Data Analysis and Documentation with R

Athar Ali Khan: Email [atharkhan1962@gmail.com](mailto:atharkhan1962@gmail.com)

November 13, 2019

Table of Contents

[Introduction to R Language 2](#_Toc26304143)

[Basics of R 2](#_Toc26304144)

[Creating a vector—c() 2](#_Toc26304145)

[Getting summary features of a data vector 3](#_Toc26304146)

[Defining a new function—function() 4](#_Toc26304147)

[Naming a vector and get barplot—barplot() 4](#_Toc26304148)

[Factor vector or categorical vector—factor() 5](#_Toc26304149)

[Concept of a factor vector 5](#_Toc26304150)

[Study relation between variables—cov(x,y) and cor(x,y) 6](#_Toc26304151)

[Get Scatter plot and fitted line—plot() 6](#_Toc26304152)

[Get plotting characters and colour options 7](#_Toc26304153)

[Creation of a matrix—matrix() and cbind(), rbind() 8](#_Toc26304154)

[Operations on matrices—%\*%,\*,t(), det(), `solve() 9](#_Toc26304155)

[Creation of listed data—list() 9](#_Toc26304156)

[Creation of data frame—data.frame() and read.table() 10](#_Toc26304157)

[Having a nice output—kable() of package knitr 10](#_Toc26304158)

[Creation of a data frame from Excel 11](#_Toc26304159)

[Graphics with R 11](#_Toc26304160)

[The function curve 11](#_Toc26304161)

[Define a new function and plot it 12](#_Toc26304162)

[Students’ t test—t.test() 13](#_Toc26304163)

[Students’ t test for stacked data 14](#_Toc26304164)

[Graphic output of Students’t analysis 16](#_Toc26304165)

[Another plot plot.design 16](#_Toc26304166)

[Analysis of variance 17](#_Toc26304167)

[Calcium trial data 17](#_Toc26304168)

[Analysis of variance—aov() 18](#_Toc26304169)

[Tuekeys Honest Significant Difference Analysis—TukeyHSD() 18](#_Toc26304170)

[Plot of TukeyHSD 19](#_Toc26304171)

[Bayesian Modeling with rstanarm package 20](#_Toc26304172)

[Find out the results of Bauyesian analysis 20](#_Toc26304173)

[Get a graphic output of it 20](#_Toc26304174)

[Intervals poterior and prective interval—posterior\_interval(), predictive\_interval 21](#_Toc26304175)

[Bayesian analysis of Calcium data—one way ANOVA 22](#_Toc26304176)

[Fit it with—stan\_glm() 22](#_Toc26304177)

[Print summary of results 22](#_Toc26304178)

[Get graphic output 23](#_Toc26304179)

[Posterior predictive check—pp\_check() 24](#_Toc26304180)

[Bayesian LOOIC—loo package 24](#_Toc26304181)

[Regression Analysis—lm() 25](#_Toc26304182)

# Introduction to R Language

R is a language as well as a software. It was introduced by Ross Ihaka and Robert Gentleman in 1996.

## Basics of R

R works like a calculator also. Let us create first chunk.

x=5  
y=2  
z=x+y  
z

## [1] 7

## Creating a vector—c()

The function c() is used to create a vector. We can do operations on a vector also. We get arithmetic transformation of data vectors also.

x=c(2,5,4,3,4)  
y=c(3,6,5,6,10)  
log(x)

## [1] 0.6931472 1.6094379 1.3862944 1.0986123 1.3862944

log(x,base=10)

## [1] 0.3010300 0.6989700 0.6020600 0.4771213 0.6020600

log(x,base=2)

## [1] 1.000000 2.321928 2.000000 1.584963 2.000000

exp(x) #antilog

## [1] 7.389056 148.413159 54.598150 20.085537 54.598150

exp(log(x))# report orginal numbers

## [1] 2 5 4 3 4

## Getting summary features of a data vector

We get summary features of a data vectors, like mean, sd, median and quartiles.

mean(x)

## [1] 3.6

mean(y)

## [1] 6

sd(x)

## [1] 1.140175

sd(y)

## [1] 2.54951

median(x)

## [1] 4

quantile(x)

## 0% 25% 50% 75% 100%   
## 2 3 4 4 5

quantile(x,prob=c(.25,.5,.75)) #quartiles

## 25% 50% 75%   
## 3 4 4

min(x)

## [1] 2

max(x)

## [1] 5

#get length of the vector  
length(x)

## [1] 5

## Defining a new function—function()

In this subsection we define functions for computing standard error of mean sem() and coefficient of variation cv(). The function dumped using the function dump() and it can be sourced using the function source().

