##### Case Study 1

An ITES company wants to understand its data related to services requests for customers. These requests are related to particular products that the company manufactures. The requests can be divided into 3 priorities: - Low, Medium and High. Each of these priority levels have a different SLA (Service Level Agreement) for resolution. The data consists of the following fields :-

1 ServiceRequestNo:- Unique id for each Service request logged on the system

2 ServiceRequestStatus:- As of date status on the Service request

3 TypeOfEnagagement:- What type of work is the company doing for the client

4 Incident/ Problem:- Is the Service request a problem?

5 SR Priority:- What is the priority level of the Service Request

6 SR Open Date:- Date on which the Service Request was logged on system

7 SR Close Date:- Date on which the Service Request was closed on system

8 Product :- The name of the product for which the Service Request was raised

9 Geography:- The continent in which the client belongs

10 Country:- The country to which the client belongs

Clean the data and make it ready for analysis of Resolution time i.e., Time, in number of days ,taken to resolve the Service Request (to be calculated as difference between SR Close Date and SR Open Date.)

Let's create a Project Plan on the lines of DCOVA&I

1. Define the problem :- Clean the data and create a Project Datamart to do Analytics for Resolution Time i.e., Time, in number of days ,taken to resolve the Service Request (to be calculated as difference between SR Close Date and SR Open Date.)

1.1. Create the y variable

2. Colect the relevant data: - The data for the project comes from one file "CaseStudy1.csv"

3. Organise the data: - Manipulate data , create derived variables thru calculation, Understand missing values

4. Visualise the data :-

4.1. Univariate analysis of y

4.2. Multi-variate analysis :- Correlation

5. Create the final Project Data mart

5.1. Drop variables :-

5.1.1. Variables used to create y

5.1.2. Variables which are non-numeric

5.2. Numeric values :- ServiceRequestStatus, TypeOfEnagagement, Incident/ Problem, SR Priority, Product, Geography, Country

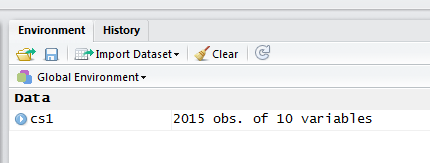
Lets look at how we can solve the CaseStudy1 using R

1. Set the path to the directory which contains the CaseStudy1 data :-

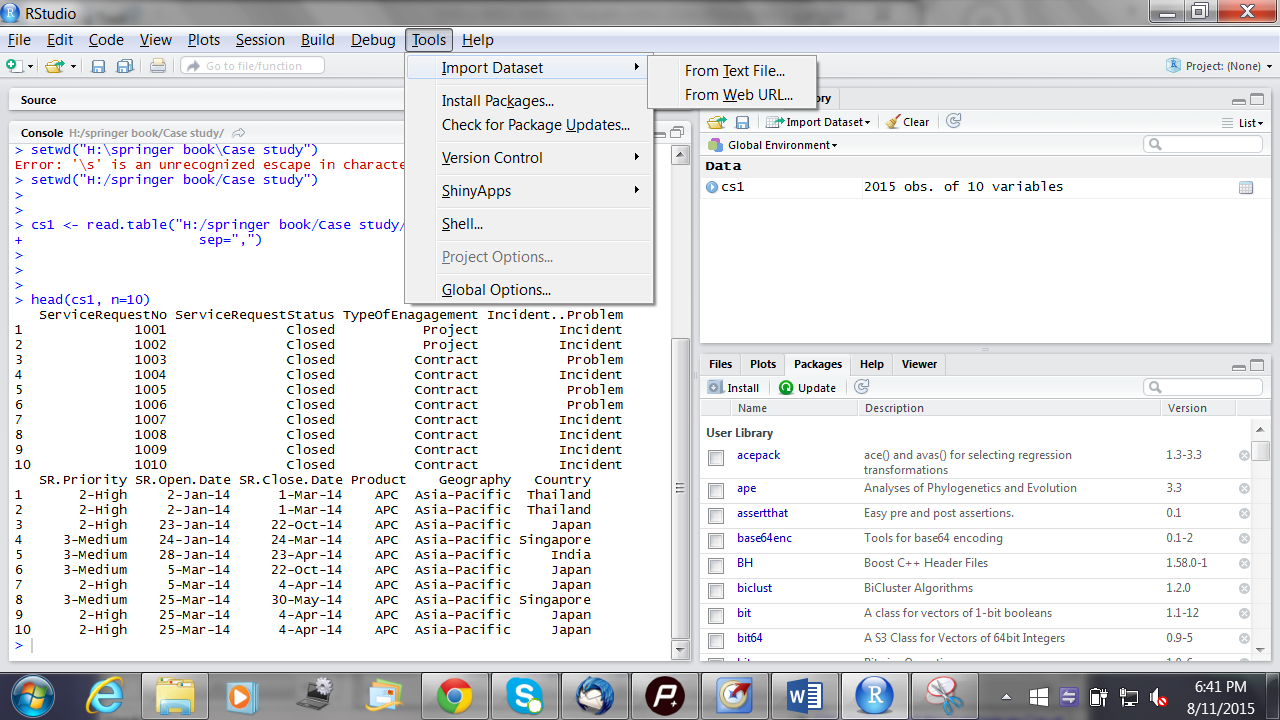
setwd("H:/springer book/Case study")

