TA: Gina Oh/ gina.oh@wisc.edu
Office Hours: MW 2:30 PM - 3:45 PM / 1219 MSC

1 Writing a R function for the simulation study

HW1 Q3: histograms of the sample means and the sample variances (normal dist).

```
  func.hist.norm <- function(n.simu, n.samp, mu, sig){
  par(mfcol=c(2,length(n.samp)))
  xlim_mean <- c(mu-sig, mu+sig); xlim_var <- c(0, 3*sig^2 )
  ylim_mean <- ylim_var <- c(0, n.simu*.55)
  for (n in n.samp) {
    sample_means <- rep(NA,n.simu)
    sample_vars  <- rep(NA,n.simu)
    for (i in 1:n.simu) {
      sample_norm <- rnorm(n,mean=mu,sd=sig)
      sample_means[i] <- mean(sample_norm)
      sample_vars[i]  <- var(sample_norm)
    }
    hist(sample_means, xlim=xlim_mean, ylim=ylim_mean,
         breaks=seq(xlim_mean[1], xlim_mean[2], length.out = 15), main = paste("n =",n))
    hist(sample_vars,  xlim=xlim_var,  ylim=ylim_var,
         breaks=seq(xlim_var[1], xlim_var[2], length.out = 15), main = paste("n =",n))
  }
  par(mfcol=c(1,1))
}
```

```
n.simu <- 100
n.samp <- c(10, 40, 160)
mu <- 4
sig <- 0.5
set.seed(181828); func.hist.norm(n.simu, n.samp, mu, sig)
set.seed(181828); func.hist.norm(n.simu=100, n.samp=c(10, 40, 160), mu=4, sig=0.5)
set.seed(181828); func.hist.norm(n.simu=50, n.samp=c(10, 30, 90), mu=10, sig=1)
set.seed(181828); func.hist.norm(n.simu=50, n.samp=c(10, 20, 40, 80), mu=10, sig=1)
```

2 Independent Two Samples: Unequal Variances

1. An example for unequal variances: $\{Y_{1i}\}_{i=1}^{40} \sim iidN(10, 6^2)$ and $\{Y_{2j}\}_{j=1}^{10} \sim iidN(5, 3^2)$.

```
set.seed(18)
samp.norm1 <- rnorm(40, mean=10, sd=6)
samp.norm2 <- rnorm(10, mean=5, sd=3)
qqnorm(samp.norm1);    qqline(samp.norm1)
qqnorm(samp.norm2);    qqline(samp.norm2)

samp.norm    <- c(samp.norm1, samp.norm2)
```

```r
samp.norm12 <- as.factor(c(rep(1,length(samp.norm1)),rep(2,length(samp.norm2))
    ))
boxplot(samp.norm ~ samp.norm12)
var(samp.norm1); var(samp.norm2)
```

2. Levene's test

```r
#install.packages("car") # install "car" package
library(car) # load "car" package to do Levene's test
leveneTest(samp.norm, samp.norm12)

```

3. Welch's T test

```r
t.test(samp.norm1, samp.norm2, var.equal=TRUE)
t.test(samp.norm1, samp.norm2, var.equal=FALSE) # Welch T test

```

(a) Assumptions

i The first sample $\{Y_{1i}\}_{i=1}^{n_1}$ is a random sample of size n_1 from $N(\mu_1, \sigma_1^2)$.

ii The second sample $\{Y_{2j}\}_{j=1}^{n_2}$ is a random sample of size n_2 from $N(\mu_2, \sigma_2^2)$.

iii The two samples $\{Y_{1i}\}_{i=1}^{n_1}$and $\{Y_{2j}\}_{j=1}^{n_2}$ are independent.

iv The variances are not the same $\sigma_1^2 \neq \sigma_2^2$.

(b) To test $H_0 : \mu_1 = \mu_2$ vs $H_A : \mu_1 \neq \mu_2$, we test

$$H_0 : \mu_1 - \mu_2 = 0 \text{ vs } H_A : \mu_1 - \mu_2 \neq 0$$

(c) Test statistic

i $T = \frac{\bar{Y_1} - \bar{Y_2} - (\mu_1 - \mu_2)}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$

ii Under $H_0 : \mu_1 - \mu_2 = 0$,

$$T = \frac{\bar{Y_1} - \bar{Y_2}}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \sim T_{\text{adf}},$$

where df $= \frac{(r_1+r_2)^2}{\frac{r_1^2}{n_1-1} + \frac{r_2^2}{n_2-1}}$, where $r_1 = \frac{s_1^2}{n_1}$ and $r_1 = \frac{s_2^2}{n_2}$

(d) Testing result

i From the data and R output,

$\bar{y_1} = 9.487486 \quad s_1^2 = 46.04482 \quad n_1 = 40$ and

$\bar{y_2} = 6.103079 \quad s_2^2 = 9.853394 \quad n_2 = 10$

ii The observed test statistic is:

$$t = \frac{\bar{y_1} - \bar{y_2}}{\sqrt{s_1^2/n_1 + s_2^2/n_2}} = \qquad\qquad = 2.3154.$$

iii The adf are:

$$\begin{aligned} r_1 &= \frac{s_1^2}{n_1} = \qquad , \\ r_2 &= \frac{s_2^2}{n_2} = \qquad , \\ \text{adf} &= \frac{(r_1 + r_2)^2}{\frac{r_1^2}{n_1 - 1} + \frac{r_2^2}{n_2 - 1}} = \qquad\qquad = 32.177 \end{aligned}$$

iv $t_{\text{adf},\alpha/2} \simeq t_{32,0.025} = 2.037 < t = 2.3154.$
Thus, reject H_0 at the significance level $\alpha = 0.05$.