#Define a function for standard error of mean  
sem=function(x) sd(x)/sqrt(length(x))  
#dump it as a text file  
dump("sem",file="sem.txt")  
#source it  
source("sem.txt")  
# Define a function for coefficient of variation  
cv=function(x) sd(x)/mean(x)\*100  
dump("cv",file="cv.txt")  
source("cv.txt")  
# let us assume height of 6 students in cm is  
height=c(168,165,162,170,178,169)  
dump("height",file="height.dat")  
source("height.dat")  
#get their sem and cv  
sem(height)

## [1] 2.216103

cv(height)

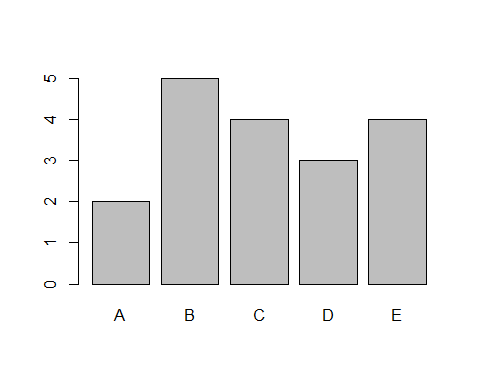
## [1] 3.218372

## Naming a vector and get barplot—barplot()

x=c(2,5,4,3,4)  
names(x)=c("A","B","C","D","E")  
x

## A B C D E   
## 2 5 4 3 4

barplot(x)



#save it as png file  
png("FigBarplot.png")  
barplot(x)  
dev.off()

## png   
## 2

# Factor vector or categorical vector—factor()

## Concept of a factor vector

A factor vector is a kind of categorical vector in R. It is termed as qualitative variable also. For example, gender is a factor variable, its labels are male and female. Income is also a factor variable and its labels are low, median and high. It is created by using the function factor. It may be noted that factor variables are treated **differently** in R for analysis and graphics.

#create a vector  
male=c(0,0,1,0,1,1)  
male

## [1] 0 0 1 0 1 1

#Change it into a facor vector  
Fmale=factor(male,labels=c("female","male"))  
Fmale

## [1] female female male female male male   
## Levels: female male

# Create a vector of rating scale  
rating=c(3,1,2,2,3,1,1)  
rating

## [1] 3 1 2 2 3 1 1

Frating=factor(rating,labels=c("good","okay","bad"))  
Frating

## [1] bad good okay okay bad good good  
## Levels: good okay bad

# Study relation between variables—cov(x,y) and cor(x,y)

We can study relation between x and y

cov(x,y) # covariance between x and y

## [1] 1.5

cor(x,y) # correlation between x and y

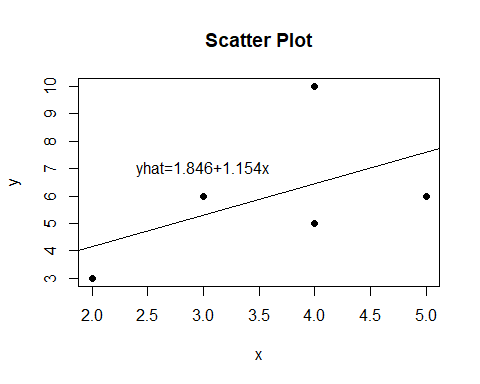
## [1] 0.5160157

## Get Scatter plot and fitted line—plot()

x=c(2,5,4,3,4)  
y=c(3,6,5,6,10)  
#Fit a line   
M1=lm(y~x)  
coef(M1)

## (Intercept) x   
## 1.846154 1.153846

#get aplot  
plot(x,y,pch=16,main="Scatter Plot")  
# add a fitted line  
abline(M1)  
text(3,7,"yhat=1.846+1.154x")

