Practical 2 - Data Management

Theory

Data Management tasks are

- **Creating Subsets**
- **Deriving**
- **Sorting**
- **Merging**
- **Aggregating**

1. Read data from CSV file in R

```
#import basic_salary data
salary<-read.csv(file.choose(),header=T)
summary(salary)
```

```
 First_Name   Last_Name   Grade      Location       Function       ba            ms
 Kavita : 2   Joshi  : 2   GR1 :23   DELHI :17   FINANCE  :13   Min.   :10940   Min.   : 2700
 Mahesh : 2   Shah   : 2   GR2 :17   MUMBAI:21   SALES    :15   1st Qu.:13785   1st Qu.:10450
 Nishi  : 2   Singh  : 2   NA's: 1   NA's  : 3   TECHNICAL:11   Median :16230   Median :12420
 Priya  : 2   Arora  : 1                          NA's     : 2   Mean   :17210   Mean   :11939
 Ajit   : 1   Bhide  : 1                                         3rd Qu.:19305   3rd Qu.:14200
 Ameet  : 1   Bhutala: 1                                         Max.   :29080   Max.   :16970
 (Other):31   (Other):32                                         NA's   :2       NA's   :4
```

2. Find missing values

```
#Number of missing observations
nmiss<-sum(is.na(salary$ba))
nmiss
[1] 12
```

3. Find mean of the column “ba”

```
mean(salary$ba )
[1] NA
```

#NA output due to missing observations

4. Remove missing observations and calculates mean

```
#removes missing observations and calculates mean
mean(salary$ba,na.rm=TRUE)

[1] 17209.74
```

5. Create subset using selected rows

```
#Display rows from 5th to 10th
salary[c(5:10), ]
```

```
   First_Name Last_Name Grade Location  Function    ba    ms
5       Sneha     Joshi   GR1    DELHI   FINANCE 20660 15660
6      Mahesh      Rane   GR1    DELHI TECHNICAL 23160 14200
7         Ram    Kanade   GR1    DELHI TECHNICAL 20160 15850
8       Nishi    Honrao   GR1    DELHI TECHNICAL 20460 15880
9       Nishi  Kulkarni   GR1     <NA>     SALES 22620 16150
10     Hameed     Singh   GR1    DELHI     SALES 23720 15120
```

6. Display only selected rows

```
salary[c(1,3,5,8), ]
```

```
  First_Name Last_Name Grade Location  Function    ba    ms
1     Mahesh     Joshi   GR1    DELHI     SALES 17990 16070
3       Neha       Rao   GR1    DELHI   FINANCE 19235 15200
5      Sneha     Joshi   GR1    DELHI   FINANCE 20660 15660
8      Nishi    Honrao   GR1    DELHI TECHNICAL 20460 15880
```

7. Create subset using selected columns

```
salary1<-salary[, c(1,2)] #Object salary1 has columns 1 and 2
head(salary1)
```

```
  First_Name Last_Name
1     Mahesh     Joshi
2     Rajesh     Kolte
3       Neha       Rao
4      Priya      Jain
5      Sneha     Joshi
6     Mahesh      Rane
```

8. Create subset using selected rows for selected columns

```
#Object salary2 has rows 1,5,8,4 and columns 1 and 2
salary2<-salary[c(1,5,8,4), c(1,2)]
salary2
```

```
  First_Name Last_Name
1     Mahesh     Joshi
5      Sneha     Joshi
8      Nishi    Honrao
4      Priya      Jain
```

9. Create subsets using Subset function

- **Condition on observations**

```
#All details of employees of DELHI with ba more than  20000
salary3<-subset(salary, Location=="DELHI" & ba > 20000)
 head(salary3)
```

```
   First_Name Last_Name Grade Location  Function    ba    ms
4       Priya      Jain   GR1    DELHI     SALES 23280 13490
5       Sneha     Joshi   GR1    DELHI   FINANCE 20660 15660
6      Mahesh      Rane   GR1    DELHI TECHNICAL 23160 14200
7         Ram    Kanade   GR1    DELHI TECHNICAL 20160 15850
8       Nishi    Honrao   GR1    DELHI TECHNICAL 20460 15880
10     Hameed     Singh   GR1    DELHI     SALES 23720 15120
```

- **Condition on variable names**

```
#Only First name and Last name of previous data
salary4<-subset(salary3,select=c(First_Name,Last_Name))
head(salary4)
```

```
   First_Name Last_Name
4       Priya      Jain
5       Sneha     Joshi
6      Mahesh      Rane
7         Ram    Kanade
8       Nishi    Honrao
10     Hameed     Singh
```

- **Condition on observations and variable names**

```
#Select details of       specific set of employees
salary5<-subset(salary,Grade=="GR1" & ba>15000,select=  c(First_Name,Grade,
Location))
head(salary5)
```

```
  First_Name Grade Location
1     Mahesh   GR1    DELHI
2     Rajesh   GR1    DELHI
3       Neha   GR1    DELHI
4      Priya   GR1    DELHI
5      Sneha   GR1    DELHI
6     Mahesh   GR1    DELHI
```

```
#Details of employees not having grade 1 and not from  Mumbai
salary6<-subset(salary,!(Grade=="GR1") & !(Location=="MUMBAI"))
```

salary6

```
  First_Name Last_Name Grade Location  Function    ba    ms
1     Mahesh     Joshi   GR1    DELHI     SALES 17990 16070
2    Rajesh      Kolte   GR1    DELHI   FINANCE 19250 14960
3       Neha       Rao   GR1    DELHI   FINANCE 19235 15200
4      Priya      Jain   GR1    DELHI     SALES 23280 13490
5      Sneha     Joshi   GR1    DELHI   FINANCE 20660 15660
6     Mahesh      Rane   GR1    DELHI TECHNICAL 23160 14200
```

