFE TIME VALUE (LTV) PREDICTION

- Using 6 months data, calculate Recency, Frequency, and Monetary like before. Apply K-means on each to respective mean, and cluster number. Finally calculate the overall score to segment the customer



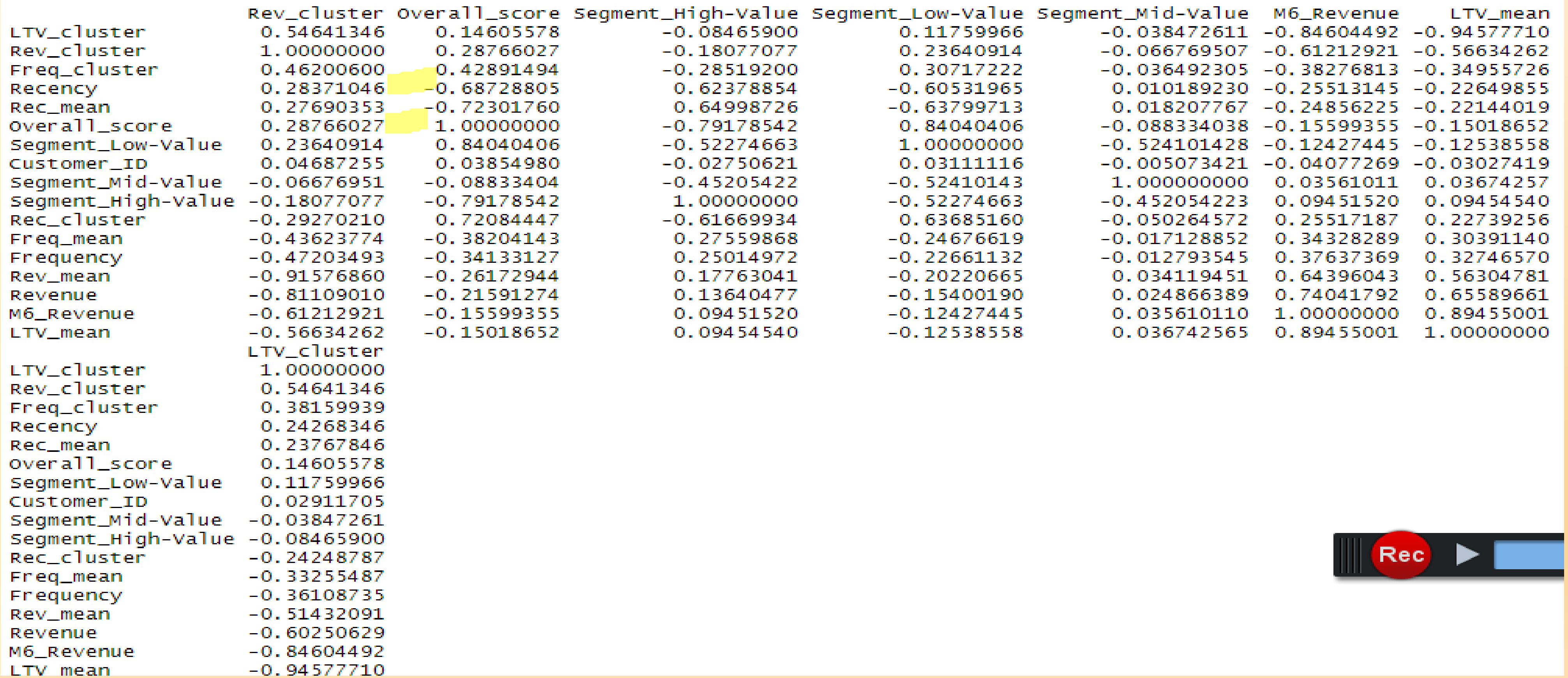
Customer_ID	Last_PurchaseDt	Recency	Rec_mean	Rec_cluster	Frequency	Freq_mean	Freq_cluster	Revenue	Rev_mean	Rev_cluster	Overall_score	Segment	M6_Revenue	LTV_mean	LTV_cluster
16180	2011-05-13	18	24.581081	3	78	59.18372	3	2745.43	1823.6838	3	9	High-Value	7472.05	8222.566	1
14607	2011-03-23	69	77.017143	1	3	16.16963	4	495.00	375.7714	4	9	High-Value	10846.10	8222.566	1
12921	2011-05-25	6	8.501645	4	117	151.96460	2	2215.01	1823.6838	3	9	High-Value	11042.90	8222.566	1
17675	2011-05-19	12	8.501645	4	128	151.96460	2	3812.71	1823.6838	3	9	High-Value	11464.67	8222.566	1
13969	2011-05-05	26	24.581081	3	92	59.18372	3	844.25	375.7714	4	10	Mid-Value	6402.41	8222.566	1
17581	2011-05-17	14	8.501645	4	91	59.18372	3	1988.83	1823.6838	3	10	Mid-Value	6567.97	8222.566	1
16210	2011-05-10	21	24.581081	3	21	16.16963	4	1716.27	1823.6838	3	10	Mid-Value	6893.69	8222.566	1
17735	2011-05-05	26	24.581081	3	184	151.96460	2	3122.99	1823.6838	3	8	High-Value	7197.41	8222.566	1
12901	2011-05-26	5	8.501645	4	51	59.18372	3	7566.20	7129.1357	2	9	High-Value	7946.90	8222.566	1
16133	2011-05-27	4	8.501645	4	86	59.18372	3	5837.74	7129.1357	2	9	High-Value	6914.64	8222.566	1
15159	2011-05-13	18	24.581081	3	82	59.18372	3	2310.40	1823.6838	3	9	High-Value	11730.37	8222.566	1
12971	2011-05-27	4	8.501645	4	73	59.18372	3	2962.65	1823.6838	3	10	Mid-Value	6169.64	8222.566	1



```
> Sales_UK_merge %>% group_by(LTV_cluster) %>% summary(M6_Revenue)
  Customer_ID    Last_PurchaseDt          Recency          Rec_mean         Rec_cluster      Frequency     
 Min.   :12747   Min.   :2011-03-01   Min.   : 0.00   Min.   : 8.502   Min.   :1.000   Min.   :  1.00  
 1st Qu.:14196   1st Qu.:2011-04-05   1st Qu.:13.00   1st Qu.: 8.502   1st Qu.:2.000   1st Qu.: 12.00  
 Median :15562   Median :2011-05-04   Median :27.00   Median :24.581   Median :3.000   Median : 23.00  
 Mean   :15537   Mean   :2011-04-25   Mean   :35.72   Mean   :35.691   Mean   :2.695   Mean   : 36.96  
 3rd Qu.:16843   3rd Qu.:2011-05-18   3rd Qu.:56.00   3rd Qu.:50.190   3rd Qu.:4.000   3rd Qu.: 46.00  
 Max.   :18287   Max.   :2011-05-31   Max.   :91.00   Max.   :77.017   Max.   :4.000   Max.   :730.00  
   Freq_mean       Freq_cluster      Revenue            Rev_mean        Rev_cluster    Overall_score  
 Min.   : 16.17   Min.   :1.000   Min.   :-1462.5   Min.   : 375.8   Min.   :2.000   Min.   : 6.00  
 1st Qu.: 16.17   1st Qu.:3.000   1st Qu.:  208.4   1st Qu.: 375.8   1st Qu.:4.000   1st Qu.: 9.00  
 Median : 16.17   Median :4.000   Median :  364.5   Median : 375.8   Median :4.000   Median :10.00  
 Mean   : 37.09   Mean   :3.613   Mean   :  627.6   Mean   : 629.1   Mean   :3.846   Mean   :10.15  
 3rd Qu.: 59.18   3rd Qu.:4.000   3rd Qu.:  726.4   3rd Qu.: 375.8   3rd Qu.:4.000   3rd Qu.:11.00  
 Max.   :614.40   Max.   :4.000   Max.   :11105.2   Max.   :7129.1   Max.   :4.000   Max.   :12.00  
   Segment            M6_Revenue         LTV_mean       LTV_cluster   
 Length:1802        Min.   :    0.00   Min.   : 404.9   Min.   :1.000  
 Class :character   1st Qu.:    7.03   1st Qu.: 404.9   1st Qu.:3.000  
 Mode  :character   Median :  515.09   Median : 404.9   Median :3.000  
                    Mean   : 1075.96   Mean   :1076.0   Mean   :2.734  
                    3rd Qu.: 1353.87   3rd Qu.: 404.9   3rd Qu.:3.000  
                    Max.   :16756.31   Max.   :8222.6   Max.   :3.000  
> remove(df_new)
```

IV - LIFE TIME VALUE (LTV) PREDICTION

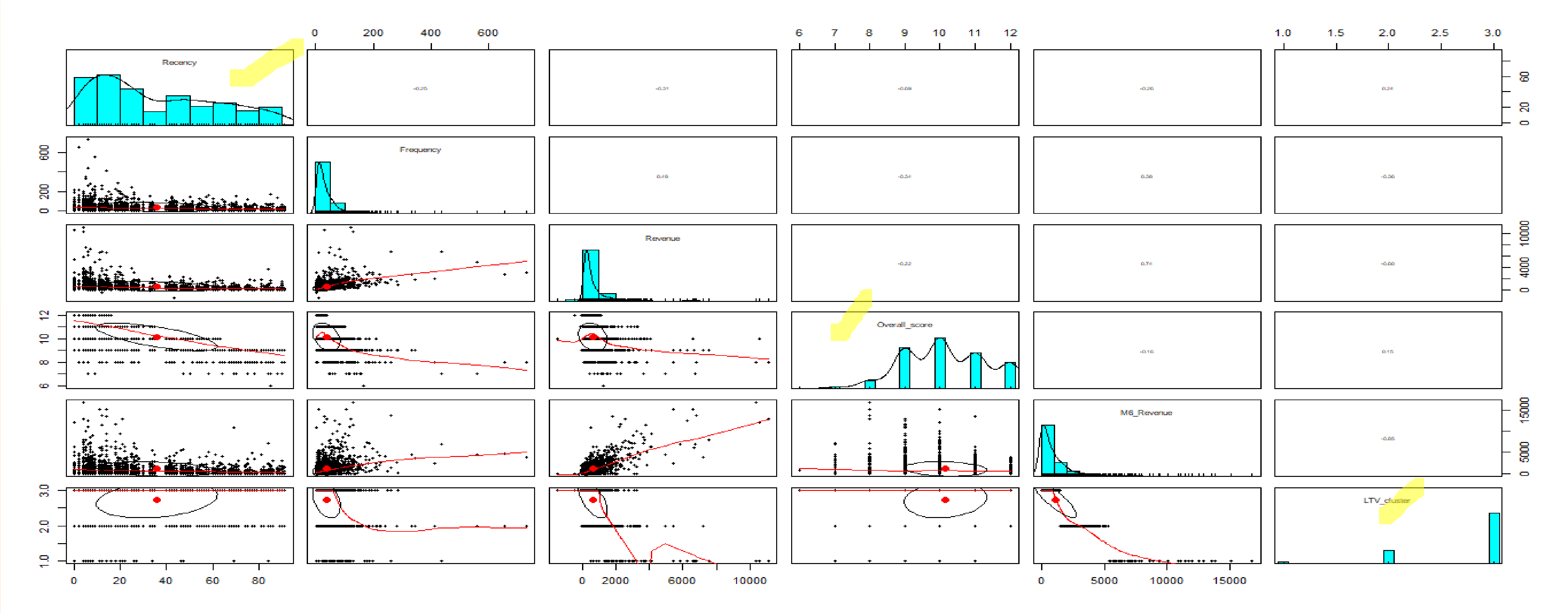
- **Correlation**
  - **Feature Engineering -** convert categorical columns to numerical columns using dummy.data.frame function
  - Apply Correlation to see the relative influence between LTV Cluster and other variables



```
                   Rev_cluster Overall_score Segment_High-Value Segment_Low-Value Segment_Mid-Value  M6_Revenue    LTV_mean
LTV_cluster         0.54641346    0.14605578        -0.08465900        0.11759966      -0.038472611 -0.84604492 -0.94577710
Rev_cluster         1.00000000    0.28766027        -0.18077077        0.23640914      -0.066769507 -0.61212921 -0.56634262
Freq_cluster        0.46200600    0.42891494        -0.28519200        0.30717222      -0.036492305 -0.38276813 -0.34955726
Recency             0.28371046   -0.68728805         0.62378854       -0.60531965       0.010189230 -0.25513145 -0.22649855
Rec_mean            0.27690353   -0.72301760         0.64998726       -0.63799713       0.018207767 -0.24856225 -0.22144019
Overall_score       0.28766027    1.00000000        -0.79178542        0.84040406      -0.088334038 -0.15599355 -0.15018652
Segment_Low-Value   0.23640914    0.84040406        -0.52274663        1.00000000      -0.524101428 -0.12427445 -0.12538558
Customer_ID         0.04687255    0.03854980        -0.02750621        0.03111116      -0.005073421 -0.04077269 -0.03027419
Segment_Mid-Value  -0.06676951   -0.08833404        -0.45205422       -0.52410143       1.000000000  0.03561011  0.03674257
Segment_High-Value -0.18077077   -0.79178542         1.00000000       -0.52274663      -0.452054223  0.09451520  0.09454540
Rec_cluster        -0.29270210    0.72084447        -0.61669934        0.63685160      -0.050264572  0.25517187  0.22739256
Freq_mean          -0.43623774   -0.38204143         0.27559868       -0.24676619      -0.017128852  0.34328289  0.30391140
Frequency          -0.47203493   -0.34133127         0.25014972       -0.22661132      -0.012793545  0.37637369  0.32746570
Rev_mean           -0.91576860   -0.26172944         0.17763041       -0.20220665       0.034119451  0.64396043  0.56304781
Revenue            -0.81109010   -0.21591274         0.13640477       -0.15400190       0.024866389  0.74041792  0.65589661
M6_Revenue         -0.61212921   -0.15599355         0.09451520       -0.12427445       0.035610110  1.00000000  0.89455001
LTV_mean           -0.56634262   -0.15018652         0.09454540       -0.12538558       0.036742565  0.89455001  1.00000000
                   LTV_cluster
LTV_cluster         1.00000000
Rev_cluster         0.54641346
Freq_cluster        0.38159939
Recency             0.24268346
Rec_mean            0.23767846
Overall_score       0.14605578
Segment_Low-Value   0.11759966
Customer_ID         0.02911705
Segment_Mid-Value  -0.03847261
Segment_High-Value -0.08465900
Rec_cluster        -0.24248787
Freq_mean          -0.33255487
Frequency          -0.36108735
Rev_mean           -0.51432091
Revenue            -0.60250629
M6_Revenue         -0.84604492
LTV_mean           -0.94577710
```

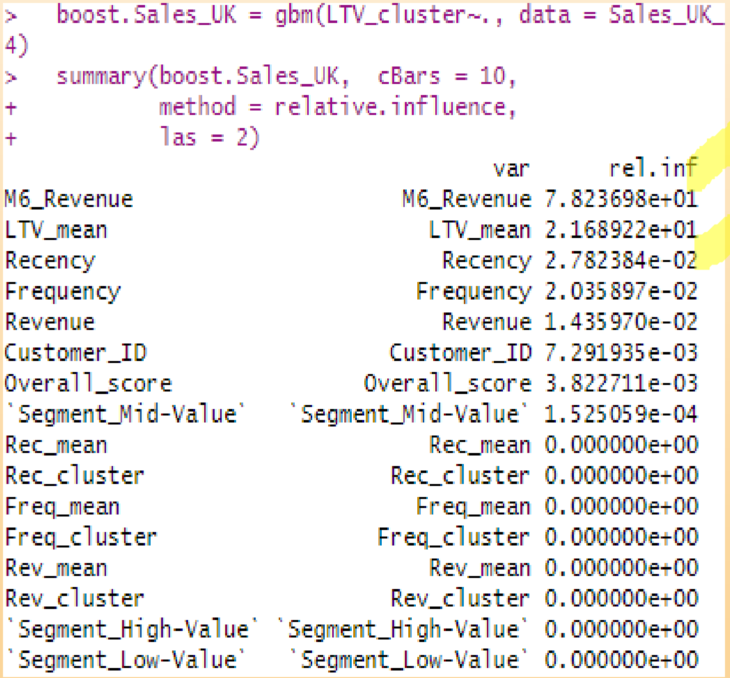
IV - LIFE TIME VALUE (LTV) PREDICTION

Correlation – using Pearson method. This confirms the correlation reported by the Correlation matrix.