2. Import the data

cs1 <- read.table("H:/springer book/Case study/CaseStudy1.csv", header=TRUE, sep=",")

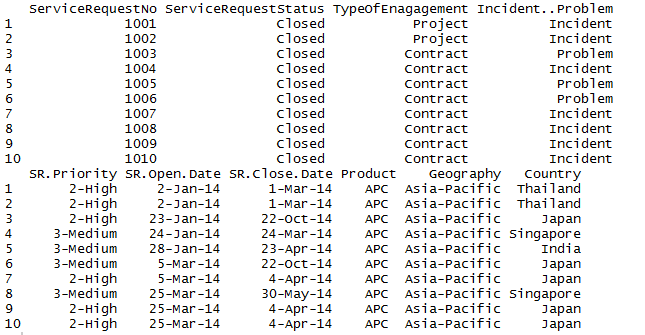


Tip :- An alternate way to import in the data is to use the button Tools > Import Data. This will open an Import Wizard which you can use to bring the data into R .

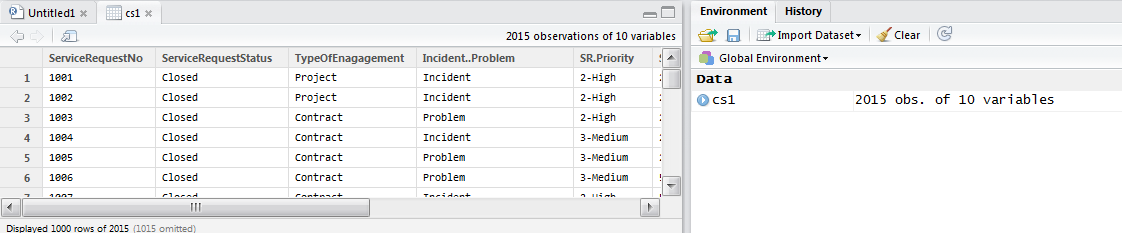


3. View first 10 lines of data to see how the dataset looks

head(cs1, n=10)



Tip :- An alternate way to view the data is to double click on the dataset created in the Global Environment window . You can see file in the Script Editor pane on the top left side !!



4. Understand the Types of Variables in the dataset

4.1 str(cs1)

'data.frame': 2015 obs. of 10 variables:

$ ServiceRequestNo : int 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 ...

$ ServiceRequestStatus: Factor w/ 6 levels "Canceled","Closed",..: 2 2 2 2 2 2 2 2 2 2 ...

$ TypeOfEnagagement : Factor w/ 5 levels "Billable","Contract",..: 4 4 2 2 2 2 2 2 2 2 ...

$ Incident..Problem : Factor w/ 3 levels "Incident","Problem",..: 1 1 2 1 2 2 1 1 1 1 ...

$ SR.Priority : Factor w/ 4 levels "1-Critical","2-High",..: 2 2 2 3 3 3 2 3 2 2 ...

$ SR.Open.Date : Factor w/ 368 levels "1-Apr-14","1-Apr-15",..: 142 142 190 204 253 315 315 219 219 219 ...

$ SR.Close.Date : Factor w/ 304 levels "","1-Apr-15",..: 7 7 143 161 145 143 240 232 240 240 ...

$ Product : Factor w/ 12 levels "AAM ","Adva",..: 3 3 3 3 3 3 3 3 3 3 ...

$ Geography : Factor w/ 1 level "Asia-Pacific": 1 1 1 1 1 1 1 1 1 1 ...