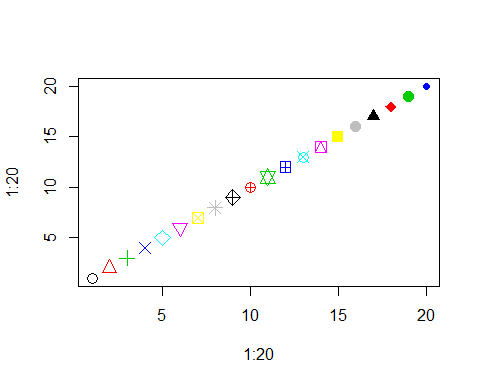


# Save it as png file  
png("Fig1.png")  
plot(x,y,pch=16,main="Scatter Plot")  
# add a fitted line  
abline(M1)  
text(3,7,"yhat=1.846+1.154x")  
dev.off()

## png   
## 2

## Get plotting characters and colour options

plot(x=1:20,y=1:20,pch=1:20,cex=1.5,col=1:20)



## save the plot as a pdf file  
pdf("FigPlottingCharacters.pdf")  
plot(x=1:20,y=1:20,pch=1:20,cex=1.5,col=1:20)  
dev.off()

## png   
## 2

# Creation of a matrix—matrix() and cbind(), rbind()

The function cbind() is meant for column wise binding whereas rbind() is meant for row wise binding.

x1=c(2,4)  
x2=c(5,4)  
cmat1=cbind(x1,x2) # columnwise binding   
cmat1

## x1 x2  
## [1,] 2 5  
## [2,] 4 4

rmat1=rbind(x1,x2) # rowwise binding  
rmat1

## [,1] [,2]  
## x1 2 4  
## x2 5 4

## Operations on matrices—%\*%,\*,t(), det(), `solve()

cmat1%\*%rmat1 # matrix multiplication

## [,1] [,2]  
## [1,] 29 28  
## [2,] 28 32

cmat1\*rmat1 # element wise multiplication

## x1 x2  
## [1,] 4 20  
## [2,] 20 16

t(cmat1) # transpose

## [,1] [,2]  
## x1 2 4  
## x2 5 4

det(cmat1) #determinant

## [1] -12

solve(rmat1) # inverse

## x1 x2  
## [1,] -0.3333333 0.3333333  
## [2,] 0.4166667 -0.1666667

## Creation of listed data—list()

Listed data combines different data types. For example, it can combine a numeric vector, a character string and a matrix together in one object.

x=c(2,4,5)  
x2=c("Radha", "Salman")  
mat3=matrix(c(1,2,5,10),ncol=2)  
LD1=list(x=x,x2=x2,mat3=mat3)  
LD1$mat3

## [,1] [,2]  
## [1,] 1 5  
## [2,] 2 10

det(LD1$mat3)

## [1] 0

# Creation of data frame—data.frame() and read.table()

Data frame is like a data matrix. It is a kind of generalization of a matrix. In matrix all the elements must be of same type, here in a data frame a numeric column can be combined together with a column of logical or binary, with a column of character string.

height=c(168,165,170,162,168) #numeric  
weight=c(60,55,68,57,58) #numeric  
female=c(0,1,0,0,1) #binary  
name=c("Ram","Nazia","Asjad","Ajmal","Radhika") #Character string   
  
#Combnie all the above in a single object, whci is a data frame  
DF1=data.frame(height,weight,female,name)  
print(DF1)

## height weight female name  
## 1 168 60 0 Ram  
## 2 165 55 1 Nazia  
## 3 170 68 0 Asjad  
## 4 162 57 0 Ajmal  
## 5 168 58 1 Radhika

## Save it using write.table()  
write.table(DF1, file="DF1.txt")  
  
# Create a data frame from text file DF1.txt using read.table()  
DF1=read.table("DF1.txt",header=TRUE)  
DF1

## height weight female name  
## 1 168 60 0 Ram  
## 2 165 55 1 Nazia  
## 3 170 68 0 Asjad  
## 4 162 57 0 Ajmal  
## 5 168 58 1 Radhika

The function read.table() can be used to read data from a website also, just give the full path of the file, like https//.../DF1.

## Having a nice output—kable() of package knitr

To have a nice output one can use the function kable() of kintr package.

Data Frame with kable

|  |  |  |  |
| --- | --- | --- | --- |
| height | weight | female | name |
| 168 | 60 | 0 | Ram |
| 165 | 55 | 1 | Nazia |
| 170 | 68 | 0 | Asjad |
| 162 | 57 | 0 | Ajmal |
| 168 | 58 | 1 | Radhika |

## Creation of a data frame from Excel

We can create data into Excel and call into R as a data frame. Note that data in Excel should be entered with first row as name of the variables. After entering data it should be saved as *comma delimited* that is as a .csv file in same working folder, and then it could be called by R using the function read.csv.