10. Perform Data Sorting functions

# Import and attach basic_salary data
salary_data<-read.csv(file.choose(),header=T)
attach(salary_data)

Sorting data is one of the common activity in preparing data for analysis Sorting is storage of data in sorted order, it can be in ascending or descending order.

attach() attaches the database to the R search path, so the variables in the database can be accessed by simply giving their names

# Sort salary_data by ba in ascending order
ba_sorted_1<-salary_data[order(ba,]
ba_sorted_1

```
   First_Name Last_Name Grade Location  Function    ba    ms
37      Archa  Narvekar   GR2   MUMBAI TECHNICAL 10940 11160
32       Anup      Save   GR2   MUMBAI     SALES 11960  7880
33     Yogesh    Lonkar   GR2   MUMBAI TECHNICAL 12390  6630
38      Shiva    Jathar   GR2   MUMBAI   FINANCE 12860 10940
41      Ketan   Kharkar   GR2   MUMBAI     SALES 13140  9800
34      Sagar    Chavan   GR2   MUMBAI   FINANCE 13390  6700
35        Dev     Patil   GR2   MUMBAI     SALES 13500 10760
39        Anu   Bhutala   GR2   MUMBAI   FINANCE 13650 10580
30       Amit     Mehta   GR2    DELHI TECHNICAL 13660  6840
29     Gaurav     Singh   GR2    DELHI     SALES 13760 13220
26     Naresh     Sinha   GR2    DELHI TECHNICAL 13810 11540
40       Nita   Punjabi   GR2   MUMBAI     SALES 14050    NA
13     Anjali     Sonar   GR1   MUMBAI      <NA> 14410 10450
31      Ameet    Mishra   GR2    DELHI   FINANCE 14780  9300
25      Priya    Mittal   GR2    DELHI TECHNICAL 15000 10680
15      Rahul    Potdar   GR1   MUMBAI     SALES 15125    NA
14      Bipin     Bhide   GR1   MUMBAI   FINANCE 15230 11010
17    Mangesh       Oak   GR1   MUMBAI     SALES 15800 12420
27     Jivesh      Shah   GR2     <NA>   FINANCE 16000 13730
28      Jigar      Shah   GR2    DELHI   FINANCE 16230    NA
18      Anand     Soman   GR1     <NA>   FINANCE 16540 12780
19     Malhar    Jadhav   GR1   MUMBAI TECHNICAL 17240 13220
```

```
24     Kavita        NS   GR1   MUMBAI      <NA> 17520    NA
22       Jina     Arora   GR1   MUMBAI     SALES 17830 13090
1      Mahesh     Joshi   GR1    DELHI     SALES 17990 16070
20     Suresh        VS   GR1   MUMBAI TECHNICAL 18310 13220
23     Kavita       NA    GR1   MUMBAI     SALES 19000  2700
3        Neha       Rao   GR1    DELHI   FINANCE 19235 15200
2      Rajesh     Kolte   GR1    DELHI   FINANCE 19250 14960
21      Rajni      Gudi   GR1   MUMBAI   FINANCE 19360 13050
7         Ram    Kanade   GR1    DELHI TECHNICAL 20160 15850
8       Nishi    Honrao   GR1    DELHI TECHNICAL 20460 15880
5       Sneha     Joshi   GR1    DELHI   FINANCE 20660 15660
9       Nishi  Kulkarni   GR1     <NA>     SALES 22620 16150
6      Mahesh      Rane   GR1    DELHI TECHNICAL 23160 14200
4       Priya      Jain   GR1    DELHI     SALES 23280 13490
10     Hameed     Singh   GR1    DELHI     SALES 23720 15120
11        Raj    Mohite   GR1    DELHI   FINANCE 26080 16970
12     Yogita      Raje   GR1    DELHI     SALES 29080  8795
16     Ganesh      Sane  <NA>   MUMBAI     SALES    NA 12120
36       Ajit    Shinde   GR2   MUMBAI TECHNICAL    NA  9580
```

```
# Sort salary_data by ba in Descending order
ba_sorted_1<-salary_data[order(-ba,]
ba_sorted_1
```

```
   First_Name Last_Name Grade Location  Function    ba    ms
12     Yogita      Raje   GR1    DELHI     SALES 29080  8795
11        Raj    Mohite   GR1    DELHI   FINANCE 26080 16970
10     Hameed     Singh   GR1    DELHI     SALES 23720 15120
4       Priya      Jain   GR1    DELHI     SALES 23280 13490
6      Mahesh      Rane   GR1    DELHI TECHNICAL 23160 14200
9       Nishi  Kulkarni   GR1     <NA>     SALES 22620 16150
5       Sneha     Joshi   GR1    DELHI   FINANCE 20660 15660
8       Nishi    Honrao   GR1    DELHI TECHNICAL 20460 15880
7         Ram    Kanade   GR1    DELHI TECHNICAL 20160 15850
21      Rajni      Gudi   GR1   MUMBAI   FINANCE 19360 13050
2      Rajesh     Kolte   GR1    DELHI   FINANCE 19250 14960
3        Neha       Rao   GR1    DELHI   FINANCE 19235 15200
23     Kavita       NA    GR1   MUMBAI     SALES 19000  2700
20     Suresh        VS   GR1   MUMBAI TECHNICAL 18310 13220
1      Mahesh     Joshi   GR1    DELHI     SALES 17990 16070
22       Jina     Arora   GR1   MUMBAI     SALES 17830 13090
24     Kavita        NS   GR1   MUMBAI      <NA> 17520    NA
19     Malhar    Jadhav   GR1   MUMBAI TECHNICAL 17240 13220
18      Anand     Soman   GR1     <NA>   FINANCE 16540 12780
28      Jigar      Shah   GR2    DELHI   FINANCE 16230    NA
27     Jivesh      Shah   GR2     <NA>   FINANCE 16000 13730
17    Mangesh       Oak   GR1   MUMBAI     SALES 15800 12420
14      Bipin     Bhide   GR1   MUMBAI   FINANCE 15230 11010
15      Rahul    Potdar   GR1   MUMBAI     SALES 15125    NA
25      Priya    Mittal   GR2    DELHI TECHNICAL 15000 10680
```

```
31      Ameet    Mishra   GR2    DELHI   FINANCE 14780  9300
13     Anjali     Sonar   GR1   MUMBAI      <NA> 14410 10450
40       Nita   Punjabi   GR2   MUMBAI     SALES 14050    NA
26     Naresh     Sinha   GR2    DELHI TECHNICAL 13810 11540
29     Gaurav     Singh   GR2    DELHI     SALES 13760 13220
30       Amit     Mehta   GR2    DELHI TECHNICAL 13660  6840
39        Anu   Bhutala   GR2   MUMBAI   FINANCE 13650 10580
35        Dev     Patil   GR2   MUMBAI     SALES 13500 10760
34      Sagar    Chavan   GR2   MUMBAI   FINANCE 13390  6700
41      Ketan   Kharkar   GR2   MUMBAI     SALES 13140  9800
38      Shiva    Jathar   GR2   MUMBAI   FINANCE 12860 10940
33     Yogesh    Lonkar   GR2   MUMBAI TECHNICAL 12390  6630
32       Anup      Save   GR2   MUMBAI     SALES 11960  7880
37      Archa  Narvekar   GR2   MUMBAI TECHNICAL 10940 11160
16     Ganesh      Sane  <NA>   MUMBAI     SALES    NA 12120
36       Ajit    Shinde   GR2   MUMBAI TECHNICAL    NA  9580
```