$ Country : Factor w/ 18 levels "Australia","Bangladesh",..: 16 16 7 14 5 7 7 14 7 7 ...

Note :- A dataset is called a data frame in R

Tip :- Do look up these websites for R coding are

1. [http://www.cookbook-r.com/Data\_input\_and\_output/Loading\_data\_from\_a\_file](http://www.cookbook-r.com/Data_input_and_output/Loading_data_from_a_file/); 2. [http://www.statmethods.net/input/importingdata.html](http://www.statmethods.net/input/importingdata.html%20/)

4.2 Dimensions - Observations and Variables

dim(cs1)

[1] 2015 10

4.3 Names of Variables

names(cs1)

[1] "ServiceRequestNo" "ServiceRequestStatus" "TypeOfEnagagement"

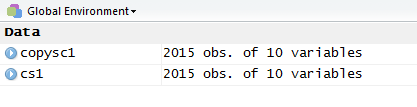
[4] "Incident..Problem" "SR.Priority" "SR.Open.Date"

[7] "SR.Close.Date" "Product" "Geography"

[10] "Country"

5. Save a copy of original data (cs1)

copysc1<-cs1

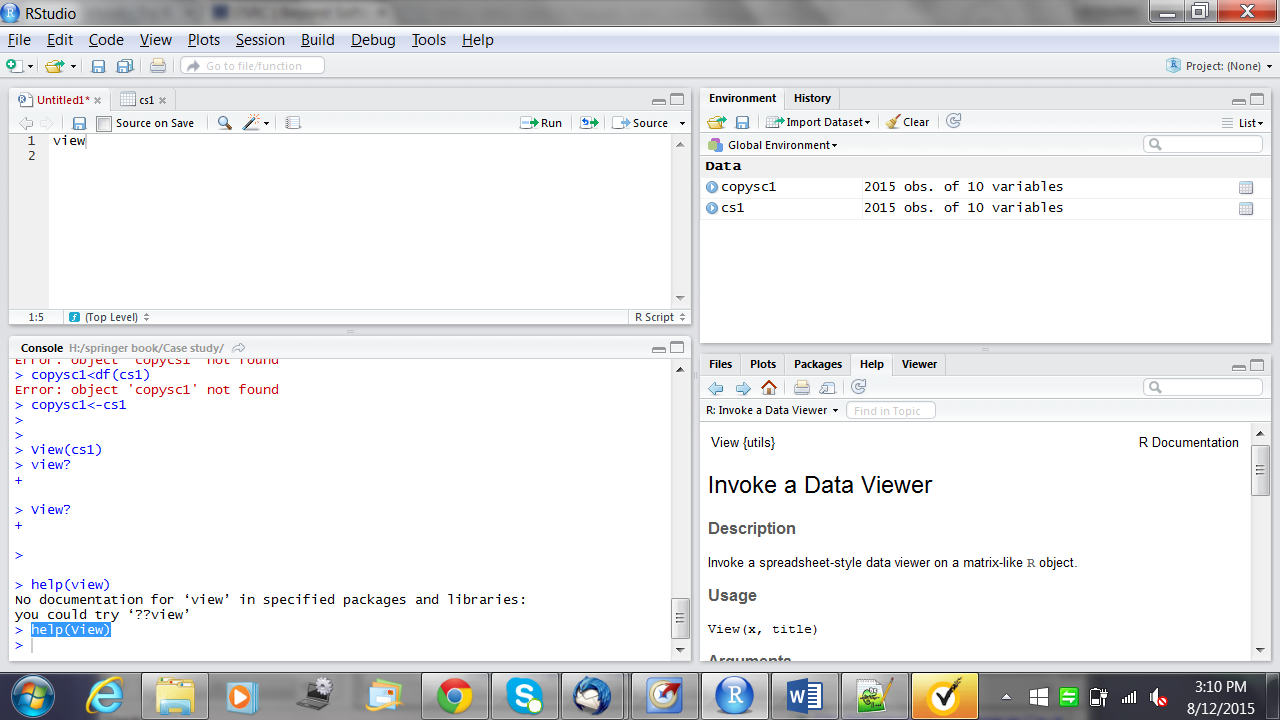


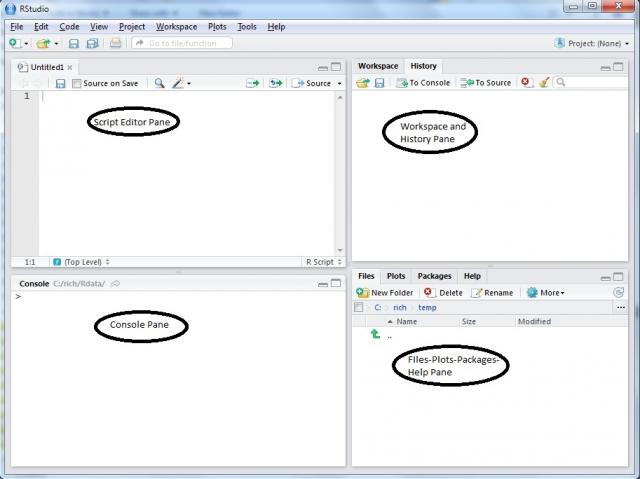
Note :- You can use = or <- interchangably in the R code.

Tip :- Auto completion of codes in R can be done by using the Tab key if you write the codes in the Script Editor Pane and NOT in the Console area.

Tip :- For help with codes in the Console area, type help(name of fucntion) and enter. It will open up help in the bottom left square.

Tip:- Click on the Help tabs to type in your query and get help





6. Create y variable of Resolution Time = difference between SR Close Date and SR Open Date.

> cs1$ResolutionTime<-cs1$SR.Close.Date-cs1$SR.Open.Date

Warning message:

In Ops.factor(cs1$SR.Close.Date, cs1$SR.Open.Date) :

‘-’ not meaningful for factors

FAQ:- What does this error mean ? It means that the format of the variable is a Factor . What is the solution ? Let us change the format of the SR.Close.Date and SR.Open.Date to numbers.OR lets convert these 2 variables into numbers and re-import the data .

> cs1$SR.Close.Date<-as.Date(cs1$SR.Close.Date ,"%m/%d/%Y" )

