```
pdf("E:\\Rmisc\\Rmisc\\graphs\\sta6207_hw1.pdf")
set.seed(97531)

############### Part 1
n.sim <- c(100,1000,10000,100000)    #### There will be 4 "sample sizes"
mu <- 100; sigma <- 20      #### Assign the mean and standard deviation

#### Normal(mu,sigma)

par(mfrow=c(2,2))         #### Set plots so that there are 2 "rows", 2
"cols"

#### Loop through the 4 "sample sizes"
for (i in 1:length(n.sim)) {
x <- rnorm(n.sim[i],mu,sigma)     ### Draw a sample of n.sim[i] from
N(mu,sigma)
print(cbind(n.sim[i],mean(x),sd(x)))     ### Print Sample size, sample
mean, sample SD
print(quantile(x,c(.025,0.25,0.50,0.75,0.975)))  ### print sample
quantiles

### put limits on y-axis, so top of density and histogram don't get "cut-
off"
### dnordnorm(mu,mu,sigma) is the height of density at its peak (mode)
yxlim <- 1.5*n.sim[i]*dnorm(mu,mu,sigma)

# Histogram of Normal Sample with Normal Super-Imposed
# Frequency histogram (why we multiply density by sample size and binsize
(1))
hist(x,breaks=seq(0,200,1),xlab="x",ylim=c(0,yxlim))
lines(40:160,1*n.sim[i]*dnorm(40:160,mu,sigma),lwd=4)
}  ### End of Loop

print(cbind(mu,sigma))       ### Print theortical mean and SD
qnorm(c(.025,.25,.5,.75,.975),mu,sigma)   ### Print theoretical quantiles

###############################################

### Part 2
set.seed(12345)

### Assign sample sizes, means, SDs, and number of simulations
n1 <- 25;  mu1 <- 60; sigma1 <- 10
n2 <- 15;  mu2 <- 50; sigma2 <- 8
n.sim <- 10000

### Assign "blank vectors" to hold results for sample means and SDs
y1.mean <- numeric(n.sim); y1.sd <- numeric(n.sim)
y2.mean <- numeric(n.sim); y2.sd <- numeric(n.sim)

### Begin looping through samples, saving mean and SD for each group
for (i in 1:n.sim) {
y1 <- rnorm(n1,mu1,sigma1)
y2 <- rnorm(n2,mu2,sigma2)
y1.mean[i] <- mean(y1);   y1.sd[i] <- sd(y1)
y2.mean[i] <- mean(y2);   y2.sd[i] <- sd(y2)
}   ### End Loop

### Compute scaled variance for each group
X1.2 <- (n1-1)*(y1.sd^2)/(sigma1^2)
X2.2 <- (n2-1)*(y2.sd^2)/(sigma2^2)

### Compute t-value for each group
t1 <- (y1.mean - mu1)/(y1.sd/sqrt(n1))
t2 <- (y2.mean - mu2)/(y2.sd/sqrt(n2))

### Compute F-value for each group
F <- ((y1.sd/sigma1)^2)/((y2.sd/sigma2)^2)

### Print sample means and SDs for scaled variance, t, F
### See notes for theoretical values (based on the relevant degrees of
freedom)
mean(X1.2); sd(X1.2)
mean(t1); sd(t1)
mean(F); sd(F)

### Print Sample quantiles
quantile(X1.2,c(.025,.25,.5,.75,.975))
quantile(t1,c(.025,.25,.5,.75,.975))
quantile(F,c(.025,.25,.5,.75,.975))

### Print theoretical quantiles
qchisq(c(.025,.25,.5,.75,.975),n1-1)
qt(c(.025,.25,.5,.75,.975),n1-1)
qf(c(.025,.25,.5,.75,.975),n1-1,n2-1)

### Histogram of scaled variance
### Using X1.2[X1.2 <= 100] leaves out very large values that "stretch
plot out"
hist(X1.2[X1.2 <= 100],breaks=0:100,
main=expression(paste("Sampling Distribution of ",X[1]^2)))
lines(0:100,1*10000*dchisq(0:100,n1-1))

### Histogram of t (Only those between -4, 4)
### Multiplies t-density by binsize*n.sin = 0.05*10000
hist(t1[abs(t1) <= 4],breaks=seq(-4,4,.05),
main=expression(paste("Sampling Distribution of ",t[1])))