Sort data by column with characters / factors

```
# Sort salary_data by Grade
gr_sorted<-salary_data[order(Grade),]
head(gr_sorted)
```

```
  First_Name Last_Name Grade Location  Function    ba    ms
1     Mahesh     Joshi   GR1    DELHI     SALES 17990 16070
2    Rajesh     Kolte   GR1    DELHI   FINANCE 19250 14960
3       Neha       Rao   GR1    DELHI   FINANCE 19235 15200
4      Priya      Jain   GR1    DELHI     SALES 23280 13490
5      Sneha     Joshi   GR1    DELHI   FINANCE 20660 15660
6     Mahesh      Rane   GR1    DELHI TECHNICAL 23160 14200
```

Sort data by column with characters / factors in Descending order

```
# Sort salary_data by Grade in descending order
gr_sorted<-salary_data[order(Grade,decreasing=TRUE),]
head(gr_sorted)
```

```
   First_Name Last_Name Grade Location  Function    ba    ms
25      Priya    Mittal   GR2    DELHI TECHNICAL 15000 10680
26     Naresh     Sinha   GR2    DELHI TECHNICAL 13810 11540
27     Jivesh      Shah   GR2     <NA>   FINANCE 16000 13730
28      Jigar      Shah   GR2    DELHI   FINANCE 16230    NA
29     Gaurav     Singh   GR2    DELHI     SALES 13760 13220
30       Amit     Mehta   GR2    DELHI TECHNICAL 13660  6840
```

Sort data by giving multiple columns; one column with characters / factors and one with numerals
# Sort salary_data by Grade and ba
grba_sorted<-salary_data[order(Grade,ba),]
head(grba_sorted)

```
   First_Name Last_Name Grade Location  Function    ba    ms
13     Anjali     Sonar   GR1   MUMBAI      <NA> 14410 10450
15      Rahul    Potdar   GR1   MUMBAI     SALES 15125    NA
14      Bipin     Bhide   GR1   MUMBAI   FINANCE 15230 11010
17    Mangesh       Oak   GR1   MUMBAI     SALES 15800 12420
18      Anand     Soman   GR1     <NA>   FINANCE 16540 12780
19     Malhar    Jadhav   GR1   MUMBAI TECHNICAL 17240 13220
```