> cs1$SR.Open.Date<-as.Date(cs1$SR.Open.Date ,"%m/%d/%Y" )

> str(cs1)

'data.frame': 2015 obs. of 11 variables:

$ ServiceRequestNo : int 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 ...

$ ServiceRequestStatus: Factor w/ 6 levels "Canceled","Closed",..: 2 2 2 2 2 2 2 2 2 2 ...

$ TypeOfEnagagement : Factor w/ 5 levels "Billable","Contract",..: 4 4 2 2 2 2 2 2 2 2 ...

$ Incident..Problem : Factor w/ 3 levels "Incident","Problem",..: 1 1 2 1 2 2 1 1 1 1 ...

$ SR.Priority : Factor w/ 4 levels "1-Critical","2-High",..: 2 2 2 3 3 3 2 3 2 2 ...

$ SR.Open.Date : Date, format: NA NA ...

$ SR.Close.Date : Date, format: NA NA ...

$ Product : Factor w/ 12 levels "AAM ","Adva",..: 3 3 3 3 3 3 3 3 3 3 ...

$ Geography : Factor w/ 1 level "Asia-Pacific": 1 1 1 1 1 1 1 1 1 1 ...

$ Country : Factor w/ 18 levels "Australia","Bangladesh",..: 16 16 7 14 5 7 7 14 7 7 ...

$ ResolutionTime : logi NA NA NA NA NA NA ...

Note :- The data types in R can be of the following

1. Basic Data Types

1.1 Numeric - Decimal numbers

1.2 Integer - Number without a fraction / decimal.

1.3 Complex - A complex number is a number that can be expressed in the form a + bi, where a and b are real numbers and i is the imaginary unit, that satisfies the equation i2 = −1.

1.4 Logical - True / False

1.5 Character - String Values

2. Vector - A vector is a sequence of data elements of the same basic type, generally numbers. Members in a vector are officially called components.

3. Matrix - A matrix is a collection of data elements arranged in a two-dimensional rectangular layout.

4. List - A list is a generic vector containing numeric, non numeric and other objects.

5. Data Frame - A data frame is a table, or two-dimensional array-like structure, in which each column contains measurements on one variable, and each row contains one case.

Re-import the data :- I formatted the two columns of SR.Open.Date and SR.Close.Date as numbers.