Weightheight=read.csv("Weightheight.csv")  
Weightheight

## Hieght Weight Gender  
## 1 168 55 male  
## 2 170 60 female  
## 3 180 68 female  
## 4 165 62 male  
## 5 168 63 male

names(Weightheight)=c("Height", "Weight", "Gender")  
WH=Weightheight  
WH

## Height Weight Gender  
## 1 168 55 male  
## 2 170 60 female  
## 3 180 68 female  
## 4 165 62 male  
## 5 168 63 male

write.csv(WH, file="WH.csv",row.names=FALSE)

# Graphics with R

There are three main graphics in R

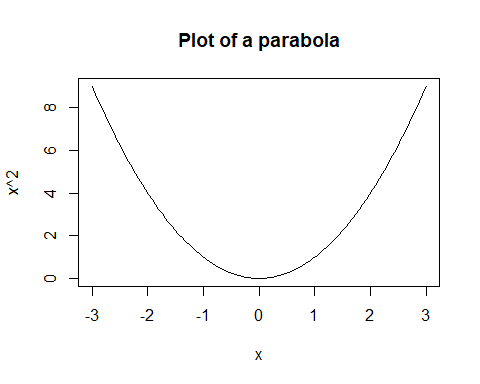
* Base graphics due to Robert Gentleman
* lattice package due to Deepayan Sarkar
* ggplot2 due Wickhem.

We shall discuss only the base graphics in this document.

## The function curve

It is a general purpose plotting function. It requires only expression or function to be plotted along with its range on x-axis. A simple example is plotting of a parabola.

curve(x^2, from=-3,to=3, main="Plot of a parabola")



#to save it as png file  
png("FigParabola.png")  
curve(x^2, from=-3,to=3, main="Plot of a parabola")  
dev.off()

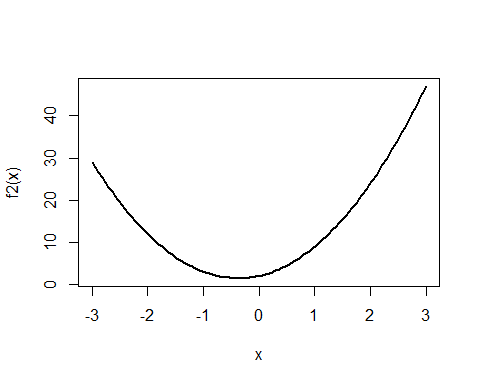
## png   
## 2

## Define a new function and plot it

Suppose you want to plot a function

You can do that by defining the function in R which can be plotted using the function curve as:

# Define the function  
f2=function(x) 2+3\*x+4\*x^2  
#plot it in the range of -3 to to 3  
curve(f2(x), from=-3,to=3, lwd=2)



## Students’ t test—t.test()

Students’ t test is used for comparing two means. We want to compare the two groups of lazy and sporty patients in terms of their average weights(in kg). Data and its analysis is reported in the following chunk.

wt\_lazy=c(76,101,66,72,88,82,79,73,76,85,75,64,76,81,86)  
wt\_sporty=c(64,65,56,62,59,76,66,82,91,57,92,80,82,67,54)  
length(wt\_lazy)

## [1] 15

length(wt\_sporty)

## [1] 15

# Use t.test() to compare means  
t.test(wt\_lazy, wt\_sporty, var.equal = TRUE)

##   
## Two Sample t-test  
##   
## data: wt\_lazy and wt\_sporty  
## t = 2.0969, df = 28, p-value = 0.04516  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.1956937 16.7376396  
## sample estimates:  
## mean of x mean of y   
## 78.66667 70.20000

# Same result can be obtained by organizing data in a data frame  
weight\_ls=data.frame(wtl=wt\_lazy, wts=wt\_sporty)  
head(weight\_ls)

## wtl wts  
## 1 76 64  
## 2 101 65  
## 3 66 56  
## 4 72 62  
## 5 88 59  
## 6 82 76

# Use t test for this data frame  
with(weight\_ls,t.test(wtl,wts, var.equal = TRUE))

##   
## Two Sample t-test  
##   
## data: wtl and wts  
## t = 2.0969, df = 28, p-value = 0.04516  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.1956937 16.7376396  
## sample estimates:  
## mean of x mean of y   
## 78.66667 70.20000

## Students’ t test for stacked data

We stack the data using the command stack so that one column represents weight and other column represents grouping factor women.