11. Perform merging / joining operations using merge function
#Import following 2 data sets
sal_data
bonus_data

outerjoin
# Outer Join includes all employee ID’s from both data sets
outerjoin<- merge(sal_data,bonus_data,by=c("Employee_ID"), all=TRUE)

```
   Employee_ID First_Name Last_Name Basic_Salary Bonus
1       E-1001     Mahesh     Joshi        16070 16070
2       E-1002     Rajesh     Kolte        14960    NA
3       E-1004      Priya      Jain        13490 13490
4       E-1005      Sneha     Joshi        15660    NA
5       E-1007        Ram    Kanade        15850    NA
6       E-1008      Nishi    Honrao        15880 15880
7       E-1009     Hameed     Singh        15120    NA
8       E-1003       <NA>      <NA>           NA 15200
9       E-1006       <NA>      <NA>           NA 14200
10      E-1010       <NA>      <NA>           NA 15120
```

innerjoin
# Inner Join includes employee ID only if present in both data sets
innerjoin<-merge(sal_data,bonus_data,by="Employee_ID")

```
  Employee_ID First_Name Last_Name Basic_Salary Bonus
1      E-1001     Mahesh     Joshi        16070 16070
2      E-1004      Priya      Jain        13490 13490
3      E-1008      Nishi    Honrao        15880 15880
```

leftjoin

```
#Left Join includes all employee ID’s from first data set
leftjoin<-merge(sal_data,bonus_data,by=c("Employee_ID"),  all.x=TRUE)
```

```
  Employee_ID First_Name Last_Name Basic_Salary Bonus
1      E-1001     Mahesh     Joshi        16070 16070
2      E-1002     Rajesh     Kolte        14960    NA
3      E-1004      Priya      Jain        13490 13490
4      E-1005      Sneha     Joshi        15660    NA
5      E-1007        Ram    Kanade        15850    NA
6      E-1008      Nishi    Honrao        15880 15880
7      E-1009     Hameed     Singh        15120    NA
```

rightjoin

```
# Right Join includes all employee ID’s from second data set
rightjoin<-merge(sal_data,bonus_data, by="Employee_ID" ,  all.y=TRUE)
```

```
  Employee_ID First_Name Last_Name Basic_Salary Bonus
1      E-1001     Mahesh     Joshi        16070 16070
2      E-1004      Priya      Jain        13490 13490
3      E-1008      Nishi    Honrao        15880 15880
4      E-1003       <NA>      <NA>           NA 15200
5      E-1006       <NA>      <NA>           NA 14200
6      E-1010       <NA>      <NA>           NA 15120
```

Appending two datasets using rbind function requires both the datasets with exactly the same number of variables with exactly the same names.
If datasets do not have the same number of variables, variables can be either dropped or created so both match.

```
#import new_emp data set
new_emp<-read.csv(file.choose(),header=T)  #append data sets
sal_data<-rbind(sal_data,new_emp)
sal_data
```

```
  Employee_ID First_Name Last_Name Basic_Salary
1      E-1001     Mahesh     Joshi        16070
2      E-1002     Rajesh     Kolte        14960
3      E-1004      Priya      Jain        13490
4      E-1005      Sneha     Joshi        15660
5      E-1007        Ram    Kanade        15850
6      E-1008      Nishi    Honrao        15880
7      E-1009     Hameed     Singh        15120
8      E-1115      Nihar       Rao        16000
9      E-1116     Rajesh Srivastav        14000
```

12. Aggregate using aggregate function

```
A<-aggregate(ba ~ Location, data=salary, FUN = mean )
```

#To calculate mean for variable ‘ba’ by Location variable

#Aggregate function by default ignores the missing data values.

#na.rm=TRUE is not required in mean function.

A

```
  Location       ba
1    DELHI 19430.29
2   MUMBAI 15037.11
```

13. Aggregate Function - Single Variable, Single Factor, Single Function

#To calculate mean for variable ‘ba’ by ‘Location’

A<-aggregate(ba ~ Location, data=salary, FUN = mean)

A

```
  Location       ba
1    DELHI 19430.29
2   MUMBAI 15037.11
```

14. Create your own function which calculates mean, median and standard deviation

#f is the name of your function

f<-function(x) c(mean=mean(x), median=median(x), sd=sd(x))

# Do not forget

na.rm=TRUE if there are missing values

# Apply your function to a variable: f(salary$ba)

```
     mean    median        sd 
17209.744 16230.000  4159.515 
```

#use previously defined function f in aggregate function

B<-aggregate(ba ~ Location,data=salary, FUN = f)

B

```
  Location   ba.mean ba.median     ba.sd
1    DELHI 19430.294 19250.000  4597.886
2   MUMBAI 15037.105 14410.000  2522.308
```

# modify your function to display integer value Output.

f<-function(x) c(mean=round(mean(x),0), median=round(median(x),0),
sd=round(sd(x),0))

```
f<-function(x) c( mean=round(mean(x),0),  median=round(median(x),0),
sd=round(sd(x),0))
C<-aggregate(cbind(ba,ms) ~ Location,data=salary, FUN=f )  C
```

```
  Location ba.mean ba.median ba.sd ms.mean ms.median ms.sd
1    DELHI   19630     19705  4672   13361     14580  3037
2   MUMBAI   14938     14030  2673   10226     10850  2933
```