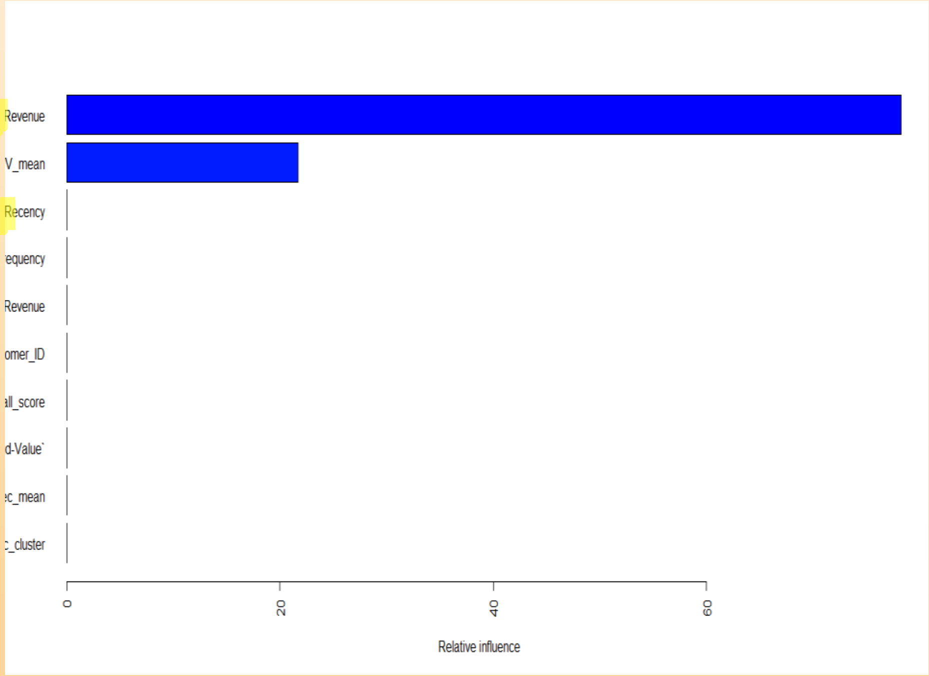


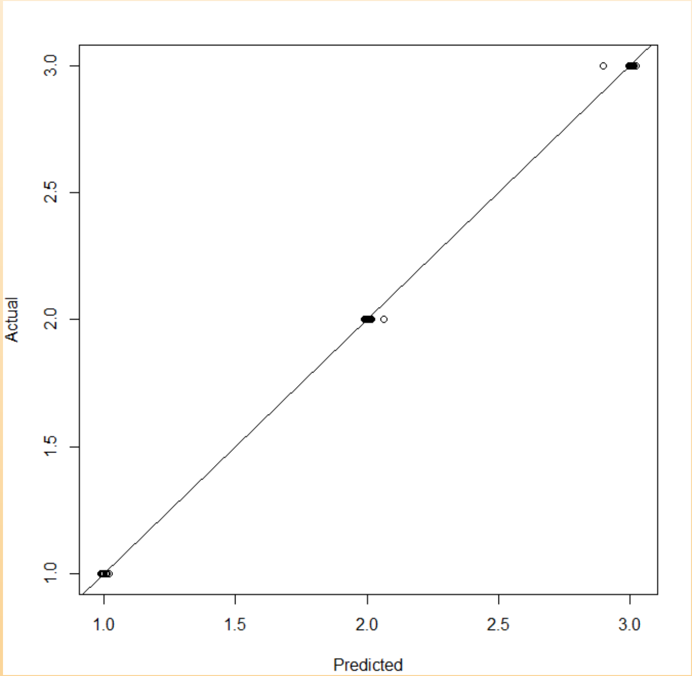
IV - LIFE TIME VALUE (LTV) PREDICTION

- The training and testing data set: 70:30 ratio
- **Gradient Boosting** - using n.trees = 5000, distribution="gaussian“, and interaction.depth=4



```
>   boost.Sales_UK = gbm(LTV_cluster~., data = Sales_UK_
4)
>   summary(boost.Sales_UK,  cBars = 10,
+           method = relative.influence,
+           las = 2)
                                        var      rel.inf
M6_Revenue                       M6_Revenue 7.823698e+01
LTV_mean                           LTV_mean 2.168922e+01
Recency                             Recency 2.782384e-02
Frequency                         Frequency 2.035897e-02
Revenue                             Revenue 1.435970e-02
Customer_ID                     Customer_ID 7.291935e-03
Overall_score                 Overall_score 3.822711e-03
`Segment_Mid-Value`     `Segment_Mid-Value` 1.525059e-04
Rec_mean                           Rec_mean 0.000000e+00
Rec_cluster                     Rec_cluster 0.000000e+00
Freq_mean                         Freq_mean 0.000000e+00
Freq_cluster                   Freq_cluster 0.000000e+00
Rev_mean                           Rev_mean 0.000000e+00
Rev_cluster                     Rev_cluster 0.000000e+00
`Segment_High-Value`   `Segment_High-Value` 0.000000e+00
`Segment_Low-Value`     `Segment_Low-Value` 0.000000e+00
```



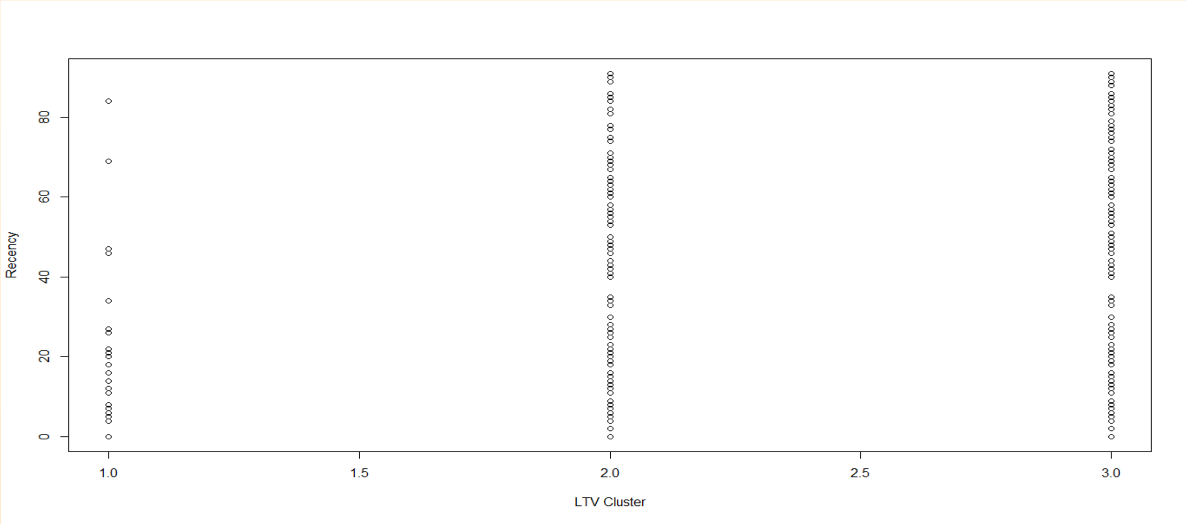


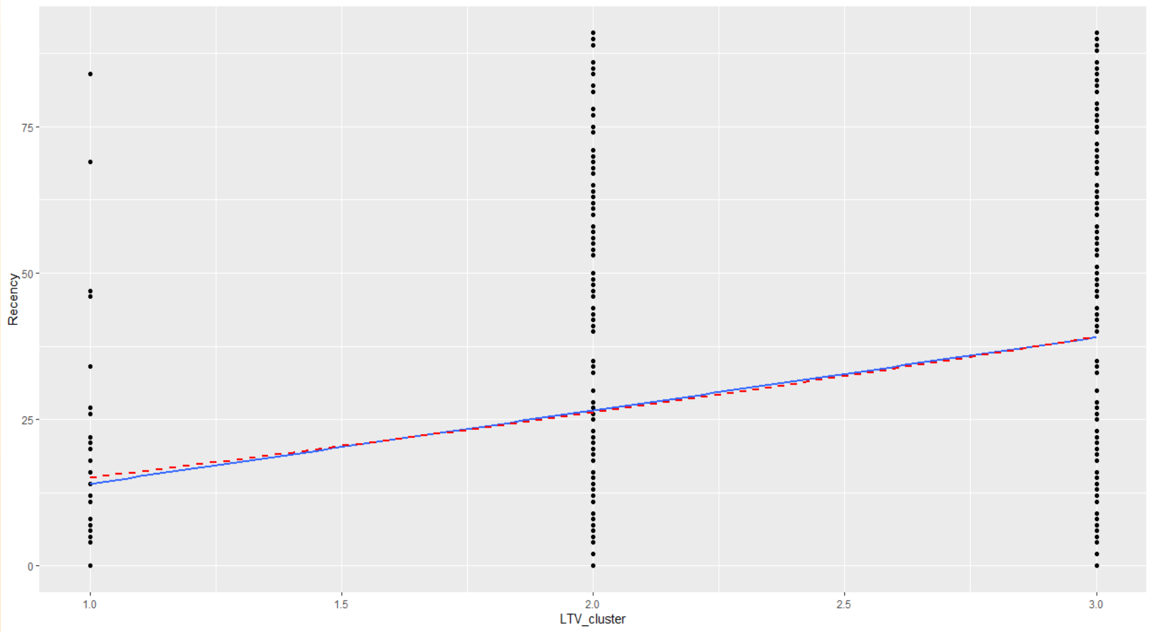
The Gradient Boosting recommends M6_Revenue and Recency will have relative influence on the LTV. Since M6_Revenue is not the future actual data, we can igore and go with Recency.

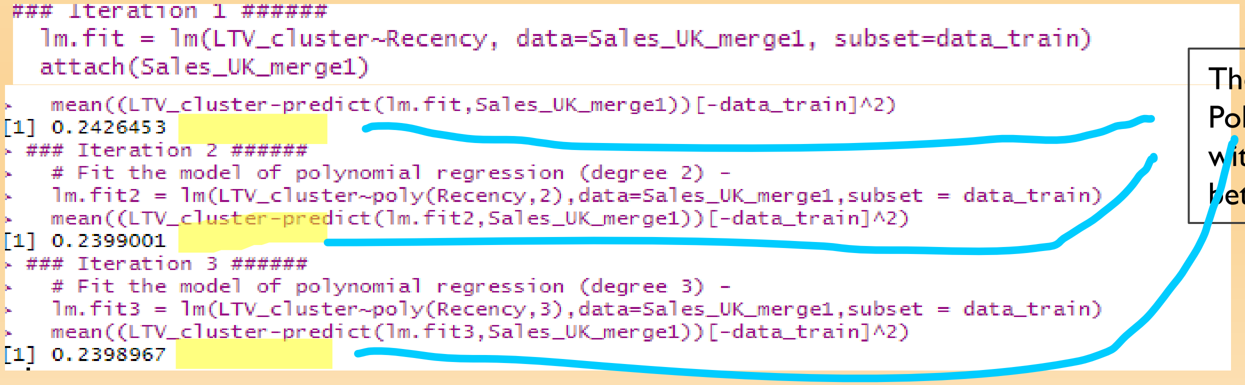
The actual and prediction are very close.