wtstack=stack(weight\_ls)  
head(wtstack)

## values ind  
## 1 76 wtl  
## 2 101 wtl  
## 3 66 wtl  
## 4 72 wtl  
## 5 88 wtl  
## 6 82 wtl

names(wtstack)=c("weight","women")  
head(wtstack)

## weight women  
## 1 76 wtl  
## 2 101 wtl  
## 3 66 wtl  
## 4 72 wtl  
## 5 88 wtl  
## 6 82 wtl

tail(wtstack)

## weight women  
## 25 57 wts  
## 26 92 wts  
## 27 80 wts  
## 28 82 wts  
## 29 67 wts  
## 30 54 wts

str(wtstack)

## 'data.frame': 30 obs. of 2 variables:  
## $ weight: num 76 101 66 72 88 82 79 73 76 85 ...  
## $ women : Factor w/ 2 levels "wtl","wts": 1 1 1 1 1 1 1 1 1 1 ...

# Save the data  
write.table(wtstack, file="wtstack.txt")  
# Now use t test  
M1=t.test(weight~women,var.equal=TRUE, data=wtstack)  
M1

##   
## Two Sample t-test  
##   
## data: weight by women  
## t = 2.0969, df = 28, p-value = 0.04516  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.1956937 16.7376396  
## sample estimates:  
## mean in group wtl mean in group wts   
## 78.66667 70.20000

names(M1)

## [1] "statistic" "parameter" "p.value" "conf.int" "estimate"   
## [6] "null.value" "alternative" "method" "data.name"

M1$p.value

## [1] 0.04516067

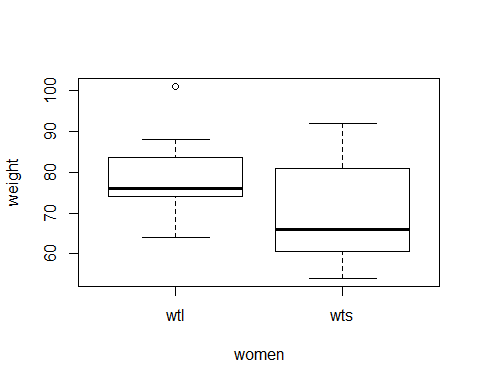
M1$conf.int

## [1] 0.1956937 16.7376396  
## attr(,"conf.level")  
## [1] 0.95

## Graphic output of Students’t analysis

Box plot is the best summary of this kind of data.

wtstack=read.table("wtstack.txt", header=TRUE)  
plot(weight~women, data=wtstack)

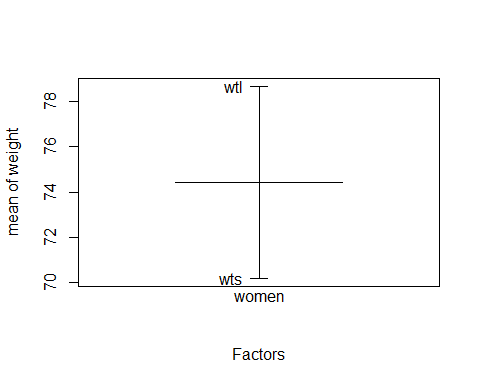


From this plot it is evident that the weight of sporty women is less than that lazy. This difference is statistically significant also. It is evident that both the weights are positively skewed.

## Another plot plot.design

In plot.design the main argument is the data frame.

plot.design(wtstack)



In the above plot the big horizontal line represents overall mean, whereas small horizontal lines represent weight of sporty wts and lazy wtl women. From this plot it is evident that sporty has much less weight than the lazy.

# Analysis of variance

To compare more than two groups t test can not be used and one has to go for analysis of variance technique.

## Calcium trial data

In a calcium trial data 4 levels of concentrations of calcium were used as A=1, B=5, C=10, D=20; root length(cm) was the response. Create data and analyze it into R using analysis of variance technique.