(e) Confidence Interval

i An approximate $(1 - \alpha)$ CI for $\mu_1 - \mu_2$ is

$$\bar{y_1} - \bar{y_2} \pm t_{\text{adf},\alpha/2} \times \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

ii Since $t_{\text{adf},\alpha/2} \simeq t_{32,0.025} = 2.037$, an approximate 95% CI for $\mu_1 - \mu_2$ is

$$(\qquad\qquad - \qquad\qquad) \pm 2.037 \times \sqrt{\left(\frac{\qquad}{}\right) + \left(\frac{\qquad}{}\right)}$$

which is [,   ].

iii Interpretation:
It has 95% confidence that $\mu_1 - \mu_2$ is between   and   .

3 Confidence Interval and Hypothesis Testing

4 Randomization Test

A source code for the randomization test (func.randTest.R) is in canvas.wisc.edu.

```
#setwd("~/HW02/") # set working directory
source("func.randTest.R")
```

```r
rand.test <- function(y1, y2, paired=NULL) {
nsims <- 10000

if (paired==FALSE || is.null(paired)) {
x  <- c(y1,y2)
gp <- as.factor(c(rep("a",length(y1)), rep("b",length(y2))))
labs <- unique(gp)
obsmeandiff <- mean(x[gp==labs[1]]) - mean(x[gp==labs[2]])
meandiffvec <- NULL
for( i in 1:nsims){
newgp <- sample(gp, size = length(gp))
newmeandiff <- mean(x[newgp==labs[1]]) - mean(x[newgp==labs[2]])
meandiffvec <- c(meandiffvec, newmeandiff)
}
pl <- sum(meandiffvec >= obsmeandiff)
pg <- sum(meandiffvec <= obsmeandiff)
pval <-  ifelse(pl < pg, 2 * pl/nsims, 2 * pg/nsims)

hhh <- hist(meandiffvec, plot=FALSE)
hist(meandiffvec, xlab="Differences of means",
main=paste("Independent Sample Randomization Test, p=", pval),
probability=TRUE, xlim=range(c(hhh$breaks, obsmeandiff)))
abline(v=obsmeandiff, col="red")
mtext(paste(round(obsmeandiff, digits=4)), side=1, at=obsmeandiff)
}

else {
d <- y1 - y2
ld <- length(d)
obsmeand <- mean(d)
meandvec <- NULL
for( i in 1:nsims){
rsigns <- 2 * rbinom(ld, 1, .5) - 1
newd <- rsigns * d
newmeand <- mean(newd)
meandvec <- c(meandvec, newmeand)
}
pl <- length(meandvec[meandvec >= obsmeand])
pg <- length(meandvec[meandvec <= obsmeand])
pval <-  ifelse(pl < pg, 2 * pl/nsims, 2 * pg/nsims)
hhh <- hist(meandvec, plot=FALSE)
hist(meandvec, xlab="Mean of differences",
main=paste("Paired Sample Randomization Test, p=", pval),
probability=TRUE, xlim=range(c(hhh$breaks, obsmeand)))
abline(v=obsmeand, col="red")
mtext(paste(round(obsmeand, digits=4)), side=1, at=obsmeand)
}
}
```

1. Independent Two Samples: "Weight Gain" data

```r
  dietA <- c(37.8, 27.5, 41.2, 26.5, 28.6)
  dietB <- c(12.3, 14.3, 19.2, 4.0, 25.9)
  set.seed(18); rand.test(dietA, dietB, paired=F)

```

2. Paired Sample: "Blood Pressure" data

```
before <- c(90, 100, 92, 96, 96, 96, 92, 98, 102, 94, 94, 102, 94, 88, 104)
after <- c(88, 92, 82, 90, 78, 86, 88, 72, 84, 102, 94, 70, 94, 92, 94)
set.seed(18); rand.test(before, after, paired = T)
```

5 Wilcoxon Test

1. Independent Two Samples: "Weight Gain" data

```
wilcox.test(dietA, dietB)
wilcox.test(dietA, dietB, exact=TRUE)
wilcox.test(dietA, dietB, exact=FALSE)
```

2. Paired Sample: "Blood Pressure" data

```
wilcox.test(before, after, paired=TRUE)
wilcox.test(before, after, paired=TRUE, exact=TRUE)
wilcox.test(before, after, paired=TRUE, exact=FALSE)
```