**Problem #1**

**Q1.**

True mean of the population: 20

Mean of the sample: 19.94794

Standard deviation of the sample: 4.774416

> mymean

[1] 19.94794

> mysd

[1] 4.774416

Both means do not exactly match. But sample mean is within 2 sd from the true mean.

R Code:

# Q1

# infer from one sample

set.seed(1001)

mysample <- rnorm(35, mean=20, sd=4)

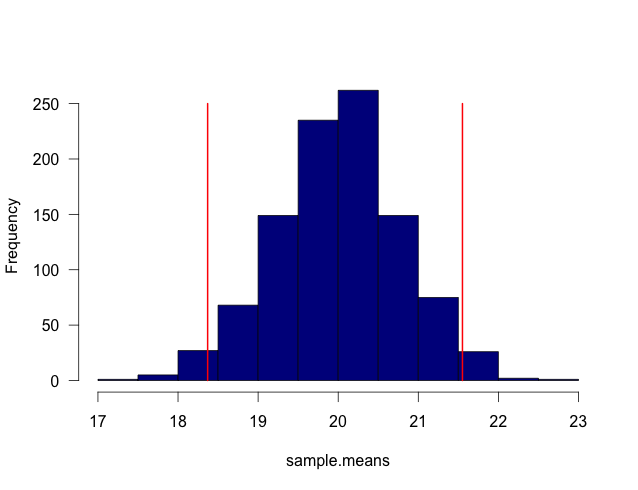
mymean <- mean(mysample)

mysd <- sd(mysample)

zcrit <- pnorm(0.975)

myci <- c(mymean - zcrit \* mysd/sqrt(35), mymean + zcrit \* mysd/sqrt(35))

**Q2.**



> sum(sample.means >= 20 - 2\*4/sqrt(n) & sample.means <= 20 + 2\*4/sqrt(n))/1000

[1] 0.951

About 95.1% of the time the sample mean will be within ±2\*sd of the true sample mean.

R Code:

# Q2

# sample 1000 times

set.seed(1001)

sample.means <- c()

n <- 25

for (i in 1:1000) { sample.means[i] <- mean(rnorm(n, 20, 4))}

hist(sample.means, las = 1, col = "darkblue", main="")

segments(sort(sample.means)[25], 0, sort(sample.means)[25], 250, col="red", lwd=2)

segments(sort(sample.means)[975], 0, sort(sample.means)[975], 250, col="red", lwd=2)

# use true population mean and sd

sum(sample.means >= 20 - 2\*4/sqrt(n) & sample.means <= 20 + 2\*4/sqrt(n))/1000

**Q3.**

> sum(sample.means >= 20 - 4/sqrt(n) & sample.means <= 20 + 4/sqrt(n))/1000

[1] 0.702

70.2% of the time the sample mean will be within ±1\*sd of the true sample mean.

If we estimate from sample mean and sample sd, then:

tcrit <- qt(1-0.05/2, df=n-1)

samp.sd <- sd(sample.means)

samp.mean <- mean(sample.means)

sum(sample.means >= samp.mean - tcrit\*samp.sd & sample.means <= samp.mean + tcrit\*samp.sd)/1000

> sum(sample.means >= samp.mean - tcrit\*samp.sd & sample.means <= samp.mean + tcrit\*samp.sd)/1000

[1] 0.956

About 95.6% of the time the sample mean will be within ±tcrit\*sd of the true sample mean.

**Q4.**

set.seed(1001)

mysample <- rnorm(35, mean=20, sd=4)

mymean <- mean(mysample)

mysd <- sd(mysample)

zcrit <- pnorm(0.975)

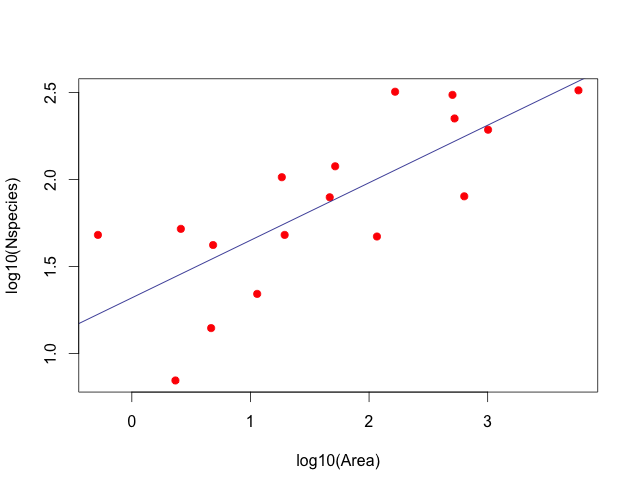
myci <- c(mymean - zcrit \* mysd/sqrt(35), mymean + zcrit \* mysd/sqrt(35))

> myci

[1] 19.27390 20.62198

Which includes the true mean 20.

