#log of positive number

x = log(2)

print(x)

#default base is exponential function

x = log(exp(2))

print(x)

#switch to base 10

x = log(2, 10)

print(x)

x = log10(2)

print(x)

#range of exponential function is non-negative

x = log(-1)

#square root

x = sqrt(4)

print(x)

# 15 random variable vector

x = rnorm(15)

print(x)

m = mean(x)

s = sd(x)

print(sprintf('mean=%s stDev=%s', m, s))

# 15 random variable vector (mean = 10, sd = 2)

x = rnorm(15, mean=10, sd=2)

print(x)

m = mean(x)

s = sd(x)

print(sprintf('mean=%s stDev=%s', m, s))

#sample mean and variance will differ from generated distrubution.

#increasing sample size will reduce variance

x = rnorm(150000)

m = mean(x)

s = sd(x)

print(sprintf('mean=%s stDev=%s', m, s))

#weights

weights = c(60, 72, 57, 90, 95, 72)

#heights

heights = c(1.80, 1.85, 1.72, 1.90, 1.74, 1.91)

#bmi

bmi = weights/heights

#plot shows a quasi-linear relationsip between height and weight

plot(heights, weights)

#average weight

avg\_weight = mean(weights)

print(sprintf('average weight=%skg', avg\_weight))

weight\_minus\_mean = weights - avg\_weight

print(weight\_minus\_mean)

print(sprintf('mean of weights minus mean=%s', mean(weight\_minus\_mean)))

data\_frame = data.frame('Catgories'=c('computer programming', 'math', 'statistics', 'machine learning', 'domain expertise', 'communication and

presentation skills', 'data visualization'), 'Values'=c(5, 4, 3, 3, 4, 4, 4))

print(data\_frame)

barplot('colnames<-'(t(data\_frame[-1]), data\_frame[,1]), las=2, col = c("green"))