> setwd("H:/springer book/Case study")

> cs1 <- read.table("H:/springer book/Case study/CaseStudy1.csv", header=TRUE, sep=",",stringsAsFactors=FALSE)

> str(cs1)

'data.frame': 2015 obs. of 10 variables:

$ ServiceRequestNo : int 2090 2863 1517 2864 2865 2866 2022 2061 2954 2996 ...

$ ServiceRequestStatus: chr "Closed" "Closed" "Closed" "Closed" ...

$ TypeOfEnagagement : chr "Project" "Contract" "Project" "No Charge Support" ...

$ Incident..Problem : chr "Incident" "Problem" "Problem" "Request for Fulfillment" ...

$ SR.Priority : chr "3-Medium" "3-Medium" "3-Medium" "3-Medium" ...

$ SR.Open.Date : num 41647 41655 41649 41666 41666 ...

$ SR.Close.Date : num 41648 41659 41663 41667 41667 ...

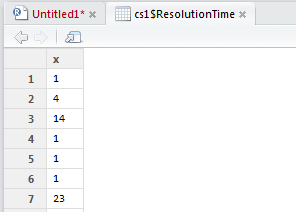
$ Product : chr "Oper" "Busi" "AAM " "Busi" ...

$ Geography : chr "Asia-Pacific" "Asia-Pacific" "Asia-Pacific" "Asia-Pacific" ...

$ Country : chr "Australia" "India" "New Zealand" "India" ...

> cs1$ResolutionTime<-cs1$SR.Close.Date-cs1$SR.Open.Date

> View(cs1$ResolutionTime)



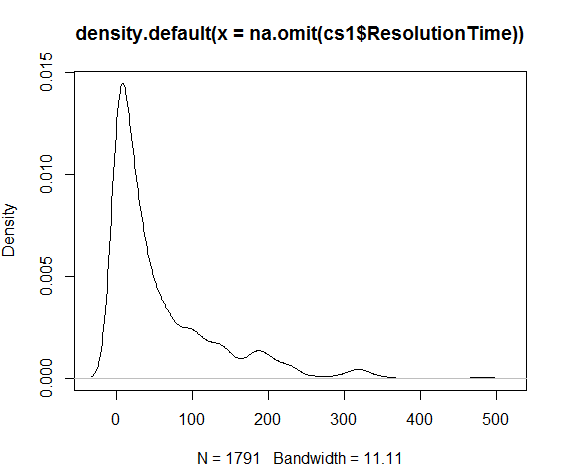
The difference in days has been computed in Resolution time

7. Understand the distribution of the y variable

> d <- density(na.omit(cs1$ResolutionTime))

> plot(d)

FAQ :- na.omit helps to omit missing values



8. Univariate Analysis of Y variable . What does the descriptive stats say ?

> install.packages('pastecs')

Installing package into ‘C:/Users/Subhashini/Documents/R/win-library/3.1’

(as ‘lib’ is unspecified)

trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.1/pastecs\_1.3-18.zip'

Content type 'application/zip' length 1636168 bytes (1.6 MB)

opened URL

downloaded 1.6 MB

package ‘pastecs’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Subhashini\AppData\Local\Temp\RtmpuqEEvG\downloaded\_packages

> library(pastecs)

Loading required package: boot

> stat.desc(cs1$ResolutionTime)

nbr.val nbr.null nbr.na min max range

1.791000e+03 1.720000e+02 2.240000e+02 0.000000e+00 4.840000e+02 4.840000e+02

sum median mean SE.mean CI.mean.0.95 var

1.010490e+05 2.800000e+01 5.642044e+01 1.679654e+00 3.294289e+00 5.052837e+03

std.dev coef.var

7.108331e+01 1.259886e+00

Note :- to change the output to an easily understood format change the format of display .

To see all the statistics, set the options as seen below .

> options(scipen=100)

> options(digits=2)

Now run the descriptive stats.

> stat.desc(cs1$ResolutionTime)

nbr.val nbr.null nbr.na min max range

1791.0 172.0 224.0 0.0 484.0 484.0

sum median mean SE.mean CI.mean.0.95 var

101049.0 28.0 56.4 1.7 3.3 5052.8

std.dev coef.var

71.1 1.3

If we only want the descriptive statistics, such as the min, max and std.dev? We can add an option as shown below.

> stat.desc(cs1$ResolutionTime, basic=F)

median mean SE.mean CI.mean.0.95 var std.dev

28.0 56.4 1.7 3.3 5052.8 71.1

coef.var

1.3

If we want only the basic statistics such as the number of observations and number of missing values.