lines(seq(-4,4,0.05),0.05*n.sim*dt(seq(-4,4,0.05),n1-1))

### Histogram of F (Only those < 6)
### Multiplies F-density by binsize*n.sinm = 0.05*10000
hist(F[F<=6],breaks=seq(0,6,0.05),main="Sampling Distribution of F")
lines(seq(0,6,0.05),0.05*n.sim*df(seq(0,6,0.05),n1-1,n2-1))


##########################################################################
###
```

```
#### Part 3a
### Read file from website
nhl_ht_wt <-
read.csv("http://www.stat.ufl.edu/~winner/data/nhl_ht_wt.csv",
                       header=T)
attach(nhl_ht_wt); names(nhl_ht_wt)


bmi <- 703*Weight/(Height^2)    ### Create bmi from Height and Weight
(N.bmi <- length(bmi))     ### Population size
(mu.bmi <- mean(bmi))      ### Population mean
(sigma.bmi <- sd(bmi)*sqrt((N.bmi-1)/N.bmi))     ### Population SD (Uses
N as denominator, not N-1)

ybar <- numeric(10000); stddev <- numeric(10000)   # Create vectors to
save sample means, SDs

for (i in 1:10000) {
y <- sample(bmi, 25)    ### Sample n=25 bmi w/out replacement
ybar[i] <- mean(y)
stddev[i] <- sd(y)
}
X2 <- (25-1)*(stddev/sigma.bmi)^2

mean(ybar); sd(ybar)
mean(X2); sd(X2)

quantile(ybar,c(.025,.25,.5,.75,.975))
quantile(X2,c(.025,.25,.5,.75,.975))
qnorm(c(.025,.25,.5,.75,.975),mu.bmi,sigma.bmi/sqrt(25))
qchisq(c(.025,.25,.5,.75,.975),25-1)

hist(ybar[(ybar >= 24.5) & (ybar <= 28.5)],breaks=seq(24.5,28.5,.01),
xlab="Mean")
lines(seq(24.5,28.5,.01),
0.01*10000*dnorm(seq(24.5,28.5,.01),mu.bmi,sigma.bmi/sqrt(25)),lwd=4)

yx2lim <- 1.1*10000*max(dchisq(0:100,25-1))
hist(X2[X2 <= 100],breaks=0:100, ylim=c(0,yx2lim),
main=expression(paste("Sampling Distribution of ",X^2," - BMI Data")))
lines(0:100,1*10000*dchisq(0:100,25-1))



#### Part 3b
rr.mar <-
read.csv("http://www.stat.ufl.edu/~winner/data/rocknroll_marathon_mf2015a
.csv",
                     header=T)
attach(rr.mar); names(rr.mar)

f.mph <- mph[Gender=="F"]      ### Subsets Females from population
m.mph <- mph[Gender=="M"]      ### Subsets Males from population
```

```
(f.mu <- mean(f.mph))
(f.sigma <- sd(f.mph))
(m.mu <- mean(m.mph))
(m.sigma <- sd(m.mph))

hist(f.mph,ylim=c(0,120),breaks=seq(4,11,1/6),main="Females/Normal")
lines(seq(4,11,1/6),dnorm(seq(4,11,1/6),f.mu,f.sigma)*length(f.mph)/6)

hist(m.mph,ylim=c(0,120),breaks=seq(4,11,1/6),main="Males/Normal")
lines(seq(4,11,1/6),dnorm(seq(4,11,1/6),m.mu,m.sigma)*length(m.mph)/6)

## Begin sampling
set.seed(45678)

ybar.f <- numeric(10000); sd.f <- numeric(10000)
ybar.m <- numeric(10000); sd.m <- numeric(10000)

for (i in 1:10000) {
y.f <- sample(f.mph, 20)
y.m <- sample(m.mph, 20)
ybar.f[i] <- mean(y.f); sd.f[i] <- sd(y.f)
ybar.m[i] <- mean(y.m); sd.m[i] <- sd(y.m)
}

X2.f <- (20-1)*(sd.f**2)/(f.sigma**2)
F.fm <- ((sd.f**2)/(f.sigma**2))/((sd.m**2)/(m.sigma**2))

mean(ybar.f); sd(ybar.f)
quantile(ybar.f,c(.025,.25,.5,.75,.975))
qnorm(c(.025,.25,.5,.75,.975),f.mu,f.sigma/sqrt(20))

mean(X2.f); sd(X2.f)
quantile(X2.f,c(.025,.25,.5,.75,.975))
qchisq(c(.025,.25,.5,.75,.975),20-1)

mean(F.fm); sd(F.fm)
quantile(F.fm,c(.025,.25,.5,.75,.975))
qf(c(.025,.25,.5,.75,.975),20-1,20-1)

dev.off()
```