15. Aggregate Function - Single Variable and Multiple Factors

```
D<-aggregate(ba ~ Location+Grade+Function,data=salary, FUN = f )
```

```
   Location Grade  Function ba.mean ba.median ba.sd
1     DELHI   GR1   FINANCE   21306     19955  3252
2    MUMBAI   GR1   FINANCE   17295     17295  2920
3     DELHI   GR2   FINANCE   15505     15505  1025
4    MUMBAI   GR2   FINANCE   13300     13390   403
5     DELHI   GR1     SALES   23518     23500  4531
6    MUMBAI   GR1     SALES   16939     16815  1792
7     DELHI   GR2     SALES   13760     13760    NA
8    MUMBAI   GR2     SALES   13162     13320   885
9     DELHI   GR1 TECHNICAL   21260     20460  1652
10   MUMBAI   GR1 TECHNICAL   17775     17775   757
11    DELHI   GR2 TECHNICAL   14157     13810   734
12   MUMBAI   GR2 TECHNICAL   11665     11665  1025
```

16. Aggregate Function - Multiple Variables and Multiple Factors

```
E<-aggregate (cbind(ba,ms) ~ Location+Grade+Function,  data=salary, FUN = f )
E
```

```
   Location Grade  Function ba.mean ba.median ba.sd ms.mean ms.median ms.sd
1     DELHI   GR1   FINANCE   21306     19955  3252   15698     15430   897
2    MUMBAI   GR1   FINANCE   17295     17295  2920   12030     12030  1442
3     DELHI   GR2   FINANCE   14780     14780    NA    9300      9300    NA
4    MUMBAI   GR2   FINANCE   13300     13390   403    9407     10580  2351
5     DELHI   GR1     SALES   23518     23500  4531   13369     14305  3230
6    MUMBAI   GR1     SALES   17543     17830  1619    9403     12420  5815
7     DELHI   GR2     SALES   13760     13760    NA   13220     13220    NA
8    MUMBAI   GR2     SALES   12867     13140   806    9480      9800  1466
9     DELHI   GR1 TECHNICAL   21260     20460  1652   15310     15850   961
10   MUMBAI   GR1 TECHNICAL   17775     17775   757   13220     13220     0
11    DELHI   GR2 TECHNICAL   14157     13810   734    9687     10680  2503
12   MUMBAI   GR2 TECHNICAL   11665     11665  1025    8895      8895  3203
```

17. Generating Frequency Tables

```
freq<-table(salary$Location, salary$Grade)
```

```
prop.table(freq)
```

```
                GR1       GR2
  DELHI  0.2972973 0.1621622
  MUMBAI 0.2702703 0.2702703
```

18. Disect data with quantiles

- **Quartiles divide the distribution into 4 equal parts. Q1: Lower Quartile(25% observations are below Q1) Q3: Upper Quartile (25% observations are above Q3) Q2 is same as median**
- **Deciles divide the distribution into 10 equal parts.**
- **5th Decile is same as median**
- **Percentiles divide the distribution into 100 equal parts.**
- **50th percentile is same as median**
- **75th percentile is same as Q3**

```
# Import basic_salary2 data and store in object salary
quantile(salary$ba,na.rm=T)
```

```
   0%   25%   50%   75%  100% 
10940 13785 16230 19305 29080 
```

```
quantile(salary$ba,prob=c(0.1,0.5,0.8),na.rm=T)
```

```
  10%   50%   80% 
13084 16230 20280 
```

