Statistical Methods for Data Science
CS 6313.001: Mini Project #5

Due on Thursday April 18, 2019 at 10am

Instructor: Prof. Min Chen

Shyam Patharla **(sxp178231)**

Contents

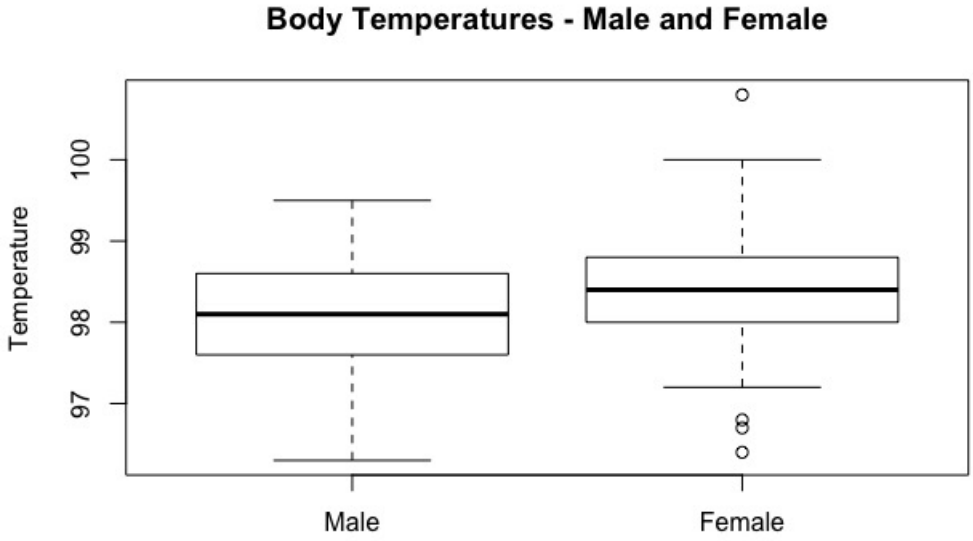
Section 1 Answers

Problem 1.1

(a) Do males and females differ in body temperature?

Let us first formulate the hypotheses:
Null Hypothesis, $H_0 : \mu_x = \mu_y$ (male and female body temperatures are same)
Alternate Hypothesis, $H_1 : \mu_x \neq \mu_y$ (male and female body temperatures are different)

Let us make a boxplot first.



Inferences:

1. We see that the two distributions do **differ** in some respects.
2. The female distribution has a higher **median** body temperature.
3. The female distribution has some **outliers** on both ends.
4. The male distribution has a slightly **wider** interquartile range.

We compute a 95% CI for two cases:

1. Assuming equal variances: [**-0.53963938 -0.03882216**]
2. Assuming unequal variances: [**-0.53964856 -0.03881298**]

Both approaches yield **similar** confidence intervals for the difference in means. These confidence intervals suggest that the difference in means is **significant**, supporting H_1.

We finally confirm our conclusions with a **T-test**. Suppose:
x.mean= mean body temperature for males
x.sd= standard deviation for body temperature for males
n=number of males
y.mean= mean body temperature for females
y.sd= standard deviation for body temperature for females
m=number of females

```
> t_obs <- (x.mean-y.mean) / sqrt((x.sd^2/n) + (y.sd^2/m))
> p_value <- 2*(1 - pt(abs(t_obs),df))
> p_value
# [1] 0.02393826
```