Calcium=expand.grid(Replicate=paste0("R",1:5), Concentration=c("A","B","C","D"))  
head(Calcium)

## Replicate Concentration  
## 1 R1 A  
## 2 R2 A  
## 3 R3 A  
## 4 R4 A  
## 5 R5 A  
## 6 R1 B

# Add a column of Length in the data frame Calcium using $ operator  
Calcium$Length=c(58,52,74,58,79,  
 80,68,72,74,85,  
 49,70,72,74,71,  
 47,49,45,48,38)  
head(Calcium)

## Replicate Concentration Length  
## 1 R1 A 58  
## 2 R2 A 52  
## 3 R3 A 74  
## 4 R4 A 58  
## 5 R5 A 79  
## 6 R1 B 80

# save it to a text file "Calcium.txt"  
write.table(Calcium, file="Calcium.txt")

## Analysis of variance—aov()

Calcium=read.table("Calcium.txt", header=TRUE)  
names(Calcium)

## [1] "Replicate" "Concentration" "Length"

# Fit a model  
calaov=aov(Length~Concentration, data=Calcium)  
summary(calaov)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Concentration 3 2463 821.0 10.75 0.000409 \*\*\*  
## Residuals 16 1222 76.4   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

From above summary of results it is evident that there is significant difference among the four group of concentrations of Calcium as p value, Pr(>F)=0.000409 which is much less than 0.05. Now we shall have to look into the pairwise comparisons using TukeyHSD command.

## Tuekeys Honest Significant Difference Analysis—TukeyHSD()

TukeyHSD requires first of its argument, a fitted object with aov and second argument which which decides for which factor TukyHSD will be implemented.

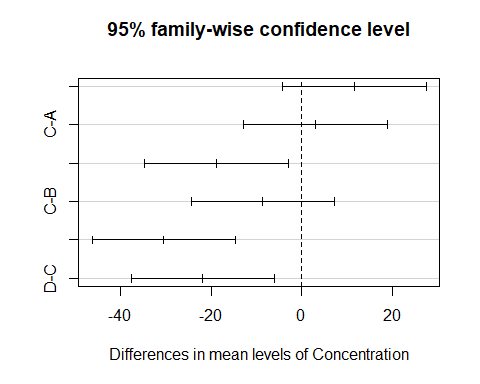
out1=TukeyHSD(calaov,which="Concentration")  
out1

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = Length ~ Concentration, data = Calcium)  
##   
## $Concentration  
## diff lwr upr p adj  
## B-A 11.6 -4.210856 27.410856 0.1954073  
## C-A 3.0 -12.810856 18.810856 0.9471343  
## D-A -18.8 -34.610856 -2.989144 0.0172308  
## C-B -8.6 -24.410856 7.210856 0.4294993  
## D-B -30.4 -46.210856 -14.589144 0.0002558  
## D-C -21.8 -37.610856 -5.989144 0.0057328

From the p value it is evident that D-A, D-B and D-C are statistically significant contrasts. It is also to be noted that corresponding intervals of these contrasts do not contain zero. On the other hand other contrasts are not significant as their corresponding intervals contain zero and p values are greater than 0.05. These facts can be nicely depicted in a graph.

## Plot of TukeyHSD

plot(out1)



From above plot it is clear that the vertical dashed line at zero is not crossing the intervals of significant contrasts.

# Bayesian Modeling with rstanarm package

Bayesian statistics is an approach to statistics which formally seeks use of prior information along with the data. These two sources of information are combined together to reach final inference.

require(rstanarm)  
# Fit model for wtstack data  
names(wtstack)  
M1=stan\_glm(weight~women,data=wtstack)

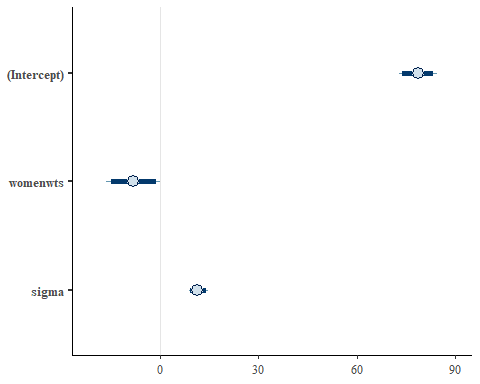
## Find out the results of Bauyesian analysis

summary(M1)

##   
## Model Info:  
##   
## function: stan\_glm  
## family: gaussian [identity]  
## formula: weight ~ women  
## algorithm: sampling  
## priors: see help('prior\_summary')  
## sample: 4000 (posterior sample size)  
## observations: 30  
## predictors: 2  
##   
## Estimates:  
## mean sd 2.5% 25% 50% 75% 97.5%  
## (Intercept) 78.6 3.0 72.8 76.6 78.6 80.4 84.4   
## womenwts -8.3 4.2 -16.4 -11.1 -8.3 -5.6 0.0   
## sigma 11.4 1.5 8.8 10.2 11.2 12.3 14.7   
## mean\_PPD 74.5 2.9 68.8 72.5 74.5 76.4 80.4   
## log-posterior -123.1 1.3 -126.4 -123.7 -122.7 -122.1 -121.6   
##   
## Diagnostics:  
## mcse Rhat n\_eff  
## (Intercept) 0.0 1.0 3642   
## womenwts 0.1 1.0 3656   
## sigma 0.0 1.0 3327   
## mean\_PPD 0.0 1.0 3616   
## log-posterior 0.0 1.0 1670   
##   
## For each parameter, mcse is Monte Carlo standard error, n\_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).