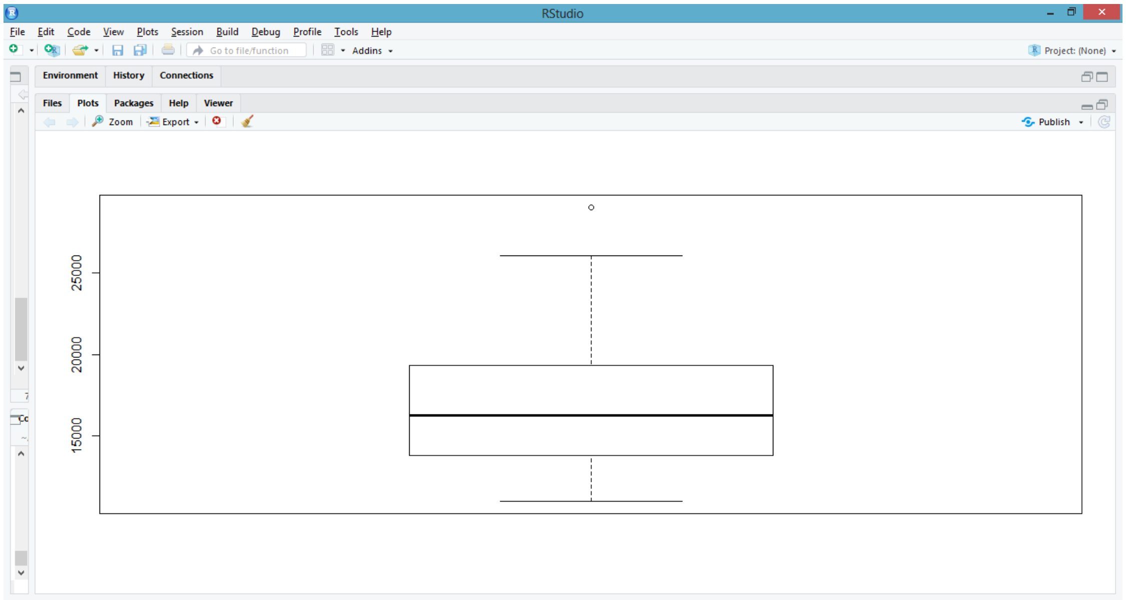
19. Defining Outliers

- **An outlier is an observation that lies an abnormal distance from other values in a random sample from a population.**
- **Before abnormal observations can be singled out, it is necessary to characterize normal observations.**
- **non-Outlier observation is >= Q1 – 1.5*IQR and <= Q3 + 1.5*IQR**
- **where IQR: Inter-quartile Range =Q3-Q1**

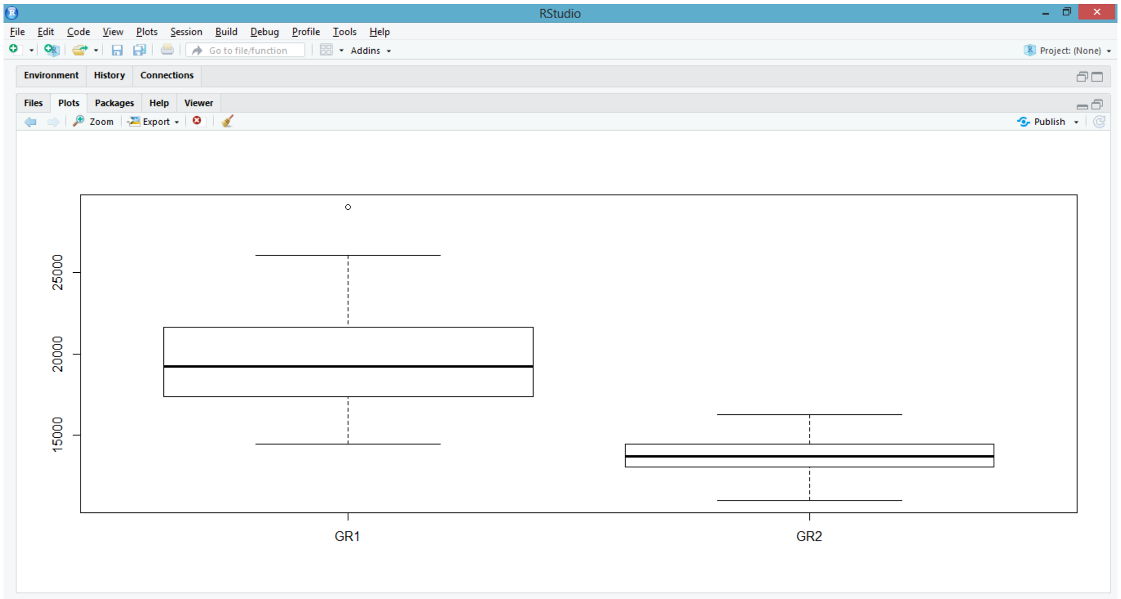
20. Box-Whisker Plot

- **Box and Whisker plot summarizes data graphically using 5 measures: Minimum,Q1,Q2,Q3 and Maximum.**
- **The body of the box goes from the first quartile (Q1) to the third quartile (Q3).**
- **The whiskers go from Q1 to smallest non outlier and Q3 to highest non outlier data points.**

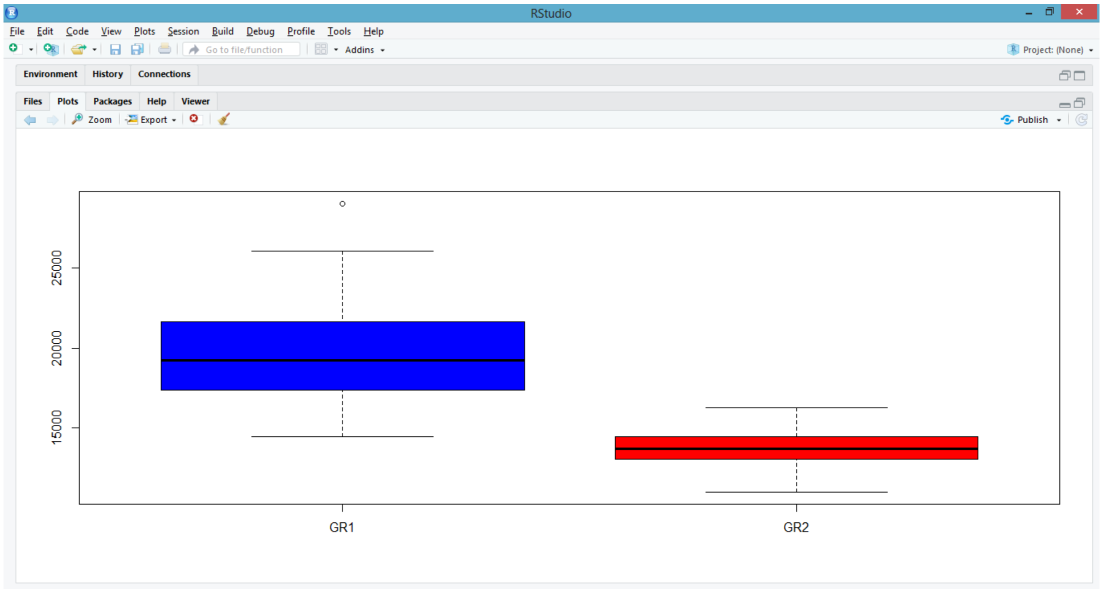
boxplot(salary$ba)