Assuming a 5% level of significance,

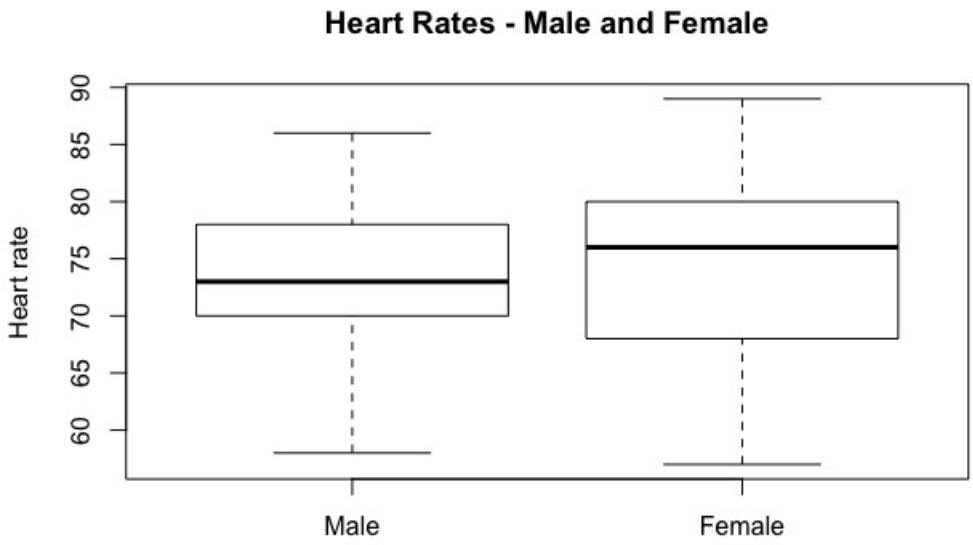
$$p - value = 0.02393826 < 0.05 \tag{1}$$

Hence, we **reject** H_0. Hence males and females differ in body temperature.

(b) Do males and females differ in heart rate?

Let us first formulate the hypotheses:
Null Hypothesis, $H_0 : \mu_x = \mu_y$ (male and female heart rates are same)
Alternate Hypothesis, $H_1 : \mu_x \neq \mu_y$ (male and female heart rates are different)

Let us make a boxplot first.



Inferences:

1. We see that the two distributions do **differ** in some respects.
2. The female distribution has a higher **median** heart rate.
3. The female distribution has a slightly **wider** interquartile range.

We compute a 95% CI for two cases:

1. Assuming equal variances: [**-3.241461, 1.672230**]
2. Assuming unequal variances: [**-3.243732, 1.674501**]

Both approaches yield **similar** confidence intervals for the difference in means. These confidence intervals indicate that the differences in means **could** be zero, but we don't have enough information to make our decision.

Hence, we proceed with a **T-test**. Suppose:
x.mean= mean heart rate for males
x.sd= standard deviation for heart rate for males
n=number of males
y.mean= mean heart rate for females
y.sd= standard deviation for heart rate for females
m=number of females

```
> t_obs <- (x.mean-y.mean) / sqrt((x.sd^2/n) + (y.sd^2/m))
> p_value <- 2*(1 - pt(abs(t_obs),df))
> p_value
# [1] 0.5286842
```