IV - LIFE TIME VALUE (LTV) PREDICTION

- The training and testing data set: 70:30 ratio, Dependent variable: LTV Cluster and Independent variable: Recency
- **Linear Regression - Polynomial - 3 degrees**







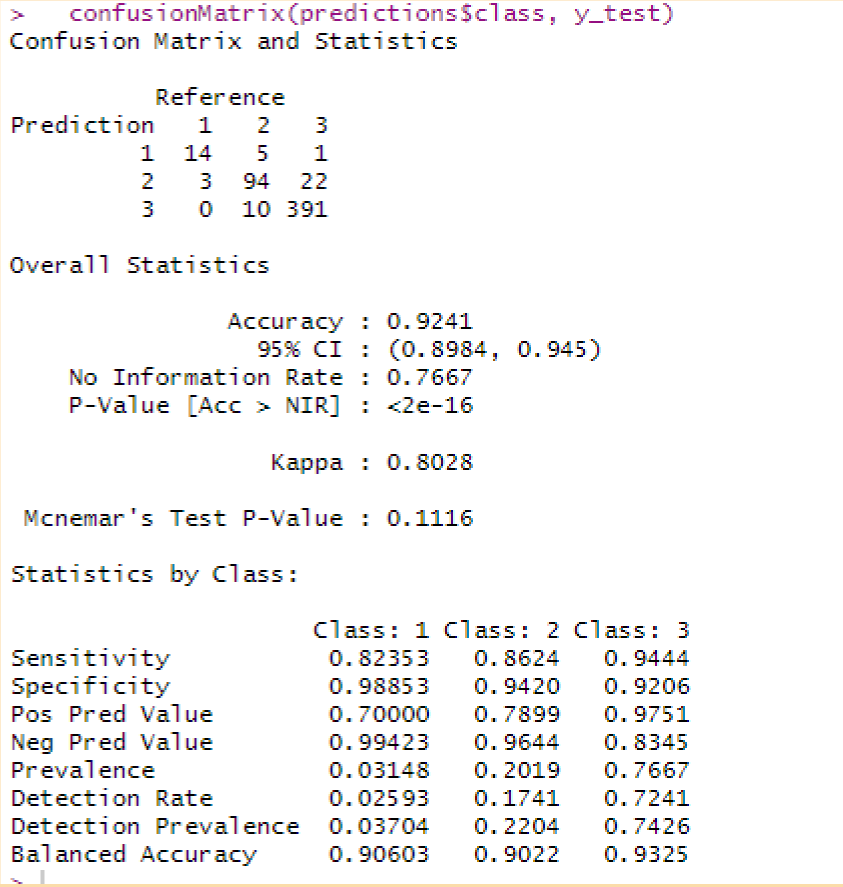
```
### Iteration 1 #######
  lm.fit = lm(LTV_cluster~Recency, data=Sales_UK_merge1, subset=data_train)
  attach(Sales_UK_merge1)
>   mean((LTV_cluster-predict(lm.fit,Sales_UK_merge1))[-data_train]^2)
[1] 0.2426453
> ### Iteration 2 ######
>   # Fit the model of polynomial regression (degree 2) -
>   lm.fit2 = lm(LTV_cluster~poly(Recency,2),data=Sales_UK_merge1,subset = data_train)
>   mean((LTV_cluster-predict(lm.fit2,Sales_UK_merge1))[-data_train]^2)
[1] 0.2399001
> ### Iteration 3 ######
>   # Fit the model of polynomial regression (degree 3) -
>   lm.fit3 = lm(LTV_cluster~poly(Recency,3),data=Sales_UK_merge1,subset = data_train)
>   mean((LTV_cluster-predict(lm.fit3,Sales_UK_merge1))[-data_train]^2)
[1] 0.2398967
```

The mean is very close. Polynomial degree 1 with 0.243 is looking better

- Blue line indicates linear regression model
- Red line - polynomial - degree 2
- Green line - polynomial - degree 3

IV - LIFE TIME VALUE (LTV) PREDICTION

- The training and testing data set: 70:30 ratio, Dependent variable: LTV_Cluster and Independent variable: Recency
- **Naïve Bayes**

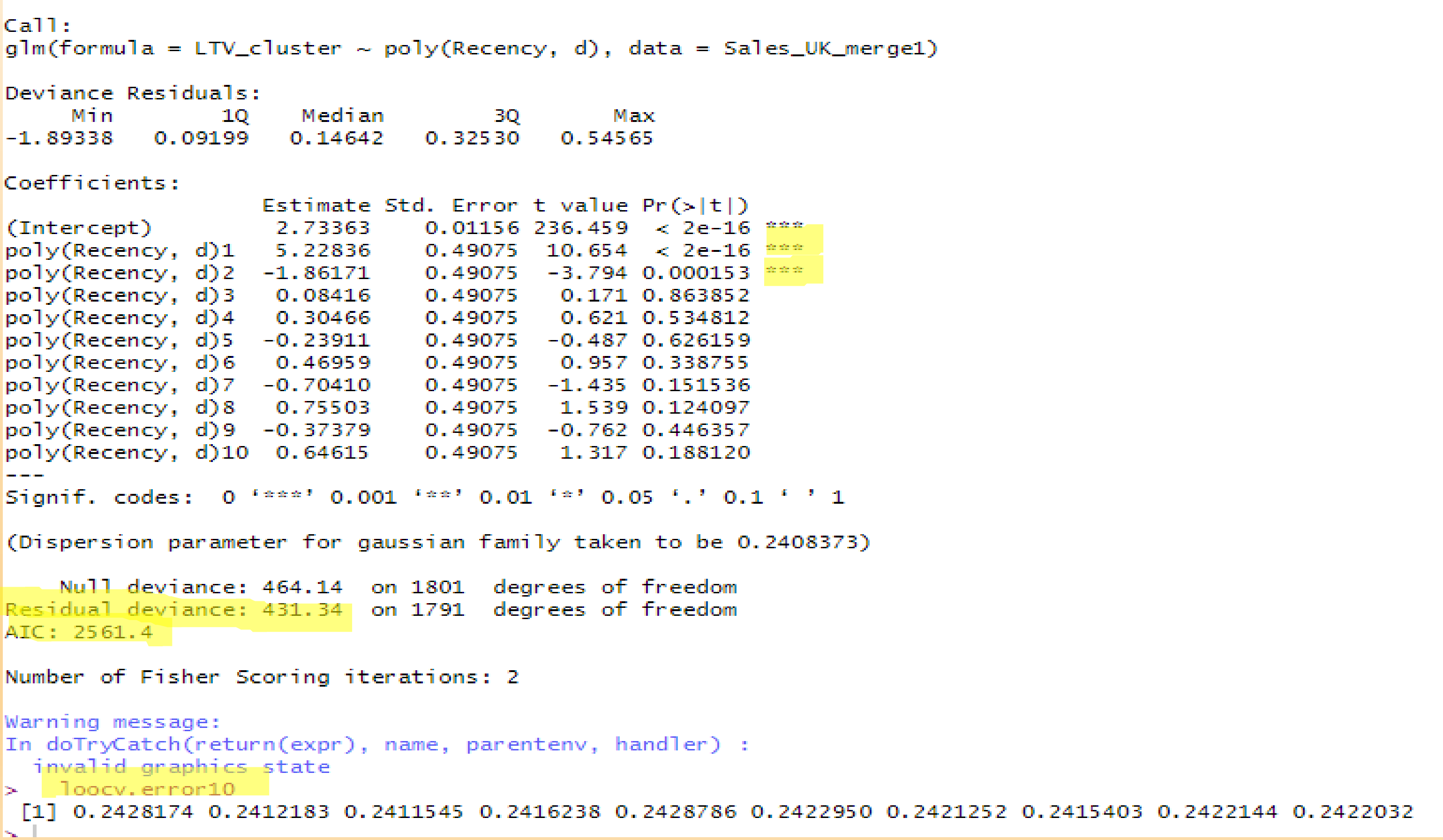


```
>   confusionMatrix(predictions$class, y_test)
Confusion Matrix and Statistics

          Reference
Prediction   1   2   3
         1  14   5   1
         2   3  94  22
         3   0  10 391

Overall Statistics
                                          
               Accuracy : 0.9241          
                 95% CI : (0.8984, 0.945) 
    No Information Rate : 0.7667          
    P-Value [Acc > NIR] : <2e-16          
                                          
                  Kappa : 0.8028          
                                          
 Mcnemar's Test P-Value : 0.1116          

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity           0.82353   0.8624   0.9444
Specificity           0.98853   0.9420   0.9206
Pos Pred Value        0.70000   0.7899   0.9751
Neg Pred Value        0.99423   0.9644   0.8345
Prevalence            0.03148   0.2019   0.7667
Detection Rate        0.02593   0.1741   0.7241
Detection Prevalence  0.03704   0.2204   0.7426
Balanced Accuracy     0.90603   0.9022   0.9325
```

The accuracy is 92%

IV - LIFE TIME VALUE (LTV) PREDICTION

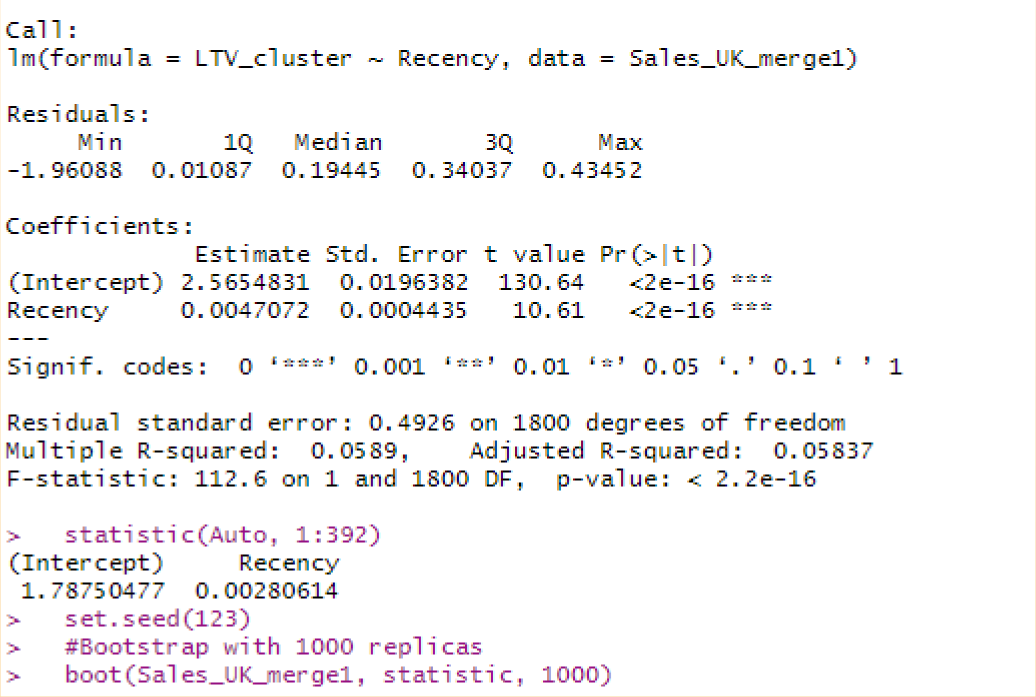
- The training and testing data set: 70:30 ratio,   Dependent variable: LTV_Cluster and Independent variable: Recency
- **LOOCV – 10 fold**



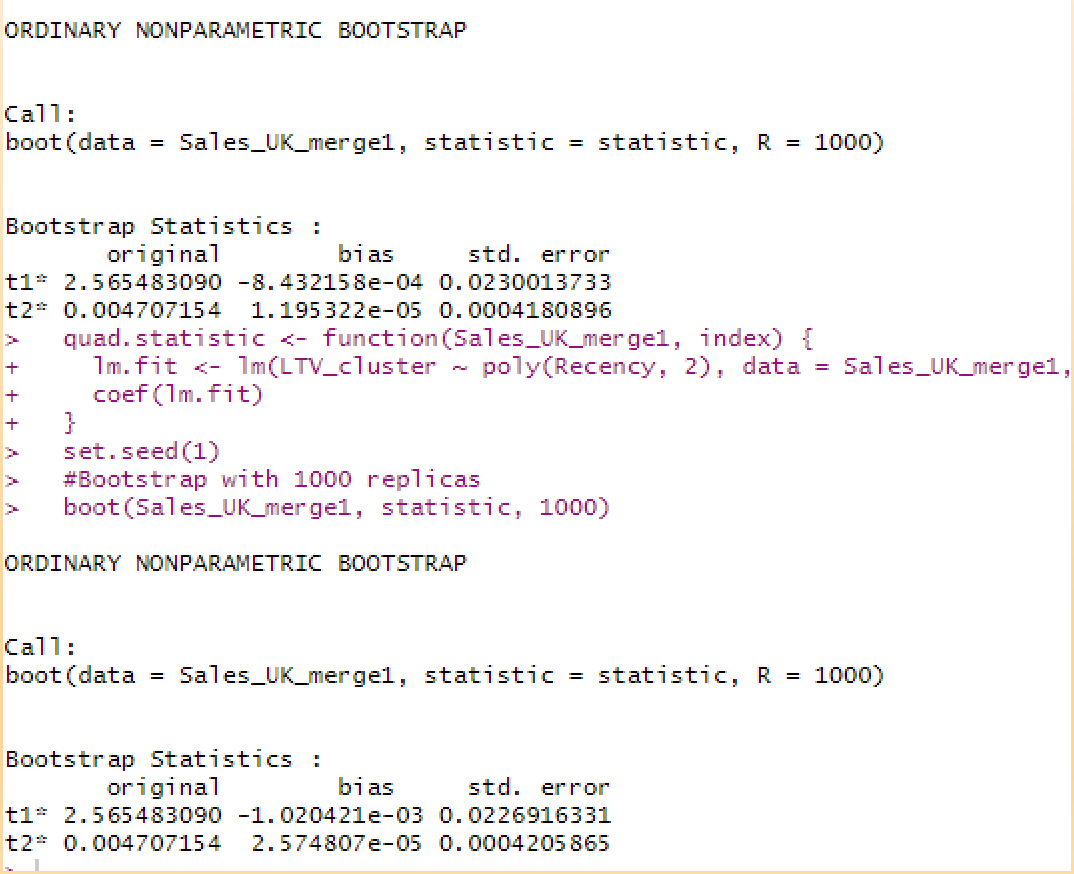
```
Call:
glm(formula = LTV_cluster ~ poly(Recency, d), data = Sales_UK_merge1)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-1.89338   0.09199   0.14642   0.32530   0.54565  

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)         2.73363    0.01156 236.459  < 2e-16 ***
poly(Recency, d)1   5.22836    0.49075  10.654  < 2e-16 ***
poly(Recency, d)2  -1.86171    0.49075  -3.794 0.000153 ***
poly(Recency, d)3   0.08416    0.49075   0.171 0.863852    
poly(Recency, d)4   0.30466    0.49075   0.621 0.534812    
poly(Recency, d)5  -0.23911    0.49075  -0.487 0.626159    
poly(Recency, d)6   0.46959    0.49075   0.957 0.338755    
poly(Recency, d)7  -0.70410    0.49075  -1.435 0.151536    
poly(Recency, d)8   0.75503    0.49075   1.539 0.124097    
poly(Recency, d)9  -0.37379    0.49075  -0.762 0.446357    
poly(Recency, d)10  0.64615    0.49075   1.317 0.188120    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.2408373)

    Null deviance: 464.14  on 1801  degrees of freedom
Residual deviance: 431.34  on 1791  degrees of freedom
AIC: 2561.4

Number of Fisher Scoring iterations: 2

Warning message:
In doTryCatch(return(expr), name, parentenv, handler) :
  invalid graphics state
>   loocv.error10
 [1] 0.2428174 0.2412183 0.2411545 0.2416238 0.2428786 0.2422950 0.2421252 0.2415403 0.2422144 0.2422032
> 
```