## Get a graphic output of it

plot(M1,prob=0.95)



## Intervals poterior and prective interval—posterior\_interval(), predictive\_interval

#get the 95% credible interval  
posterior\_interval(M1,prob=.95)

## 2.5% 97.5%  
## (Intercept) 72.773350 84.39192035  
## womenwts -16.354889 0.03661925  
## sigma 8.793902 14.73868201

#get predictive interval  
predictive\_interval(M1,prob=0.95)

## 2.5% 97.5%  
## 1 54.98381 101.56543  
## 2 55.25650 102.55710  
## 3 55.03550 102.23846  
## 4 54.95031 102.13908  
## 5 55.34485 103.04446  
## 6 55.15665 102.70747  
## 7 55.16869 101.26864  
## 8 55.70168 102.97061  
## 9 55.45342 102.16876  
## 10 55.24857 102.02172  
## 11 56.29969 102.50884  
## 12 54.98455 101.53882  
## 13 55.23073 101.78191  
## 14 55.32650 102.46172  
## 15 55.07867 102.39101  
## 16 47.45656 92.76241  
## 17 48.30283 92.61788  
## 18 46.97429 93.11755  
## 19 46.55027 95.36813  
## 20 46.50444 93.41340  
## 21 46.86115 94.30181  
## 22 47.68971 93.44379  
## 23 47.20425 92.48104  
## 24 46.02651 93.01476  
## 25 46.49693 93.32821  
## 26 47.66294 93.64801  
## 27 46.22713 94.21335  
## 28 47.82677 93.20862  
## 29 47.40380 93.67325  
## 30 48.03862 93.28681

## Bayesian analysis of Calcium data—one way ANOVA

Calcium=read.table("Calcium.txt",header=TRUE)  
str(Calcium)

## 'data.frame': 20 obs. of 3 variables:  
## $ Replicate : Factor w/ 5 levels "R1","R2","R3",..: 1 2 3 4 5 1 2 3 4 5 ...  
## $ Concentration: Factor w/ 4 levels "A","B","C","D": 1 1 1 1 1 2 2 2 2 2 ...  
## $ Length : int 58 52 74 58 79 80 68 72 74 85 ...

### Fit it with—stan\_glm()

Mcalc=stan\_glm(Length~Concentration,data=Calcium)

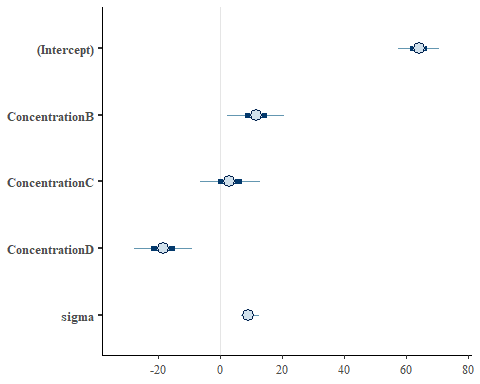
### Print summary of results

summary(Mcalc)