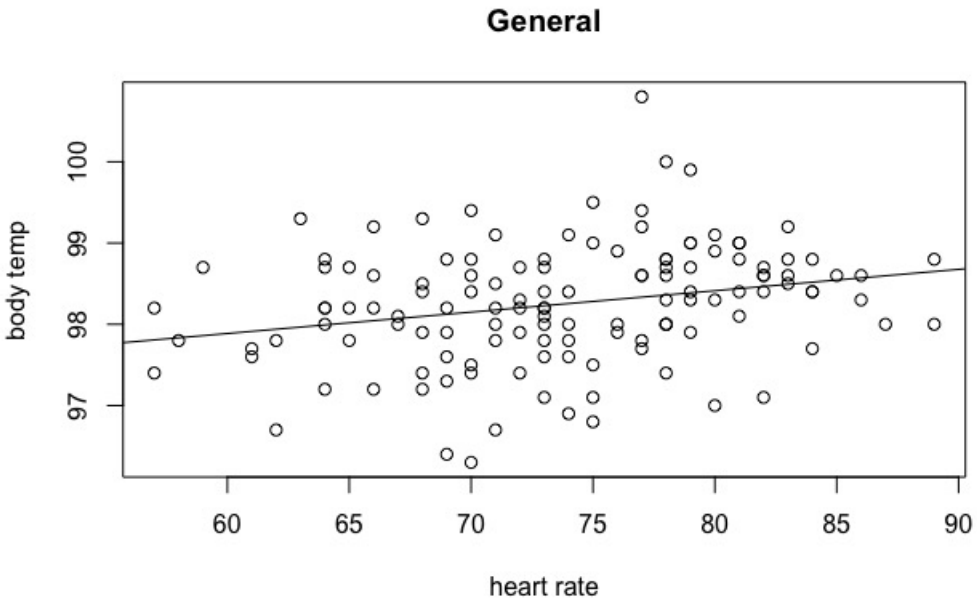
Assuming a 5% level of significance,

$$p - value = 0.5286842 > 0.05 \tag{2}$$

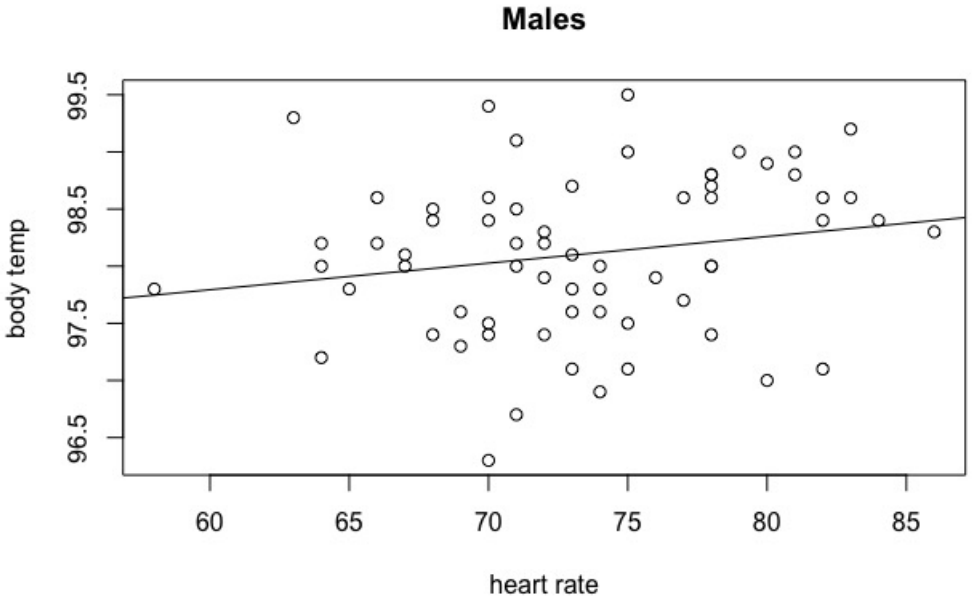
Hence, we **accept** H_0. Hence males and females do not differ in heart rate.

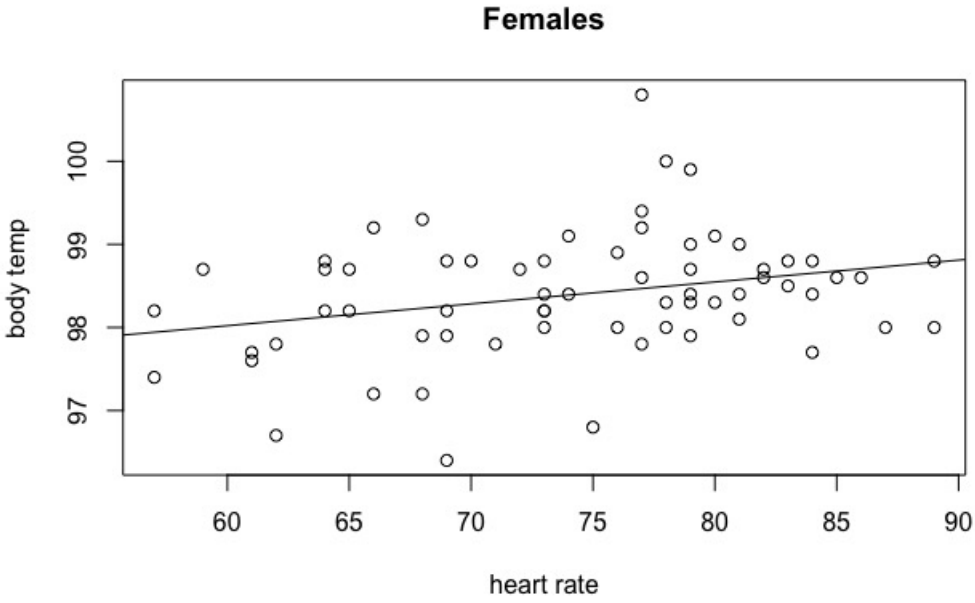
(c) Is there a linear relationship between body temperature and heart rate? Does the relationship depend on temperature?