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Environment History Connections
Files Plots Packages Help Viewer
Zoom Export
Publish
15000
20000
25000

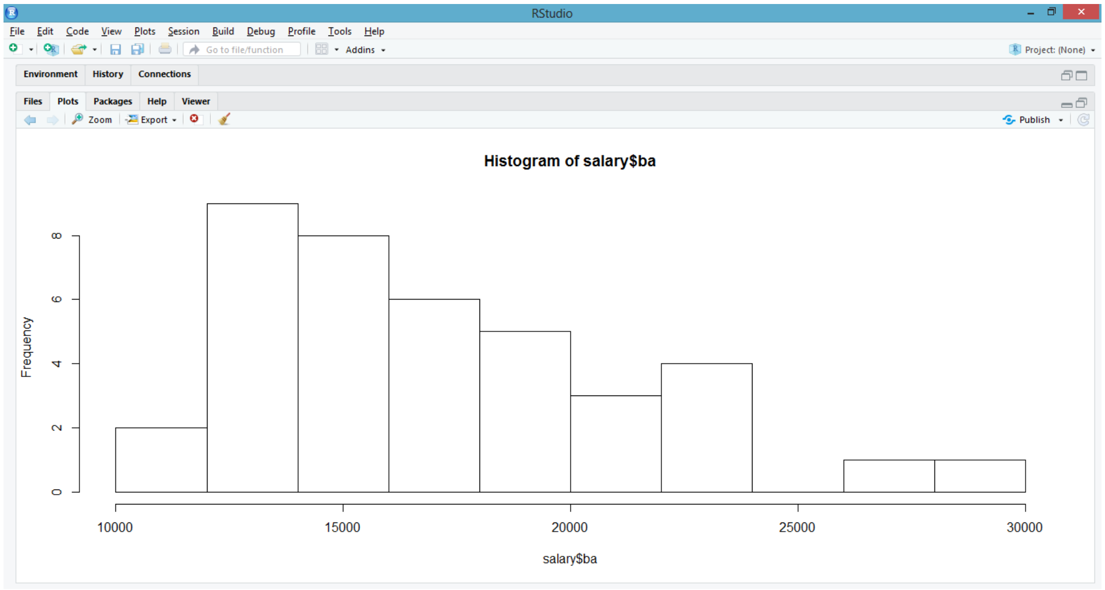
boxplot(ba~Grade,data=salary)


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Environment History Connections
Files Plots Packages Help Viewer
Zoom Export
Publish
15000
20000
25000
GR1
GR2

21. Adding Colour to boxplot
boxplot(ba~Grade,data=salary,col=(c("blue","red")))

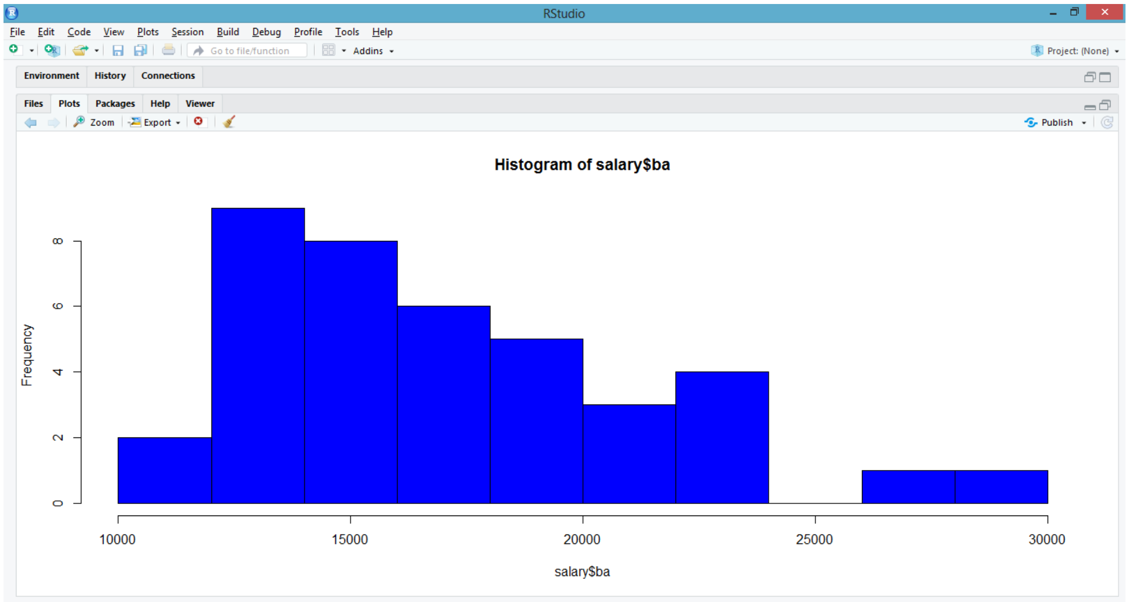

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Environment History Connections
Files Plots Packages Help Viewer
Zoom Export
Publish
15000
20000
25000
GR1
GR2