##   
## Model Info:  
##   
## function: stan\_glm  
## family: gaussian [identity]  
## formula: Length ~ Concentration  
## algorithm: sampling  
## priors: see help('prior\_summary')  
## sample: 4000 (posterior sample size)  
## observations: 20  
## predictors: 4  
##   
## Estimates:  
## mean sd 2.5% 25% 50% 75% 97.5%  
## (Intercept) 64.0 4.1 56.0 61.2 64.2 66.7 72.2   
## ConcentrationB 11.6 5.6 0.2 8.1 11.7 15.3 22.7   
## ConcentrationC 3.1 6.0 -8.4 -0.8 3.0 7.0 15.1   
## ConcentrationD -18.4 5.8 -29.8 -22.2 -18.5 -14.6 -7.3   
## sigma 9.3 1.8 6.5 8.0 9.0 10.3 13.4   
## mean\_PPD 63.1 3.0 57.2 61.2 63.1 65.0 68.8   
## log-posterior -82.4 1.8 -86.8 -83.4 -82.0 -81.1 -80.0   
##   
## Diagnostics:  
## mcse Rhat n\_eff  
## (Intercept) 0.1 1.0 1748   
## ConcentrationB 0.1 1.0 1750   
## ConcentrationC 0.1 1.0 1920   
## ConcentrationD 0.1 1.0 1915   
## sigma 0.0 1.0 2490   
## mean\_PPD 0.0 1.0 3856   
## log-posterior 0.0 1.0 1414   
##   
## For each parameter, mcse is Monte Carlo standard error, n\_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).

## Get graphic output

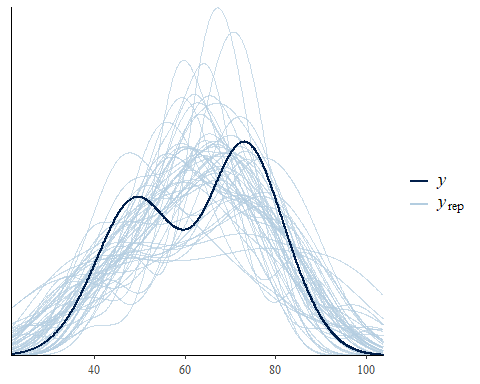
plot(Mcalc)



It is evident from this plot that D-A and B-A are significant, whereas D-C is not significant.

## Posterior predictive check—pp\_check()

pp\_check(Mcalc)



From this output it is evident that there is close agreement between observed and predicted .

## Bayesian LOOIC—loo package

Leave one out criteria is a criteria to check the predictive ability of the fitted model. It is meant for comparing model on the basis of loo criteria. However, here we are fitting it for a single model. For comparing two or more models, lower value of **looic** indicates better fit of the model.

require(loo)  
loo(Mcalc)

##   
## Computed from 4000 by 20 log-likelihood matrix  
##   
## Estimate SE  
## elpd\_loo -75.1 3.3  
## p\_loo 4.4 1.3  
## looic 150.1 6.6  
## ------  
## Monte Carlo SE of elpd\_loo is 0.1.  
##   
## Pareto k diagnostic values:  
## Count Pct. Min. n\_eff  
## (-Inf, 0.5] (good) 18 90.0% 715   
## (0.5, 0.7] (ok) 2 10.0% 160   
## (0.7, 1] (bad) 0 0.0% <NA>   
## (1, Inf) (very bad) 0 0.0% <NA>   
##   
## All Pareto k estimates are ok (k < 0.7).  
## See help('pareto-k-diagnostic') for details.

# Regression Analysis—lm()

Analysis of usair data which is dicussed by Wang et al(2018): Bayesian Regression Modeling with INLA. CRC Press. In this data SO2content in air is the response and there are 5 predictors

require(brinla)  
data(usair)  
names(usair)

## [1] "SO2" "negtemp" "manuf" "pop" "wind" "precip" "days"

M1usair=lm(SO2~.,data=usair)  
summary(M1usair)$coef

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 111.72848064 47.31810073 2.361221 0.0240867374  
## negtemp 1.26794109 0.62117952 2.041183 0.0490557189  
## manuf 0.06491817 0.01574825 4.122245 0.0002277862  
## pop -0.03927674 0.01513274 -2.595482 0.0138461970  
## wind -3.18136579 1.81501910 -1.752800 0.0886503978  
## precip 0.51235896 0.36275507 1.412410 0.1669175999  
## days -0.05205019 0.16201386 -0.321270 0.7499724652

It may be noted that except wind, precip and days all the predictors are significant. We can use stepwise regression to see which variables are to be dropped

library(MASS)  
M1step=stepAIC(M1usair,trace=FALSE)  
M1step$anova

## Stepwise Model Path   
## Analysis of Deviance Table  
##   
## Initial Model:  
## SO2 ~ negtemp + manuf + pop + wind + precip + days  
##   
## Final Model:  
## SO2 ~ negtemp + manuf + pop + wind + precip  
##   
##   
## Step Df Deviance Resid. Df Resid. Dev AIC  
## 1 34 7283.266 226.3703  
## 2 - days 1 22.10994 35 7305.376 224.4945

Thus only days are dropped from the model.