IV - LIFE TIME VALUE (LTV) PREDICTION

- The training and testing data set: 70:30 ratio, Dependent variable: LTV_Cluster and Independent variable: Recency
- **Bootstrap**



```
Call:
lm(formula = LTV_cluster ~ Recency, data = Sales_UK_merge1)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.96088  0.01087  0.19445  0.34037  0.43452 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.5654831  0.0196382  130.64   <2e-16 ***
Recency     0.0047072  0.0004435   10.61   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4926 on 1800 degrees of freedom
Multiple R-squared:  0.0589,	Adjusted R-squared:  0.05837 
F-statistic: 112.6 on 1 and 1800 DF,  p-value: < 2.2e-16

>   statistic(Auto, 1:392)
(Intercept)     Recency 
 1.78750477  0.00280614 
>   set.seed(123)
>   #Bootstrap with 1000 replicas
>   boot(Sales_UK_merge1, statistic, 1000)
```



```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Sales_UK_merge1, statistic = statistic, R = 1000)


Bootstrap Statistics :
       original        bias    std. error
t1* 2.565483090 -8.432158e-04 0.0230013733
t2* 0.004707154  1.195322e-05 0.0004180896
>   quad.statistic <- function(Sales_UK_merge1, index) {
+     lm.fit <- lm(LTV_cluster ~ poly(Recency, 2), data = Sales_UK_merge1,
+     coef(lm.fit)
+   }
>   set.seed(1)
>   #Bootstrap with 1000 replicas
>   boot(Sales_UK_merge1, statistic, 1000)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Sales_UK_merge1, statistic = statistic, R = 1000)


Bootstrap Statistics :
       original        bias    std. error
t1* 2.565483090 -1.020421e-03 0.0226916331
t2* 0.004707154  2.574807e-05 0.0004205865
```

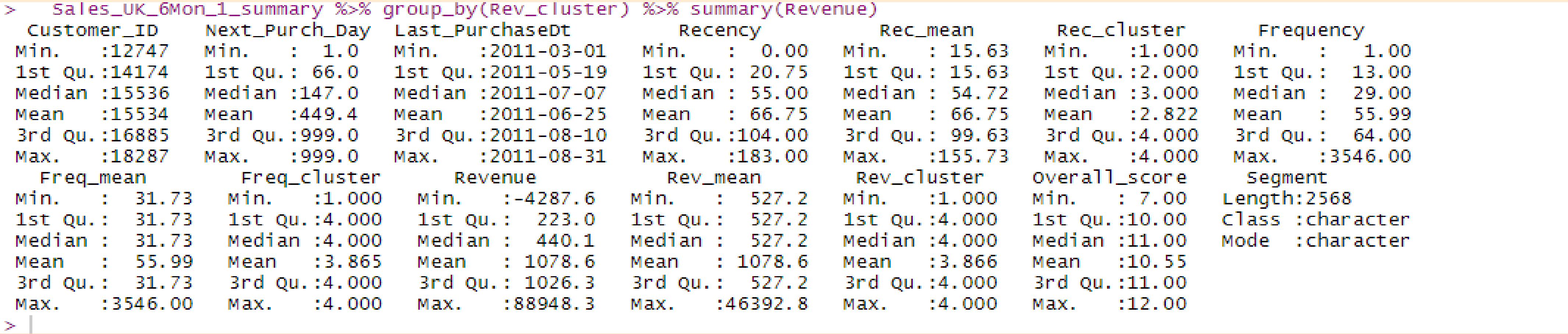
V - PREDICTION OF NEXT PURCHASE DAY

V - PREDICTION OF NEXT PURCHASE DAY

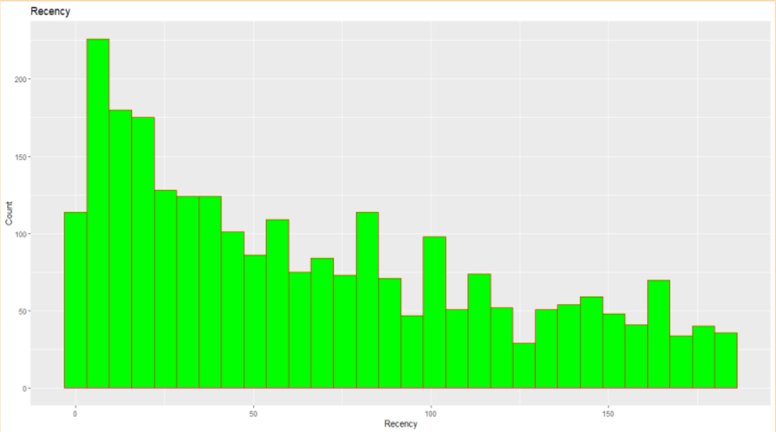
- **Data Preparation**
  - **To implement it correctly, we need to split our dataset.**
    - We will use one 6 months data to identify the purchase pattern of data, calculate RFM and use the last 3 months (Sep-Nov) data to predict. So we need to create two data frames first and append RFM scores to them.
      - Sales_UK_6Mon_1 <- Sales_UK_Data %>% subset(InvoiceDate1 >= "2011-03-01" & InvoiceDate1 < "2011-09-01")
      - Sales_UK_Next <- Sales_UK_Data %>% subset(InvoiceDate1 >= "2011-09-01" & InvoiceDate1 < "2011-12-01")
    - Using 6 months data, calculate Recency, Frequency, and Monetary like before. Apply K-means on each to respective mean, and cluster number. Finally calculate the overall score to segment the customer. Calculate the next Purchase day using the available data
    - Using 3 months data, calculate the Revenue.
    - Merge the RFM dataset created using 3 months data and the Revenue dataset created using 3 months data
    - Apply K-means on the 3 months data Revenue to identify the LTV cluster
- **Correlation**
- **Machine Learning Techniques – Using LTV cluster apply the following:**
  - Gradient Boosting
  - Linear Regression an Polynomial (up to degree 3)
  - Naïve Bayes
  - LOOCV – Cross Validation
  - K Fold Model and Bootstrap

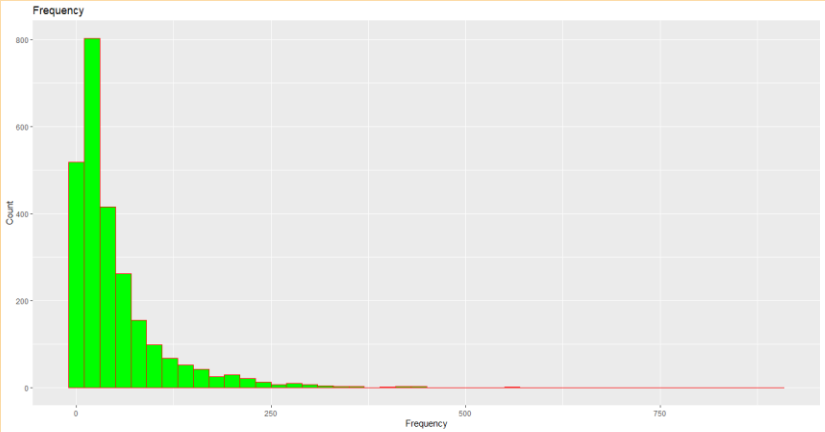
V - PREDICTION OF NEXT PURCHASE DAY

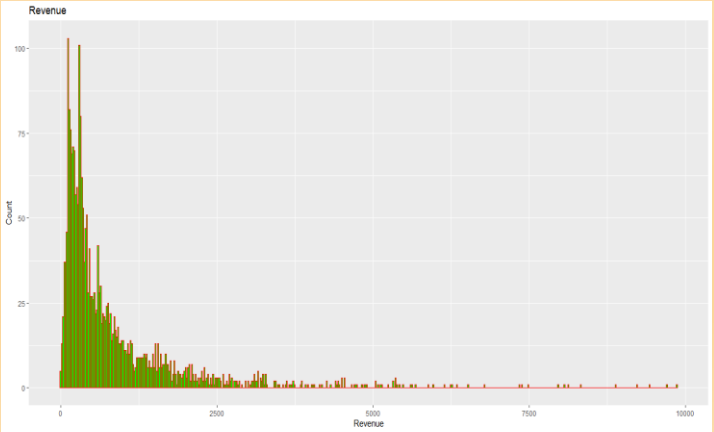
- Using 6 months data, calculate Recency, Frequency, and Monetary like before. Apply K-means on each to respective mean, and cluster number. Finally calculate the overall score to segment the customer



```
>   Sales_UK_6Mon_1_summary %>% group_by(Rev_cluster) %>% summary(Revenue)
  Customer_ID     Next_Purch_Day  Last_PurchaseDt          Recency          Rec_mean        Rec_cluster      Frequency     
 Min.   :12747   Min.   :  1.0   Min.   :2011-03-01   Min.   :  0.00   Min.   : 15.63   Min.   :1.000   Min.   :   1.00  
 1st Qu.:14174   1st Qu.: 66.0   1st Qu.:2011-05-19   1st Qu.: 20.75   1st Qu.: 15.63   1st Qu.:2.000   1st Qu.:  13.00  
 Median :15536   Median :147.0   Median :2011-07-07   Median : 55.00   Median : 54.72   Median :3.000   Median :  29.00  
 Mean   :15534   Mean   :449.4   Mean   :2011-06-25   Mean   : 66.75   Mean   : 66.75   Mean   :2.822   Mean   :  55.99  
 3rd Qu.:16885   3rd Qu.:999.0   3rd Qu.:2011-08-10   3rd Qu.:104.00   3rd Qu.: 99.63   3rd Qu.:4.000   3rd Qu.:  64.00  
 Max.   :18287   Max.   :999.0   Max.   :2011-08-31   Max.   :183.00   Max.   :155.73   Max.   :4.000   Max.   :3546.00  
   Freq_mean        Freq_cluster       Revenue            Rev_mean        Rev_cluster    Overall_score     Segment         
 Min.   :  31.73   Min.   :1.000   Min.   :-4287.6   Min.   :  527.2   Min.   :1.000   Min.   : 7.00   Length:2568       
 1st Qu.:  31.73   1st Qu.:4.000   1st Qu.:  223.0   1st Qu.:  527.2   1st Qu.:4.000   1st Qu.:10.00   Class :character  
 Median :  31.73   Median :4.000   Median :  440.1   Median :  527.2   Median :4.000   Median :11.00   Mode  :character  
 Mean   :  55.99   Mean   :3.865   Mean   : 1078.6   Mean   : 1078.6   Mean   :3.866   Mean   :10.55                     
 3rd Qu.:  31.73   3rd Qu.:4.000   3rd Qu.: 1026.3   3rd Qu.:  527.2   3rd Qu.:4.000   3rd Qu.:11.00                     
 Max.   :3546.00   Max.   :4.000   Max.   :88948.3   Max.   :46392.8   Max.   :4.000   Max.   :12.00                     
> 
```

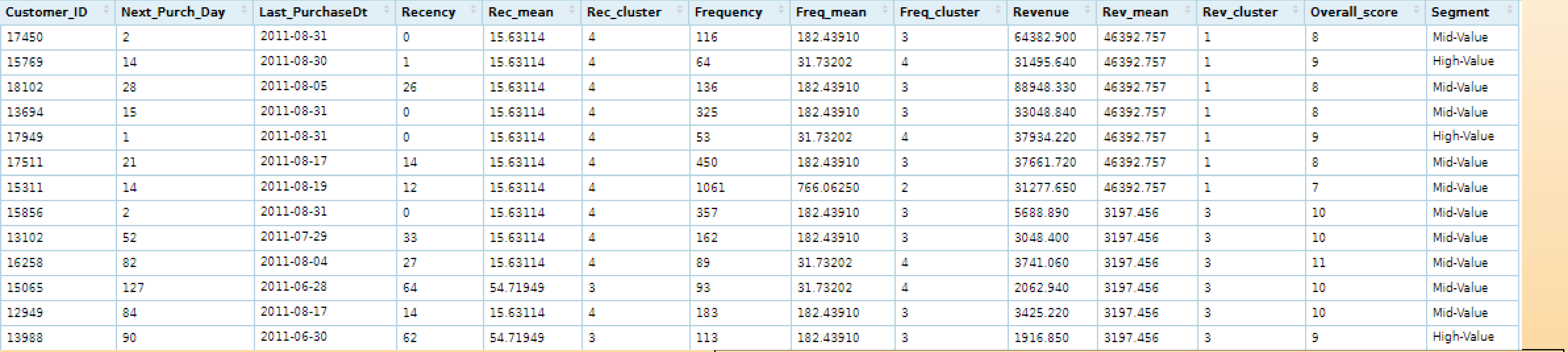




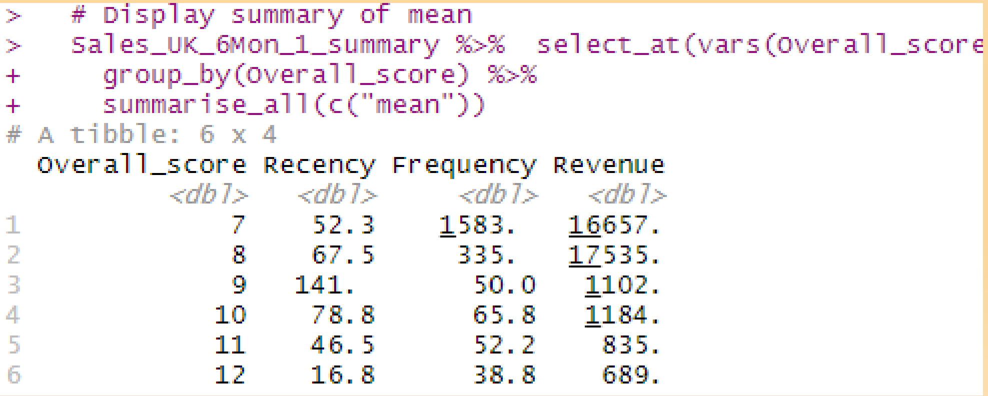


V - PREDICTION OF NEXT PURCHASE DAY

- Using 6 months data, calculate Recency, Frequency, and Monetary like before. Apply K-means on each to respective mean, and cluster number. Finally calculate the overall score to segment the customer