Let us first make a scatterplot of body temperature against heart rate for all genders.



Let us now make scatterplots of body temperature against heart rate for males and females.





1. The correlation coefficient is $\rho = 0.2536564$ for the whole population. The data does not fit the line well.

2. The correlation coefficient is $\rho = 0.1955894$ for the male population. The data does not fit the line well.

3. The correlation coefficient is $\rho = 0.2869312$ for the female population, higher than the ρ for the male and whole population. The data does fit the line well compared to the

male distribution.

4. We can conclude that:

   (a) there is **no** linear relationship between heart rate and body temperature for the **whole** population.

   (b) there is **no** linear relationship between heart rate and body temperature for **males**.

   (c) there **might** be a linear relationship between heart rate and body temperature for **females** but the strength of the linear relationship is not very high.

Problem 1.2

(a) For a given setting of (n,lambda), compute Monte Carlo estimates for coverage probabilities of the 2 individuals

Let n=30, λ=0.1, nsim=5000. We get:

```
> cover.probs(30,0.1,5000)
# [1]  0.9136 0.9360
```

The coverage probability for both the large-sample and bootstrap intervals is slightly **less** than our confidence level of 95%.

(b) Repeat (a) for remaining combinations of (n,lambda) and present a summary of the results

	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
n=5	0.8098	0.8204	0.7974	0.8124
n=10	0.8652	0.8696	0.8732	0.8748
n=30	0.9152	0.9136	0.9166	0.915
n=100	0.9412	0.9368	0.9434	0.9422

Table 1: Coverage probabilities for large-sample interval

	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
n=5	0.8968	0.8970	0.8984	0.9006
n=10	0.9222	0.9152	0.920	0.9156
n=30	0.9366	0.9360	0.9366	0.9434
n=100	0.9442	0.9420	0.9476	0.9430

Table 2: Coverage probabilities for bootstrap interval

(c) Interpret the results

Inferences for both intervals:

1. For a given value of n, the coverage probability **does not change** much with respect to λ.
2. For a given value of λ, the coverage probability **increases** with increasing n.
3. The coverage probabilities for n=100 are **closest** to our confidence level of 95% irrespective of λ.

Sub-questions:

1. For the large sample interval, how large does n have to be for the interval to be accurate?
   The large sample interval gives coverage probabilities close to our confidence level of 95% for n=100.
2. For the bootstrap interval, how large does n have to be for the interval to be accurate?
   The bootstrap interval gives coverage probabilities close to 0.95 for for n=100.
3. Do these answers depend on λ?
   No, the answers to the above questions **do not** depend on λ.
4. Can we say that one method is more accurate than the other? Which interval would you recommend?
   The bootstrap interval appears to be working **better** for smaller n. But for large n, both are equally good irrespective of λ.
   I would recommend the **bootstrap** interval.

(d) Do the answers to (c) depend on the specific values of lambda that were fixed in advance?

No, the conclusions in (c) **do not** depend on λ.