> stat.desc(cs1$ResolutionTime, desc=F)

nbr.val nbr.null nbr.na min max range sum

1791 172 224 0 484 484 101049

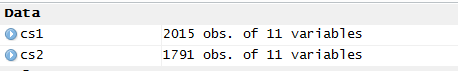
9. Drop the observations with Y missing .

Nbr.null = 172

Nbr.na = 224 (which contains the Nbr.null = 172)

Therefore, total number of observations to be dropped is Nbr.na = 224

> cs2<-cs1[!is.na(cs1$ResolutionTime),]



Tip :- Why remove observations with NA / data missing ?

Since Y is the independent variable and we have enough data (>500 obs) we choose to use the observations which have Y present.

10. Do some visualisations to understand the data

Install a new set of packages to help create the tables and charts !!

> install.packages('gmodels')

Installing package into ‘C:/Users/Subhashini/Documents/R/win-library/3.1’

(as ‘lib’ is unspecified)

trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.1/gmodels\_2.16.2.zip'

Content type 'application/zip' length 73922 bytes (72 KB)

opened URL

downloaded 72 KB

package ‘gmodels’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Subhashini\AppData\Local\Temp\RtmpuqEEvG\downloaded\_packages

> library(gmodels)

> CrossTable(cs2$ServiceRequestStatus)

Cell Contents

|-------------------------|

| N |

| N / Table Total |

|-------------------------|

Total Observations in Table: 1791

| Canceled | Closed |

|-----------|-----------|

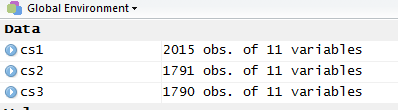
| 1 | 1790 |

| 0.001 | 0.999 |

|-----------|-----------|

We need to drop the case which has status 'Cancelled' . Why ? Because there is only one such case and obviously an outlier . Also, Resolution Time is measured only for cases that are closed !!

> cs3 <- cs2[ which(cs2$ServiceRequestStatus=='Closed'), ]



Frequency table for TypeOfEnagagement

> mytable <- table(cs3$TypeOfEnagagement)

>

> mytable

Billable Contract No Charge Support Project

43 489 351 901

Unverified

6

Install libraries:- library(Lahman)

library(plyr)

> Table1 <- ddply(cs3, "Incident..Problem", summarise, total = mean(ResolutionTime))

> Table1

Incident..Problem total

1 Incident 48

2 Problem 79

3 Request for Fulfillment 49

> Table2 <- ddply(cs3, "Product", summarise, total = mean(ResolutionTime))

> Table2

Product total

1 AAM 54

2 Adva 129

3 APC 77

4 Blen 21

5 Busi 50

6 DynA 24

7 Oper 69

8 PHD 82

9 Proc 20

10 Prod 21

11 Unsp 75

12 USD 46

Sort the table Table2 to understand contribution of each Product

Note :- Default sorting order is ascending . To sort in descending order , add a - (minus) sign in front of the variable !!

#sort by var1 (ascending) and var2 (descending)  
newdata <- olddata[order(var1, -var2),]

> names(Table2)

[1] "Product" "total"

> Table2a <- Table2[order(Table2$total),]

> Table2a

Product total

9 Proc 20

10 Prod 21

4 Blen 21

6 DynA 24

12 USD 46

5 Busi 50

1 AAM 54

7 Oper 69

11 Unsp 75

3 APC 77

8 PHD 82

2 Adva 129

> Table3 <- ddply(cs3, "Geography", summarise, total = mean(ResolutionTime))

> Table3

Geography total

1 Asia-Pacific 56

Tip :- All the the cases considered are from the same geography . This field is thus redundant since there is no variation.

Create table to understand the Country variable

> Table4 <- ddply(cs3, "Country", summarise, total = mean(ResolutionTime))

> Table4

Country total

1 Australia 84.2

2 Bangladesh 125.7

3 Brunei Darussalam 8.0

4 China 54.2

5 India 47.7

6 Indonesia 78.0

7 Japan 55.4

8 Korea South 39.3

9 Malaysia 55.2

10 New Zealand 42.4

11 Oman 3.7

12 Papua New Guinea 74.1

13 Philippines 52.9

14 Singapore 64.1

15 Taiwan 39.8

16 Thailand 43.5

17 Unspecified 68.9

18 Vietnam 23.2

Lets sort this table in Desceending order of total .