Customer_ID	Next_Purch_Day	Last_PurchaseDt	Recency	Rec_mean	Rec_cluster	Frequency	Freq_mean	Freq_cluster	Revenue	Rev_mean	Rev_cluster	Overall_score	Segment
17450	2	2011-08-31	0	15.63114	4	116	182.43910	3	64382.900	46392.757	1	8	Mid-Value
15769	14	2011-08-30	1	15.63114	4	64	31.73202	4	31495.640	46392.757	1	9	High-Value
18102	28	2011-08-05	26	15.63114	4	136	182.43910	3	88948.330	46392.757	1	8	Mid-Value
13694	15	2011-08-31	0	15.63114	4	325	182.43910	3	33048.840	46392.757	1	8	Mid-Value
17949	1	2011-08-31	0	15.63114	4	53	31.73202	4	37934.220	46392.757	1	9	High-Value
17511	21	2011-08-17	14	15.63114	4	450	182.43910	3	37661.720	46392.757	1	8	Mid-Value
15311	14	2011-08-19	12	15.63114	4	1061	766.06250	2	31277.650	46392.757	1	7	Mid-Value
15856	2	2011-08-31	0	15.63114	4	357	182.43910	3	5688.890	3197.456	3	10	Mid-Value
13102	52	2011-07-29	33	15.63114	4	162	182.43910	3	3048.400	3197.456	3	10	Mid-Value
16258	82	2011-08-04	27	15.63114	4	89	31.73202	4	3741.060	3197.456	3	11	Mid-Value
15065	127	2011-06-28	64	54.71949	3	93	31.73202	4	2062.940	3197.456	3	10	Mid-Value
12949	84	2011-08-17	14	15.63114	4	183	182.43910	3	3425.220	3197.456	3	10	Mid-Value
13988	90	2011-06-30	62	54.71949	3	113	182.43910	3	1916.850	3197.456	3	9	High-Value

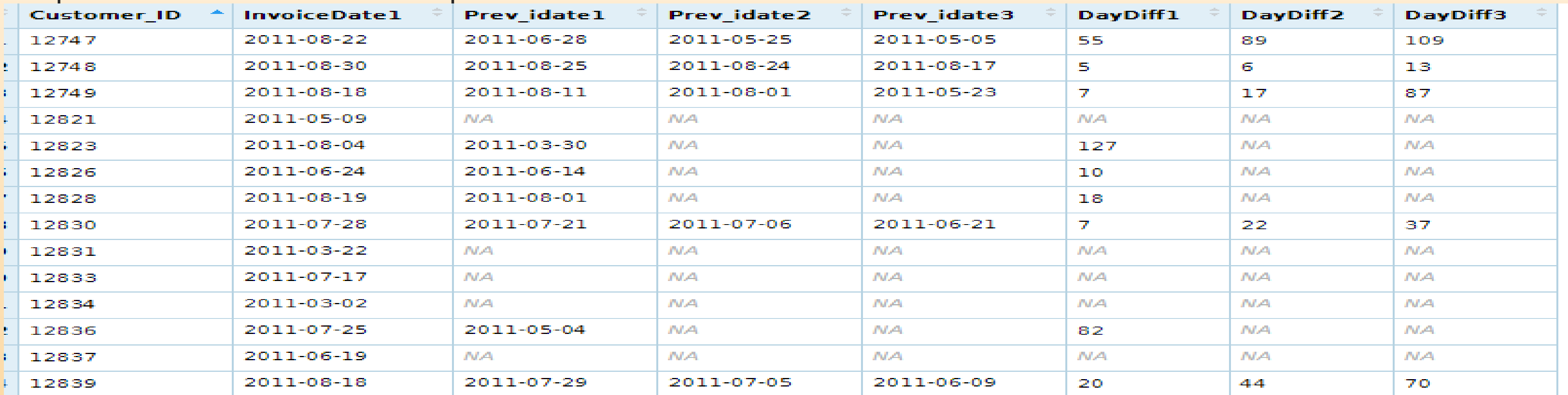


```
>   # Display summary of mean
>   Sales_UK_6Mon_1_summary %>%  select_at(vars(Overall_score
+     group_by(Overall_score) %>%
+     summarise_all(c("mean"))
# A tibble: 6 x 4
  Overall_score Recency Frequency Revenue
          <dbl>   <dbl>     <dbl>   <dbl>
1             7    52.3     1583.  16657.
2             8    67.5      335.  17535.
3             9   141.        50.0  1102.
4            10    78.8       65.8  1184.
5            11    46.5       52.2   835.
6            12    16.8       38.8   689.
```

```
Sales_UK_3Mon_summary$Segment <- 'Low-Value'
Sales_UK_3Mon_summary$Segment[between(Sales_UK_3Mon_summary$Overall_score,7,8)] <- 'High-Value’
Sales_UK_3Mon_summary$Segment[between(Sales_UK_3Mon_summary$Overall_score,10,11)] <- 'High-Value'
Sales_UK_3Mon_summary$Segment[Sales_UK_3Mon_summary$Overall_score == 9] <- 'Mid-Value'
```

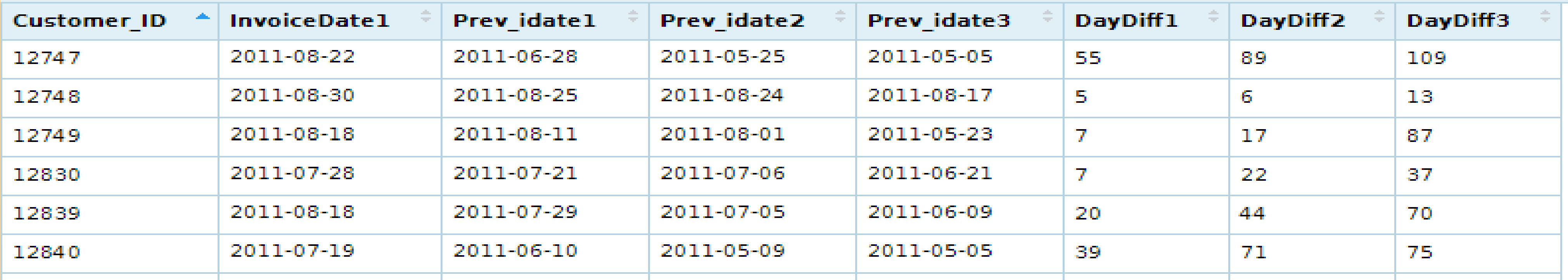
V - PREDICTION OF NEXT PURCHASE DAY

- Using the invoice date, find out the last 3 purchased dates for each customer, and calculate the difference between the last invoice date and 3 prev purhcases. NA means there is no purchase. Shift function is used here.



Customer_ID	InvoiceDate1	Prev_idate1	Prev_idate2	Prev_idate3	DayDiff1	DayDiff2	DayDiff3
12747	2011-08-22	2011-06-28	2011-05-25	2011-05-05	55	89	109
12748	2011-08-30	2011-08-25	2011-08-24	2011-08-17	5	6	13
12749	2011-08-18	2011-08-11	2011-08-01	2011-05-23	7	17	87
12821	2011-05-09	NA	NA	NA	NA	NA	NA
12823	2011-08-04	2011-03-30	NA	NA	127	NA	NA
12826	2011-06-24	2011-06-14	NA	NA	10	NA	NA
12828	2011-08-19	2011-08-01	NA	NA	18	NA	NA
12830	2011-07-28	2011-07-21	2011-07-06	2011-06-21	7	22	37
12831	2011-03-22	NA	NA	NA	NA	NA	NA
12833	2011-07-17	NA	NA	NA	NA	NA	NA
12834	2011-03-02	NA	NA	NA	NA	NA	NA
12836	2011-07-25	2011-05-04	NA	NA	82	NA	NA
12837	2011-06-19	NA	NA	NA	NA	NA	NA
12839	2011-08-18	2011-07-29	2011-07-05	2011-06-09	20	44	70

- Drop records with NA to get the clean purchase history of customers. This helps in predicting the next purchase day.



Customer_ID	InvoiceDate1	Prev_idate1	Prev_idate2	Prev_idate3	DayDiff1	DayDiff2	DayDiff3
12747	2011-08-22	2011-06-28	2011-05-25	2011-05-05	55	89	109
12748	2011-08-30	2011-08-25	2011-08-24	2011-08-17	5	6	13
12749	2011-08-18	2011-08-11	2011-08-01	2011-05-23	7	17	87
12830	2011-07-28	2011-07-21	2011-07-06	2011-06-21	7	22	37
12839	2011-08-18	2011-07-29	2011-07-05	2011-06-09	20	44	70
12840	2011-07-19	2011-06-10	2011-05-09	2011-05-05	39	71	75

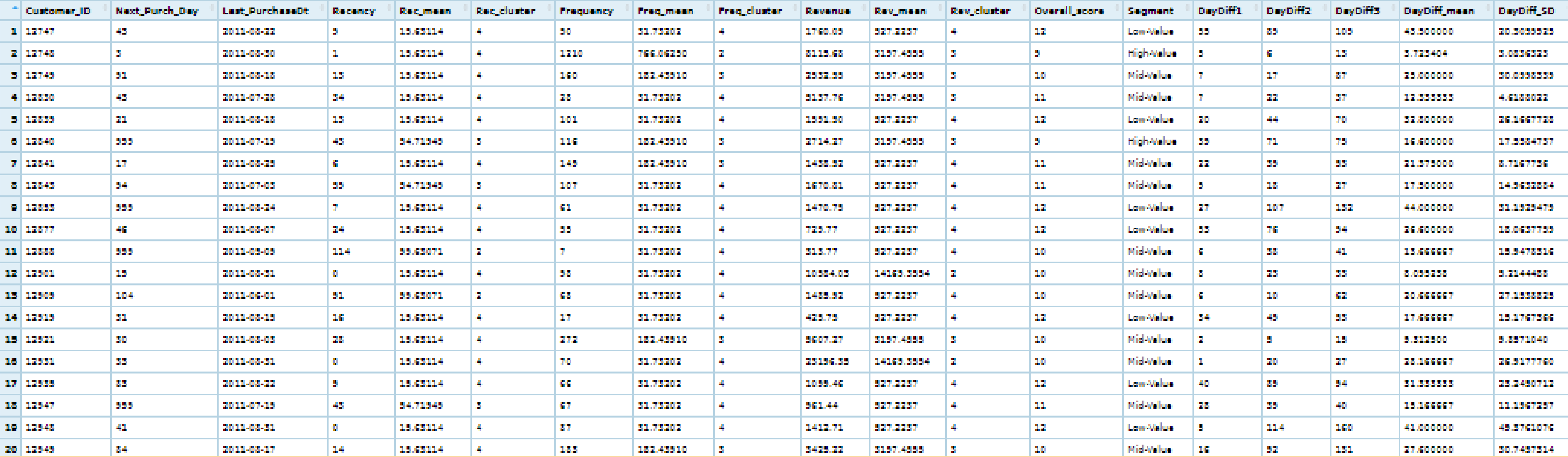
V - PREDICTION OF NEXT PURCHASE DAY

- Merge all the columns to get the complete view of RFM, Last Purchase date, Next Purchase day, difference between the last purchase date and previous purchase dates and their associated Mean and SD

Next purchase day

RFM score

Details of last 3 purchase dates



	Customer_ID	Next_Purch_Day	Last_PurchaseDt	Recency	Rec_mean	Rec_cluster	Frequency	Freq_mean	Freq_cluster	Revenue	Rev_mean	Rev_cluster	Overall_score	Segment	DayDiff1	DayDiff2	DayDiff3	DayDiff_mean	DayDiff_SD
1	12747	43	2011-08-22	9	15.65114	4	90	31.75202	4	1760.09	927.2237	4	12	Low-Value	55	85	109	43.500000	20.5035525
2	12748	3	2011-08-30	1	15.65114	4	1210	766.06250	2	8115.68	3197.4555	3	9	High-Value	3	6	13	3.723404	3.0856323
3	12749	51	2011-08-18	13	15.65114	4	160	182.43910	3	2932.55	3197.4555	3	10	Mid-Value	7	17	87	29.000000	30.0598339
4	12830	43	2011-07-28	34	15.65114	4	28	31.75202	4	5137.76	3197.4555	3	11	Mid-Value	7	22	37	12.333333	4.6188022
5	12839	21	2011-08-18	13	15.65114	4	101	31.75202	4	1591.50	927.2237	4	12	Low-Value	20	44	70	32.800000	26.1667728
6	12840	999	2011-07-19	43	54.71949	3	116	182.43910	3	2714.27	3197.4555	3	9	High-Value	39	71	75	16.600000	17.5584737
7	12841	17	2011-08-25	6	15.65114	4	149	182.43910	3	1438.52	927.2237	4	11	Mid-Value	22	39	93	21.375000	8.7167736
8	12843	94	2011-07-03	59	54.71949	3	107	31.75202	4	1670.81	927.2237	4	11	Mid-Value	9	18	27	17.500000	14.9632884
9	12853	999	2011-08-24	7	15.65114	4	61	31.75202	4	1470.79	927.2237	4	12	Low-Value	27	107	132	44.000000	51.1929479
10	12877	46	2011-08-07	24	15.65114	4	59	31.75202	4	729.77	927.2237	4	12	Low-Value	53	76	94	26.600000	18.0637759
11	12888	999	2011-05-09	114	99.65071	2	7	31.75202	4	313.77	927.2237	4	10	Mid-Value	6	38	41	13.666667	19.5478316
12	12901	19	2011-08-31	0	15.65114	4	98	31.75202	4	10984.03	14169.3594	2	10	Mid-Value	8	23	33	8.095238	9.2144488
13	12909	104	2011-06-01	91	99.65071	2	68	31.75202	4	1489.92	927.2237	4	10	Mid-Value	6	10	62	20.666667	27.1558825
14	12919	31	2011-08-15	16	15.65114	4	17	31.75202	4	429.79	927.2237	4	12	Low-Value	34	49	93	17.666667	19.1767366
15	12921	30	2011-08-03	28	15.65114	4	272	182.43910	3	9607.27	3197.4555	3	10	Mid-Value	2	9	19	9.312500	9.8571040
16	12931	33	2011-08-31	0	15.65114	4	70	31.75202	4	23196.39	14169.3594	2	10	Mid-Value	1	20	27	28.166667	26.9177760
17	12933	83	2011-08-22	9	15.65114	4	66	31.75202	4	1099.46	927.2237	4	12	Low-Value	40	85	94	31.333333	23.2480712
18	12947	999	2011-07-19	43	54.71949	3	67	31.75202	4	961.44	927.2237	4	11	Mid-Value	28	39	40	19.166667	11.1967297
19	12948	41	2011-08-31	0	15.65114	4	87	31.75202	4	1412.71	927.2237	4	12	Low-Value	9	114	160	41.000000	49.5761076
20	12949	84	2011-08-17	14	15.65114	4	183	182.43910	3	3429.22	3197.4555	3	10	Mid-Value	16	92	131	27.600000	30.7457314