**Q5.**



The 95% confidence intervals for intercept and slope are:

> ci.int

2.5% 97.5%

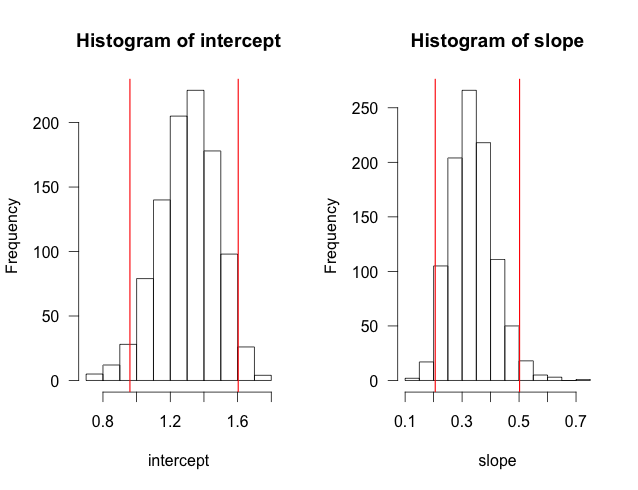
0.9649754 1.6036269

> ci.slope

2.5% 97.5%

0.2059936 0.5017975

Based on the confidence interval, we can say the coefficient values are different from zero.



R Code:

# resampling for two variables

dat <- read.csv("Galapagos.csv", header = T)

attach(dat)

plot(Area, Nspecies, pch=19, col="red")

# linear regression

fit <- lm(Nspecies~Area)

abline(fit, col="darkblue")

# draw the logrithm plot

plot(log10(Area), log10(Nspecies), pch=19, col="red")

#plot(log10(Nspecies)~log10(Area), pch=19, col="red")

# linear regression

Lfit <- lm(log10(Nspecies)~log10(Area))

abline(Lfit, col="darkblue")

Lco <- Lfit$coefficients

# bootstrapping the sample pairs

runs <- 1000

len <- length(Nspecies)

slope <- c()

intercept <- c()

for(i in 1:runs){

samp <- dat[sample(nrow(dat), size=len, replace=T), ]

fit <- lm(log10(samp$Nspecies) ~ log10(samp$Area))

intercept[i] <- fit$coefficients[1]

slope[i] <- fit$coefficients[2]

}

mean.int <- mean(intercept)

mean.slope <- mean(slope)

par(mfrow=c(1,2))

hist(intercept, las=1)

abline(v=sort(intercept)[25], col="red", lwd=1.5)

abline(v=sort(intercept)[975], col="red", lwd=1.5)

ci.int<-c(quantile(intercept, 0.025),

quantile(intercept, 0.975))

hist(slope, las=1)

abline(v=sort(slope)[25], col="red", lwd=1.5)

abline(v=sort(slope)[975], col="red", lwd=1.5)

ci.slope<-c(quantile(slope, 0.025),

quantile(slope, 0.975))

**Problem #2**

**1)**

Mean of the boostrapped mean (13.29232) is close to but different from the mean of original data (13.17).

> mean(insects.sm)

[1] 13.29232

> insects.mean

[1] 13.17

**2)**

The standard error of the bootstrapped mean is 4.204322. By typical parametric formula, the standard error is 4.479427.

> boot.sd

[1] 4.204322

> insects.se

[1] 4.479427

**3)**

90% CI using parametric formula:

> insects.ci

[1] 5.125466 21.214534

90% CI using percentile bootstrapped means:

> insects.boot.ci

5% 95%

6.813333 20.851083

R Code:

# Problem #2

insects <- c(0.14, 15.49, 29.04, 6.36, 1.83, 5.40, 31.89,

3.92, 0.54, 2.01, 12.67, 48.75)

hist(insects, las = 1, ylab = "No. species",

xlab = "Abundance", main="", col="lightgrey")

ins.n <- length(insects)

insects.mean <- mean(insects)

insects.sm <- c()

for (i in 1:1000){

insect.sample <- sample(insects, ins.n, replace=T)

insects.sm[i] <- mean(insect.sample)

}

mean(insects.sm)

boot.sd <- sd(insects.sm)

insects.se <- sd(insects)/sqrt(ins.n)

# parametric approach to c.i.

t <- qt(1 - 0.10/2, df=ins.n-1)

insects.ci <- c(insects.mean - t\*insects.se, insects.mean + t\*insects.se)

# percentile approach to c.i.

insects.boot.ci <- c(quantile(insects.sm, 0.05), quantile(insects.sm, 0.95))