> Table4a <- Table4[order(-Table2$total),]

> Table4a

Country total

2 Bangladesh 125.7

8 Korea South 39.3

3 Brunei Darussalam 8.0

11 Oman 3.7

7 Japan 55.4

1 Australia 84.2

5 India 47.7

12 Papua New Guinea 74.1

6 Indonesia 78.0

4 China 54.2

10 New Zealand 42.4

9 Malaysia 55.2

We can see a lot of variation in the Average time taken to resolve requests across different countries

Table to explore Average Resolution Time across Service Priority

> Table5 <- ddply(cs3, "SR.Priority", summarise, total = mean(ResolutionTime))

> Table5

SR.Priority total

1 1-Critical 53

2 2-High 49

3 3-Medium 60

4 4-Low 64

This table is interesting !! The Average Resolution Time for High Priority is less than for Critical

11. To create the final Project Datamart , the following data manipulations should be done

11.1 Remove Geography as a field since all cases are for Asia-Pacific

11.2 Remove Service Request Status as all status is closed

11.3 Remove SR Open Date as it is used to calculate the Y variables

11.4 Remove SR Close Date as it is used to calculate the Y variable

11.5 Convert TypeOfEnagagement, Incident..Problem, SR.Priority, Product, Country into numeric fields (dummy and derived variables).

> cs3$ServiceRequestStatus<-NULL



> cs3$SR.Open.Date<-NULL



> cs3$SR.Close.Date<-NULL



> cs3$Geography<-NULL



Create Dummy variables in the cs3 data frame

> for(level in unique(cs3$TypeOfEnagagement)){cs3[paste("dummy", level, sep = "\_")] <- ifelse(cs3$TypeOfEnagagement == level, 1, 0)}

> View(cs3)



> for(level in unique(cs3$Incident..Problem)){cs3[paste("dummy", level, sep = "\_")] <- ifelse(cs3$Incident..Problem == level, 1, 0)}

> for(level in unique(cs3$Product)){cs3[paste("dummy", level, sep = "\_")] <- ifelse(cs3$Product == level, 1, 0)}

> for(level in unique(cs3$Country)){cs3[paste("dummy", level, sep = "\_")] <- ifelse(cs3$Country == level, 1, 0)}

For SR.Priority , we need to extract the first character of the string

> cs3$Priority<- substring(cs3$SR.Priority, 1, 1)

( syntax = substr(x, start, stop))

Lets now view the Variables in the cs3 dataframe - which will serve as the Project Datamart going forward

> names (cs3)

[1] "ServiceRequestNo" "TypeOfEnagagement"

[3] "Incident..Problem" "SR.Priority"

[5] "Product" "Country"

[7] "ResolutionTime" "dummy\_Project"

[9] "dummy\_Contract" "dummy\_No Charge Support"

[11] "dummy\_Billable" "dummy\_Unverified"

[13] "dummy\_Incident" "dummy\_Problem"

[15] "dummy\_Request for Fulfillment" "dummy\_Oper"

[17] "dummy\_Busi" "dummy\_AAM "

[19] "dummy\_Blen" "dummy\_Unsp"

[21] "dummy\_PHD " "dummy\_APC "

[23] "dummy\_USD " "dummy\_Adva"

[25] "dummy\_Proc" "dummy\_DynA"

[27] "dummy\_Prod" "dummy\_Australia"

[29] "dummy\_India" "dummy\_New Zealand"

[31] "dummy\_Singapore" "dummy\_Japan"

[33] "dummy\_Korea South" "dummy\_Thailand"

[35] "dummy\_Taiwan" "dummy\_Malaysia"

[37] "dummy\_Indonesia" "dummy\_Vietnam"

[39] "dummy\_Philippines" "dummy\_Bangladesh"

[41] "dummy\_Unspecified" "dummy\_China"

[43] "dummy\_Brunei Darussalam" "dummy\_Papua New Guinea"

[45] "dummy\_Oman" "Priority"

Let us now drop the original variables on which the dummy and derived variables have been created

> cs4<-subset(cs3, , -c(2:6))



Now save the Project Datamart (cs4 file) and other files for later use

> save(cs3,file="H:/springer book/Case study/cs3.Rda")

> save(cs4,file="H:/springer book/Case study/cs4.Rda")

Note :- You can load R datafiles using the - load("path/data.Rda") - command