22. Histogram is useful in visualizing a distribution.
To construct a histogram, the first step is to "bin" the range of values—that is, divide the entire range of values into a series of intervals—and then count how many values fall into each interval(frequency)
hist(salary$ba)



23. Adding Colour in Histogram

```
hist(salary$ba,col="blue")
```

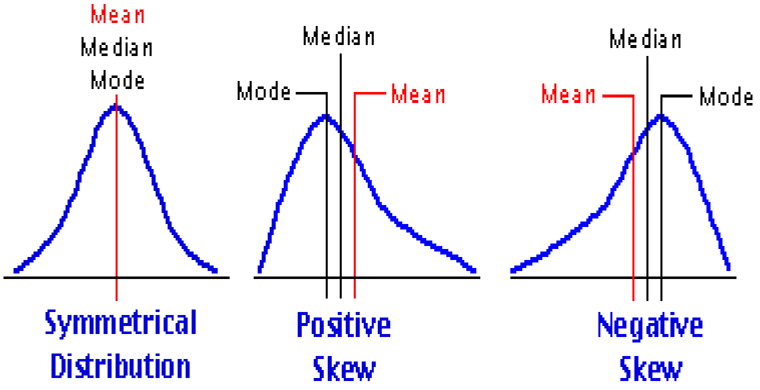


24. Skewness

- Skewness is a measure of 'lack of symmetry' of the data.
- positive skew: The right tail is longer; the mass of the distribution is concentrated on the left.
- negative skew: The left tail is longer; the mass of the distribution is concentrated on the right.
- If the distribution is symmetric, then the mean is equal to the median, and the distribution has zero skewness.
- The Normal distribution is symmetric distribution.

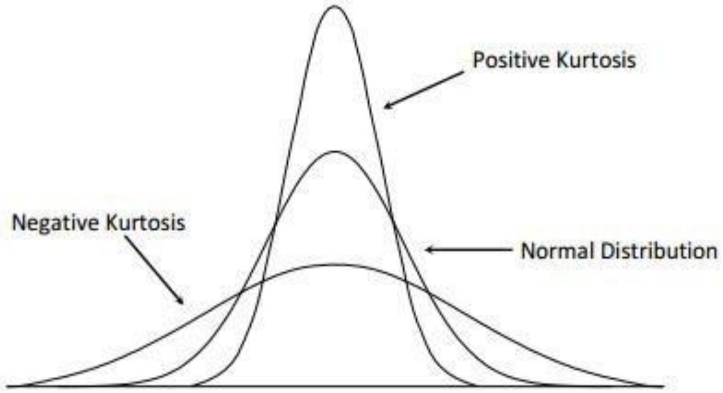
Visualizing Skewness

- The Pearson measure of skewness is defined as (mean − mode) / standard deviation.



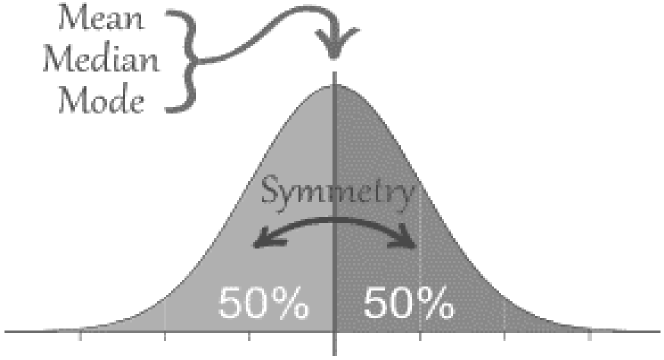
25. Kurtosis

- **Kurtosis is defined as a measure of "peakedness'.**
- **Kurtosis is generally measured relative to Normal distribution, which means excess of kurtosis is measured.**
- **Normal distribution is termed as mesokurtic distribution.**
- **A leptokurtic distribution has a more acute peak, positive kurtosis**
- **A platykurtic distribution has a flatter peak, negative kurtosis.**



Normal Distribution

- **Normal distribution used for continuous variables.**
- **Normal curve is a symmetric bell-shaped curve.**
- **It is also known as the Gaussian distribution.**
- **Many statistical methods assume that population is normally distributed.**



#Measure skewness and kurtosis
install.packages(“e1071")
library(e1071)
#use basic_salary2 data

```
skewness(salary$ba,na.rm=T,type=2)
[1] 0.9033507
```

```
kurtosis(salary$ba,na.rm=T,type=2)
[1] 0.4996513
```

```
f<-function(x)skewness(x,na.rm=T,type=2)
aggregate(ba~Grade,data=salary,FUN=f)
```

```
  Grade         ba
1   GR1 0.85500651
2   GR2 0.08682743
```

```
f1<-function(x)kurtosis(x,na.rm=T,type=2)
aggregate(ba~Grade,data=salary,FUN=f1)
```

```
  Grade        ba
1   GR1 0.6293901
2   GR2 0.3485238
```