Segment

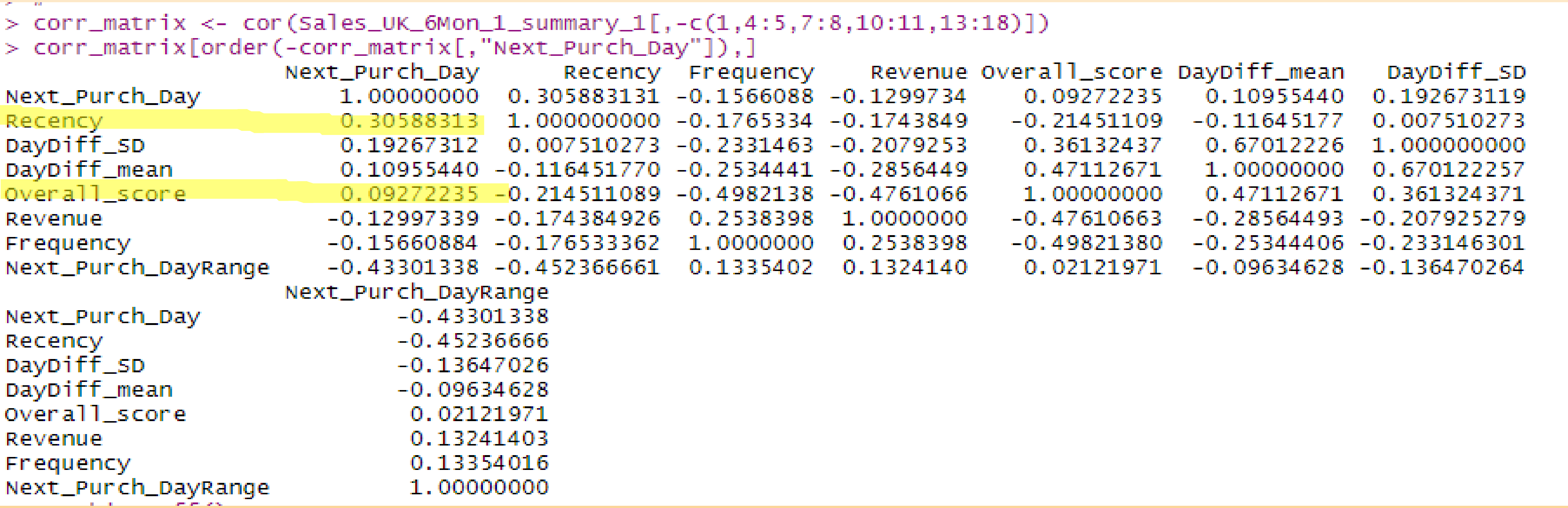
V - PREDICTION OF NEXT PURCHASE DAY

- Using the Next purchase day, classify customers under class name 0 to 2 as follows:
  - Class name: 2 => Customers will purchase in the next 0-20 days
  - Class name: 1 => Customers will purchase in the next 21-49 days
  - Class name: 0 => Customers will purchase in the next >= 50 days

```
# Categorize the next purchase day into three to take action and communicate. These boundaries can be
# modified as per business needs
#---------------------------------------------------------------------------------------------
#  Class name: 2 => Customers will purchase in the next 0-20 days
#  Class name: 1 => Customers will purchase in the next 21-49 days
#  Class name: 0 => Customers will purchase in the next >= 50 days
#---------------------------------------------------------------------------------------------
Sales_UK_6Mon_1_summary_1$Next_Purch_DayRange <- 0
Sales_UK_6Mon_1_summary_1$Next_Purch_DayRange[between(Sales_UK_6Mon_1_summary_1$Next_Purch_Day,21,49)] <- 2
Sales_UK_6Mon_1_summary_1$Next_Purch_DayRange[between(Sales_UK_6Mon_1_summary_1$Next_Purch_Day,0,20)] <- 1
```

V - PREDICTION OF NEXT PURCHASE DAY

- **Correlation**
  - **Feature Engineering -** convert categorical columns to numerical columns using dummy.data.frame function
  - Apply Correlation to see the relative influence between LTV Cluster and other variables

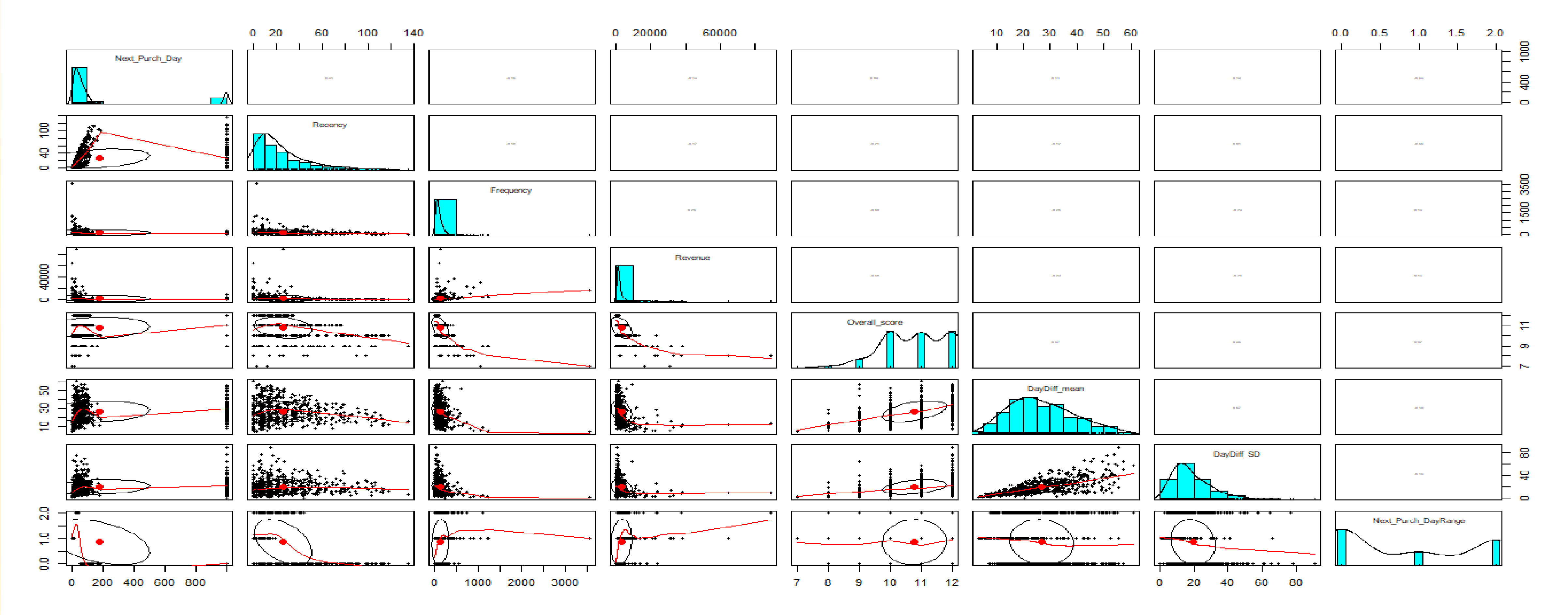


```
> corr_matrix <- cor(Sales_UK_6Mon_1_summary_1[,-c(1,4:5,7:8,10:11,13:18)])
> corr_matrix[order(-corr_matrix[,"Next_Purch_Day"]),]
                   Next_Purch_Day      Recency  Frequency    Revenue Overall_score DayDiff_mean   DayDiff_SD
Next_Purch_Day         1.00000000  0.305883131 -0.1566088 -0.1299734    0.09272235   0.10955440  0.192673119
Recency                0.30588313  1.000000000 -0.1765334 -0.1743849   -0.21451109  -0.11645177  0.007510273
DayDiff_SD             0.19267312  0.007510273 -0.2331463 -0.2079253    0.36132437   0.67012226  1.000000000
DayDiff_mean           0.10955440 -0.116451770 -0.2534441 -0.2856449    0.47112671   1.00000000  0.670122257
Overall_score          0.09272235 -0.214511089 -0.4982138 -0.4761066    1.00000000   0.47112671  0.361324371
Revenue               -0.12997339 -0.174384926  0.2538398  1.0000000   -0.47610663  -0.28564493 -0.207925279
Frequency             -0.15660884 -0.176533362  1.0000000  0.2538398   -0.49821380  -0.25344406 -0.233146301
Next_Purch_DayRange   -0.43301338 -0.452366661  0.1335402  0.1324140    0.02121971  -0.09634628 -0.136470264
                   Next_Purch_DayRange
Next_Purch_Day             -0.43301338
Recency                    -0.45236666
DayDiff_SD                 -0.13647026
DayDiff_mean               -0.09634628
Overall_score               0.02121971
Revenue                     0.13241403
Frequency                   0.13354016
Next_Purch_DayRange         1.00000000
```

Inference: In the above, only Recency and Overall score are very correlating to other variables.

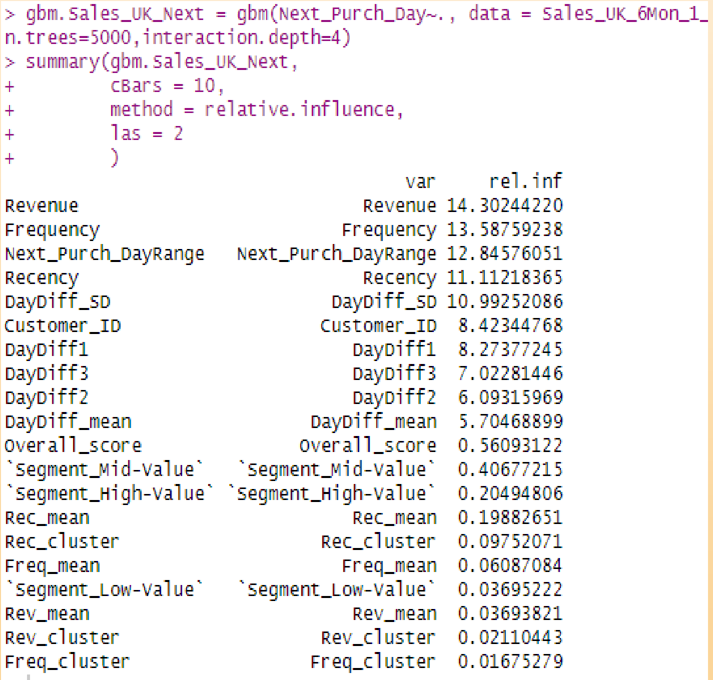
V - PREDICTION OF NEXT PURCHASE DAY

- **Correlation –** using Pearson method. This confirms the correlation reported by the Correlation matrix.

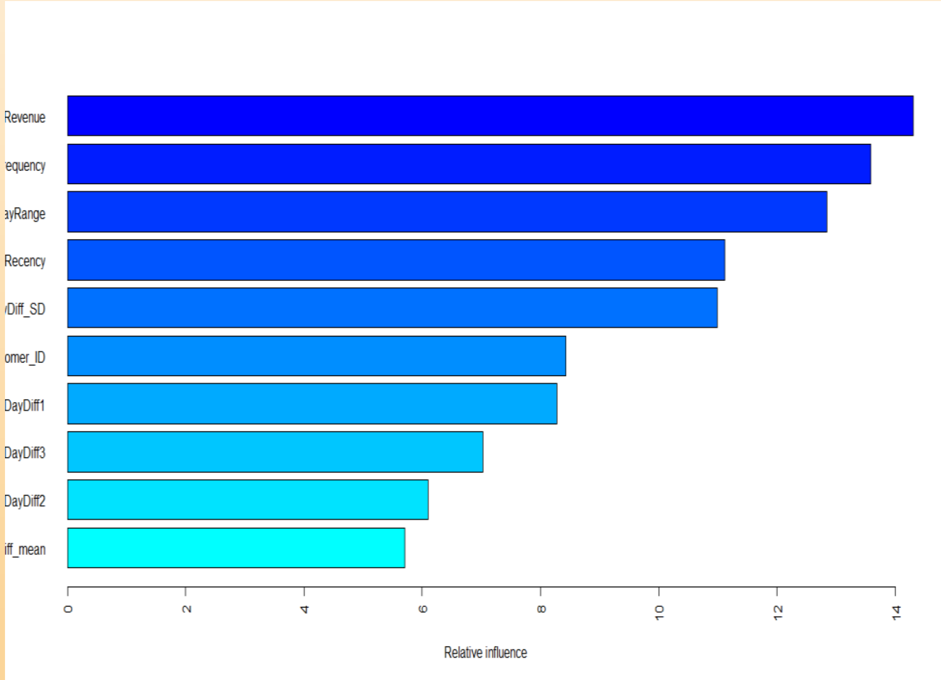


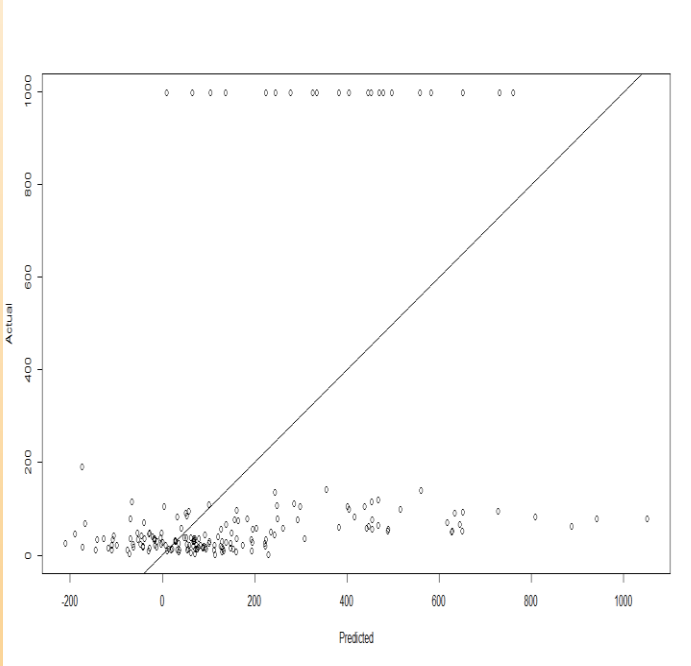
V - PREDICTION OF NEXT PURCHASE DAY

- The training and testing data set: 70:30 ratio
- **Gradient Boosting** - using n.trees = 5000, distribution="gaussian“, and interaction.depth=4



```
> gbm.Sales_UK_Next = gbm(Next_Purch_Day~., data = Sales_UK_6Mon_1_
n.trees=5000,interaction.depth=4)
> summary(gbm.Sales_UK_Next,
+         cBars = 10,
+         method = relative.influence,
+         las = 2
+         )
                                     var     rel.inf
Revenue                          Revenue 14.30244220
Frequency                      Frequency 13.58759238
Next_Purch_DayRange   Next_Purch_DayRange 12.84576051
Recency                          Recency 11.11218365
DayDiff_SD                    DayDiff_SD 10.99252086
Customer_ID                  Customer_ID  8.42344768
DayDiff1                        DayDiff1  8.27377245
DayDiff3                        DayDiff3  7.02281446
DayDiff2                        DayDiff2  6.09315969
DayDiff_mean                DayDiff_mean  5.70468899
overall_score              overall_score  0.56093122
`Segment_Mid-Value`   `Segment_Mid-Value`  0.40677215
`Segment_High-Value` `Segment_High-Value`  0.20494806
Rec_mean                        Rec_mean  0.19882651
Rec_cluster                  Rec_cluster  0.09752071
Freq_mean                      Freq_mean  0.06087084
`Segment_Low-Value`   `Segment_Low-Value`  0.03695222
Rev_mean                        Rev_mean  0.03693821
Rev_cluster                  Rev_cluster  0.02110443
Freq_cluster                Freq_cluster  0.01675279
```



