**########## "for-loops" and generating random numbers ##########**

for (i in 1:7){ # print squares from 1 to 7, inclusive

print(c(i, i^2))

}

for (i in c(-4, 10,3,2)){ # we can also use an explicit list...

print(c(i, i ^3))

}

# print from 6 to 2, inclusive and going downwards

# ... the number i and a triplet of random numbers between 7 and 9.76

# drawn from a UNIFORM distribution

for (i in 6:2){

x=runif(3, 7, 9.76)

print(c(i, x))

}

# similarly, to produce pairs of random numbers from the NORMAL distribution

# replace runif() with rnorm(2, 1.1, 2.7) for Gaussian with mean 1.1 and sd 2.7

runif(7, 1, 4) # # seven random numbers in range (1, 4]

trunc(runif(7, 1, 4),0) # seven random integers, either 1, 2, or 3 (NB 4 is never produced)

sample(1:3, replace=T) # similar, but will only produce three numbers. Useful for bootstrapping

**################## BOOTSTRAPPING #############################**

x=rnorm(20,-0.5,1) # produce 20 random norma;l deviates with mean = -0.7 and sd = 1

x[21] = 10 # 21st is an outlier

boxplot(x)

t.test(x)

wilcox.test(x)

median(x)

# (A) do 100 replications and store in yall

yall=0

for (i in 1:100){

y = x[sample(1:21, replace=T)]

yall[i] = quantile(y, .5)

}

boxplot(yall)

# (B) count how many times yall[i] is larger than zero

ycount = 0

for (i in 1:100){

if (yall[i] > 0){

ycount = ycount + 1

}

}

ycount

# Combine (A) and (B) with 10000 replications, and lay it out more neatly

yall = 0

ycount = 0

for (i in 1:10000){

y = x[sample(1:21, replace=T)]

yall[i] = quantile(y, .5) # median

if (yall[i] > 0){

ycount = ycount + 1

}

}

boxplot(yall)

ycount

ycount / 10000 # for a 1-tailed test

# Flexibility of bootrarapping: we could replace quantile(y, .5) with

# (quantile(y, .25) + quantile(y, .5) + quantile(y, .75)) / 3

# for a test that combines Q1, Md, Q3 into a measure of location

**############ Box-Cox transformation ################**

aa=read.table("clipboard",header=T) # read from greenvehicleNOSPACE.xls(x)

A = aa$Displacement

B = aa$CmbMPG

plot(A,B)

points(lowess(A,B),pch=19,col="red")

i = 0

RR = 0

for (L in c(-1, -.5, .00001, .25, .5, .75, 1, 1.5, 2, 2.5, 3)){

i = i + 1

RR[i]= cor.test(PatAb,(WordAb^L-1)/L)$estimate

}

plot(c(-1, -.5, .00001, .25, .5, .75, 1, 1.5, 2, 2.5, 3),RR,pch=19,cex=2, main="Box-Cox transformations\n(finding the best L)", xlab="L",ylab="R")

lines(c(1,1), c(0,1), col="red")

lines(c(0,0), c(0,1), col="blue")