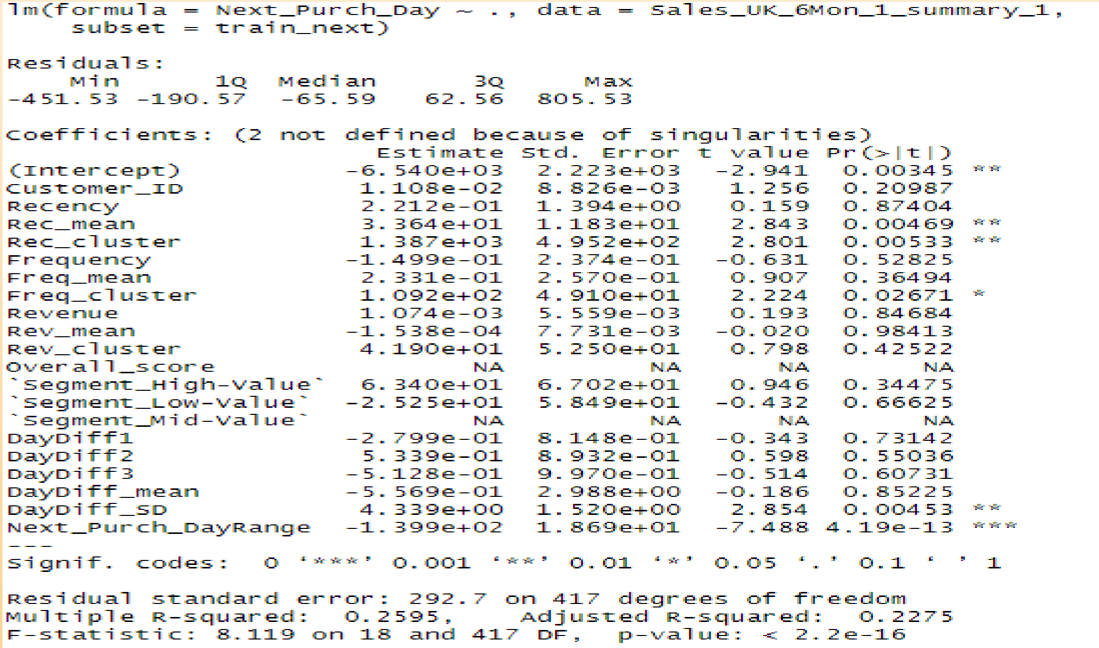


The Gradient Boosting recommends 6 mo. Revenue and Frequency will have relative influence on the next purchase day. We will go with the Revenue

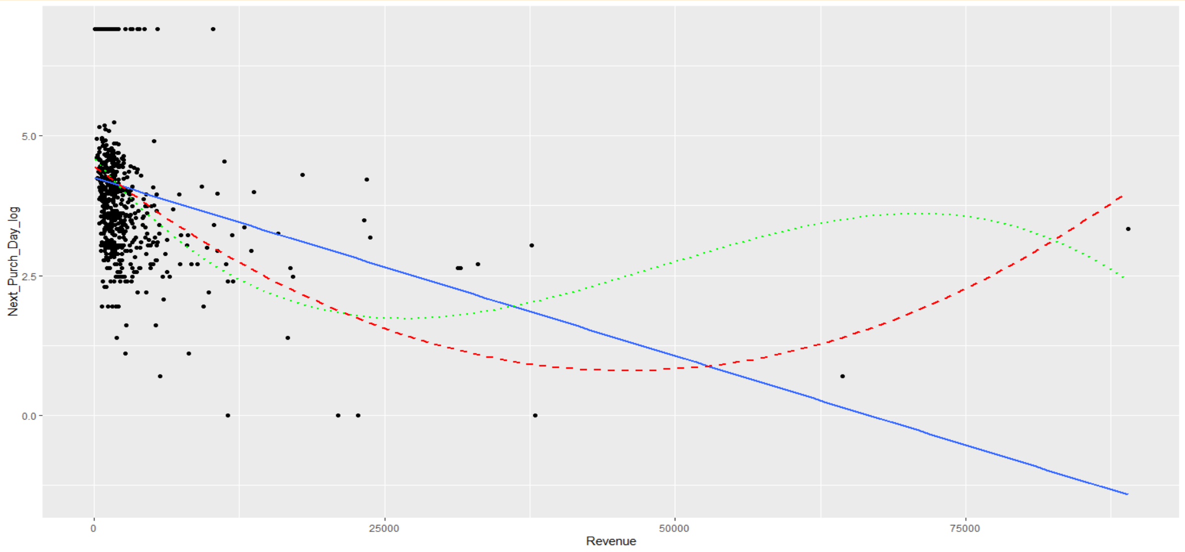
Couldn’t figure out the results

V - PREDICTION OF NEXT PURCHASE DAY

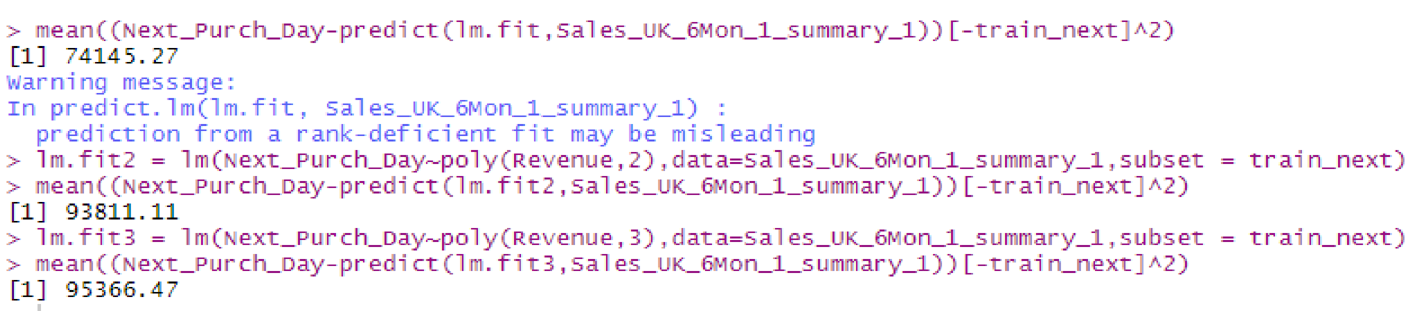
- The training and testing data set: 70:30 ratio,   Dependent variable: Next_Purchase_Day and Independent variable: Revenue
- **Linear Regression - Polynomial - 3 degrees**



```
lm(formula = Next_Purch_Day ~ ., data = Sales_UK_6Mon_1_summary_1,
    subset = train_next)

Residuals:
    Min      1Q  Median      3Q     Max
-451.53 -190.57  -65.59   62.56  805.53

Coefficients: (2 not defined because of singularities)
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)          -6.540e+03  2.223e+03  -2.941  0.00345 **
Customer_ID           1.108e-02  8.826e-03   1.256  0.20987
Recency               2.212e-01  1.394e+00   0.159  0.87404
Rec_mean              3.364e+01  1.183e+01   2.843  0.00469 **
Rec_cluster           1.387e+03  4.952e+02   2.801  0.00533 **
Frequency            -1.499e-01  2.374e-01  -0.631  0.52825
Freq_mean             2.331e-01  2.570e-01   0.907  0.36494
Freq_cluster          1.092e+02  4.910e+01   2.224  0.02671 *
Revenue               1.074e-03  5.559e-03   0.193  0.84684
Rev_mean             -1.538e-04  7.731e-03  -0.020  0.98413
Rev_cluster           4.190e+01  5.250e+01   0.798  0.42522
Overall_score                NA         NA      NA       NA
`Segment_High-Value`  6.340e+01  6.702e+01   0.946  0.34475
`Segment_Low-Value`  -2.525e+01  5.849e+01  -0.432  0.66625
`Segment_Mid-Value`          NA         NA      NA       NA
DayDiff1             -2.799e-01  8.148e-01  -0.343  0.73142
DayDiff2              5.339e-01  8.932e-01   0.598  0.55036
DayDiff3             -5.128e-01  9.970e-01  -0.514  0.60731
DayDiff_mean         -5.569e-01  2.988e+00  -0.186  0.85225
DayDiff_SD            4.339e+00  1.520e+00   2.854  0.00453 **
Next_Purch_DayRange  -1.399e+02  1.869e+01  -7.488 4.19e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 292.7 on 417 degrees of freedom
Multiple R-squared:  0.2595,	Adjusted R-squared:  0.2275
F-statistic: 8.119 on 18 and 417 DF,  p-value: < 2.2e-16
```



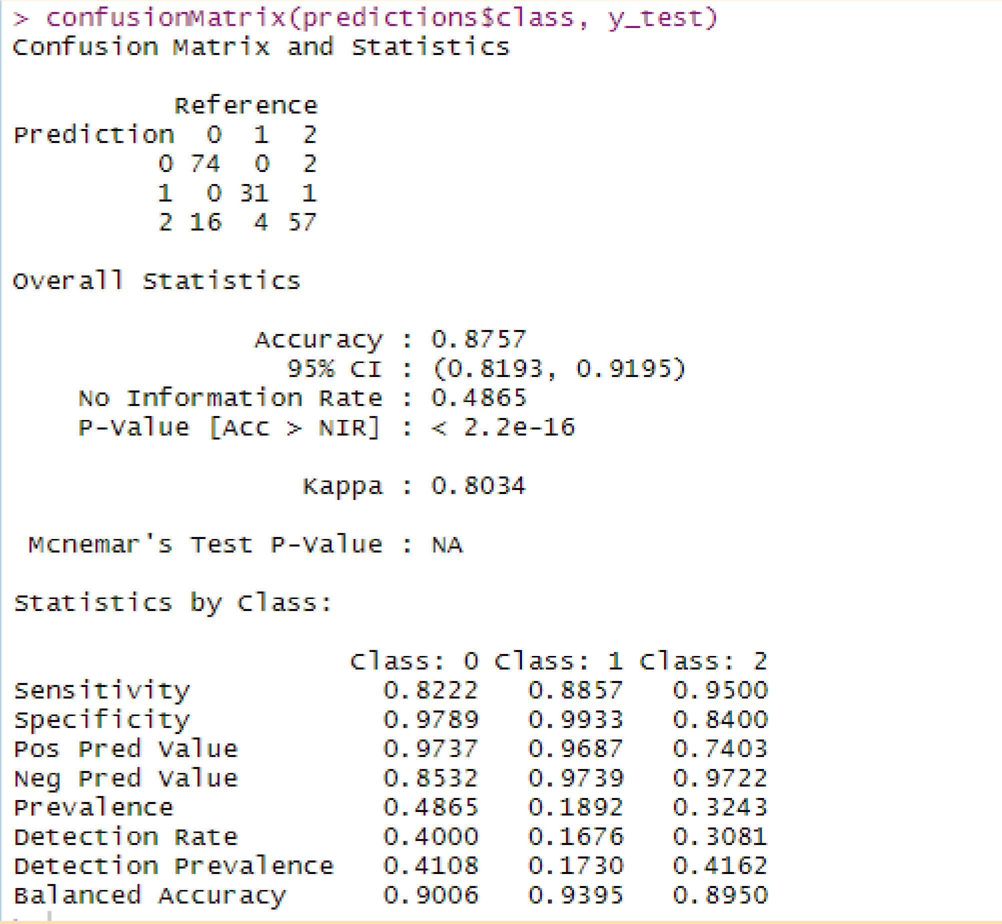
- Blue line indicates linear regression model
- Red line - polynomial - degree 2
- Green line - polynomial - degree 3



```
> mean((Next_Purch_Day-predict(lm.fit,Sales_UK_6Mon_1_summary_1))[-train_next]^2)
[1] 74145.27
Warning message:
In predict.lm(lm.fit, Sales_UK_6Mon_1_summary_1) :
  prediction from a rank-deficient fit may be misleading
> lm.fit2 = lm(Next_Purch_Day~poly(Revenue,2),data=Sales_UK_6Mon_1_summary_1,subset = train_next)
> mean((Next_Purch_Day-predict(lm.fit2,Sales_UK_6Mon_1_summary_1))[-train_next]^2)
[1] 93811.11
> lm.fit3 = lm(Next_Purch_Day~poly(Revenue,3),data=Sales_UK_6Mon_1_summary_1,subset = train_next)
> mean((Next_Purch_Day-predict(lm.fit3,Sales_UK_6Mon_1_summary_1))[-train_next]^2)
[1] 95366.47
```

V - PREDICTION OF NEXT PURCHASE DAY

- The training and testing data set: 70:30 ratio, Dependent variable: Next_Purchase_Day and Independent variable: Revenue
- **Naïve Bayes**

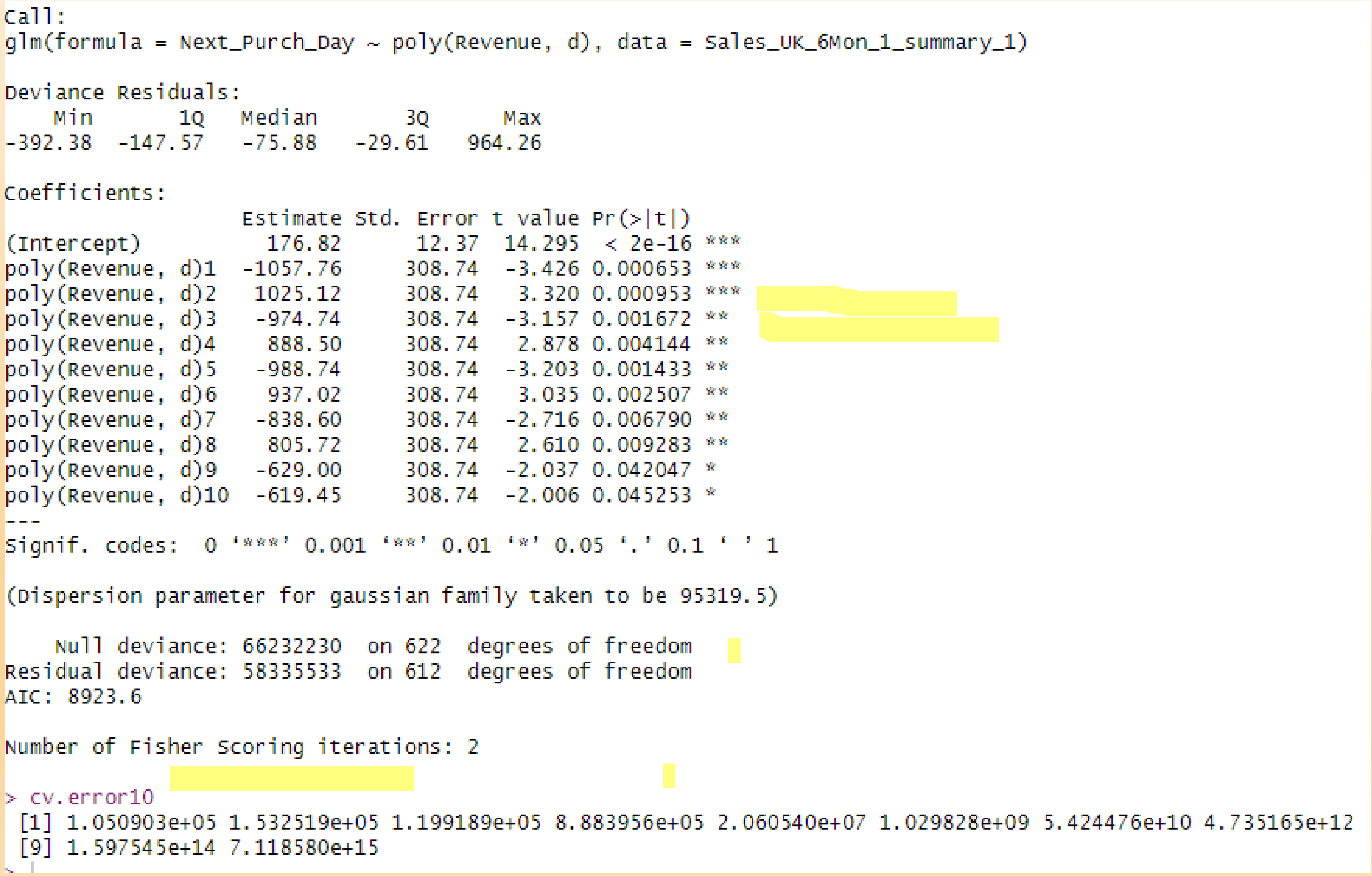


```
> confusionMatrix(predictions$class, y_test)
Confusion Matrix and Statistics

          Reference
Prediction  0  1  2
         0 74  0  2
         1  0 31  1
         2 16  4 57

Overall Statistics

               Accuracy : 0.8757
                 95% CI : (0.8193, 0.9195)
    No Information Rate : 0.4865
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8034

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.8222   0.8857   0.9500
Specificity            0.9789   0.9933   0.8400
Pos Pred Value         0.9737   0.9687   0.7403
Neg Pred Value         0.8532   0.9739   0.9722
Prevalence             0.4865   0.1892   0.3243
Detection Rate         0.4000   0.1676   0.3081
Detection Prevalence   0.4108   0.1730   0.4162
Balanced Accuracy      0.9006   0.9395   0.8950
```

The accuracy is 88%

V - PREDICTION OF NEXT PURCHASE DAY

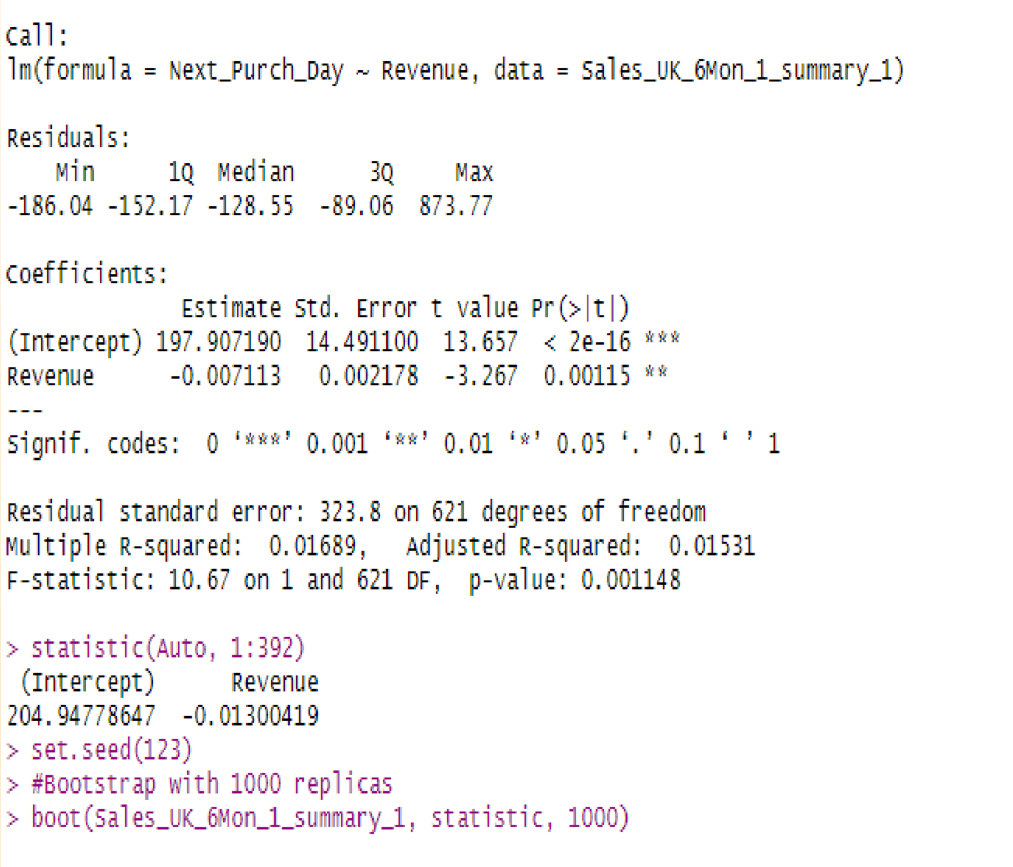
- The training and testing data set: 70:30 ratio,   Dependent variable: Next_Purchase_Day and Independent variable: Revenue
- **LOOCV – 10 fold**



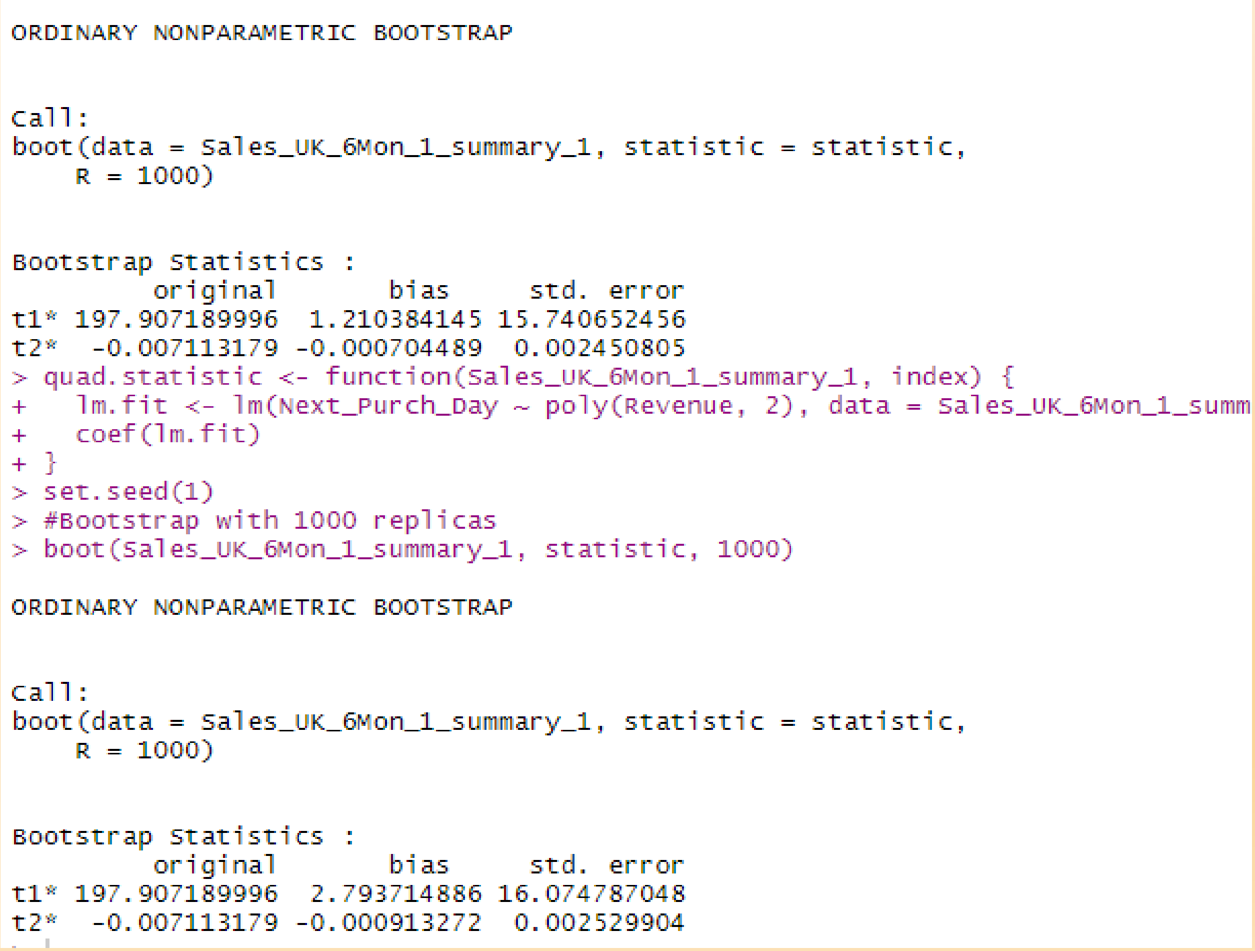
```
Call:
glm(formula = Next_Purch_Day ~ poly(Revenue, d), data = Sales_UK_6Mon_1_summary_1)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-392.38  -147.57   -75.88   -29.61   964.26  

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)          176.82      12.37  14.295  < 2e-16 ***
poly(Revenue, d)1  -1057.76     308.74  -3.426 0.000653 ***
poly(Revenue, d)2   1025.12     308.74   3.320 0.000953 ***
poly(Revenue, d)3   -974.74     308.74  -3.157 0.001672 ** 
poly(Revenue, d)4    888.50     308.74   2.878 0.004144 ** 
poly(Revenue, d)5   -988.74     308.74  -3.203 0.001433 ** 
poly(Revenue, d)6    937.02     308.74   3.035 0.002507 ** 
poly(Revenue, d)7   -838.60     308.74  -2.716 0.006790 ** 
poly(Revenue, d)8    805.72     308.74   2.610 0.009283 ** 
poly(Revenue, d)9   -629.00     308.74  -2.037 0.042047 *  
poly(Revenue, d)10  -619.45     308.74  -2.006 0.045253 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 95319.5)

    Null deviance: 66232230  on 622  degrees of freedom
Residual deviance: 58335533  on 612  degrees of freedom
AIC: 8923.6

Number of Fisher Scoring iterations: 2

> cv.error10
 [1] 1.050903e+05 1.532519e+05 1.199189e+05 8.883956e+05 2.060540e+07 1.029828e+09 5.424476e+10 4.735165e+12
 [9] 1.597545e+14 7.118580e+15
```

V - PREDICTION OF NEXT PURCHASE DAY

- The training and testing data set: 70:30 ratio,   Dependent variable: Next_Purchase_Day and Independent variable: Revenue
- **Bootstrap**



```
Call:
lm(formula = Next_Purch_Day ~ Revenue, data = Sales_UK_6Mon_1_summary_1)

Residuals:
    Min      1Q  Median      3Q     Max 
-186.04 -152.17 -128.55  -89.06  873.77 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 197.907190  14.491100  13.657  < 2e-16 ***
Revenue      -0.007113   0.002178  -3.267  0.00115 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 323.8 on 621 degrees of freedom
Multiple R-squared:  0.01689,	Adjusted R-squared:  0.01531 
F-statistic: 10.67 on 1 and 621 DF,  p-value: 0.001148

> statistic(Auto, 1:392)
 (Intercept)      Revenue 
204.94778647  -0.01300419 
> set.seed(123)
> #Bootstrap with 1000 replicas
> boot(Sales_UK_6Mon_1_summary_1, statistic, 1000)
```



```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Sales_UK_6Mon_1_summary_1, statistic = statistic, 
    R = 1000)


Bootstrap Statistics :
         original        bias     std. error
t1* 197.907189996  1.210384145 15.740652456
t2*  -0.007113179 -0.000704489  0.002450805
> quad.statistic <- function(Sales_UK_6Mon_1_summary_1, index) {
+   lm.fit <- lm(Next_Purch_Day ~ poly(Revenue, 2), data = Sales_UK_6Mon_1_summ
+   coef(lm.fit)
+ }
> set.seed(1)
> #Bootstrap with 1000 replicas
> boot(Sales_UK_6Mon_1_summary_1, statistic, 1000)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Sales_UK_6Mon_1_summary_1, statistic = statistic, 
    R = 1000)


Bootstrap Statistics :
         original        bias     std. error
t1* 197.907189996  2.793714886 16.074787048
t2*  -0.007113179 -0.000913272  0.002529904
```