Section 2 R Code

```
#########################
# R Code for Question 1(A)
#########################

# import sql library
library(sqldf)

# Reading the vapor.csv file
```

```
data <- read.csv.sql(file="/users/psprao/downloads/stats/datasets/bodytemp-
    heartrate.csv")

# Getting the male and female body temperatures
male.temp<-sqldf("select * from data where gender=1")[,1]
female.temp<-sqldf("select * from data where gender=2")[,1]

#Boxplot
par(mfrow=c(1,1))
boxplot(male.temp,female.temp,
        range=1.5,main="Body Temperatures - Male and Female",
        ylab="Temperature",names = c("Male", "Female"))



# Getting sample statistics
n=NROW(male.temp)
x.mean=mean(male.temp)
x.var=var(male.temp)
x.sd=sd(male.temp)

y.sd=sd(female.temp)
y.mean=mean(female.temp)
y.var=var(female.temp)
y=NROW(female.temp)

# Estimator for the difference in means
mean.diff.estimator<-x.mean - y.mean


# Getting a 95% CI for difference in means (assuming equal variances)
pooled.var<-((n-1)*x.var+(m-1)*y.var)/(n+m-2)
mean.diff.ci<- mean.diff.estimator+c(-1,1)*qt(1-(1-0.95)/2,n+m-2)*sqrt(pooled.
    var/m+pooled.var/n)
print(mean.diff.ci)
# [1] -0.53963938 -0.03882216


# Getting a 95% CI for difference in means (assuming unequal variances)
# getting degrees of freedom using Satherwaite's approximation
a<-(x.sd^2/n + y.sd^2/m)^2
b<-(x.sd^4)/((n^2)*(n-1))
c<-(y.sd^4)/((m^2)*(m-1))
df<-a/(b+c)
mean.diff.ci<- mean.diff.estimator+c(-1,1)*qt(1-(1-0.95)/2,df)*sqrt(x.sd^2/n+y
    .sd^2/m)

print(mean.diff.ci)
# [1] -0.53964856 -0.03881298
```

```
# T-test for a 2 sided hypothesis
# Null hypothesis: body temperatures of male and females do not differ.
# Alternate hypothesis: body temperatures of males and females differ
# tobs value
t_obs<-(x.mean-y.mean)/sqrt((x.sd^2/n) + (y.sd^2/m))

# p-value for a 2-sided test
p_value<-2*(1 - pt(abs(t_obs),df))
# > p_value
# 0.02393826
```

```
#########################
# R Code for Question 1(B)
#########################

# import sql library
library(sqldf)

# Reading the vapor.csv file
data <- read.csv.sql(file="/users/psprao/downloads/stats/datasets/bodytemp-
   heartrate.csv")

# Getting the male and female heart rate values
male.hrate<-sqldf("select * from data where gender=1")[,3]
female.hrate<-sqldf("select * from data where gender=2")[,3]

# Boxplot
par(mfrow=c(1,1))
boxplot(male.hrate,female.hrate,
        range=1.5,main="Heart Rates - Male and Female",
        ylab="Heart rate",names = c("Male", "Female"))


# Getting sample statistics
n<-NROW(male.hrate)
x.mean<-mean(male.hrate)
x.var<-var(male.hrate)
x.sd<-sd(male.hrate)

y.mean<-mean(female.hrate)
y.var<-var(female.hrate)
m<-NROW(female.hrate)
y.sd<-sd(female.hrate)


# Estimator for the difference in means
mean.diff.estimator<-x.mean - y.mean
```

```
# Getting a 95% CI for difference in means (assuming equal variances)
pooled.var<-((n-1)*x.var+(m-1)*y.var)/(n+m-2)
mean.diff.ci<- mean.diff.estimator+c(-1,1)*qt(1-(1-0.95)/2,n+m-2)*sqrt(pooled.
    var/m+pooled.var/n)
# > mean.diff.ci
# [1] -3.241461  1.672230


# Getting a 95% CI for difference in means (assuming unequal variances)
# getting degrees of freedom using Satherwaite's approximation
a<-(x.sd^2/n + y.sd^2/m)^2
b<-(x.sd^4)/((n^2)*(n-1))
c<-(y.sd^4)/((m^2)*(m-1))
df<-a/(b+c)

mean.diff.ci<- mean.diff.estimator+c(-1,1)*qt(1-(1-0.95)/2,df)*sqrt(x.sd^2/n+y
    .sd^2/m)

# > mean.diff.ci
# [1] -3.243732  1.674501


# T-test for a 2 sided hypothesis
# Null hypothesis: heart rates of male and females do not differ.
# Alternate hypothesis: heart rates  of males and females differ
# t_obs value
t_obs<-(x.mean-y.mean)/sqrt((x.sd^2/n) + (y.sd^2/m))

# p-value for a 2-sided test
p_value<-2*(1 - pt(abs(t_obs),df))
# > p_value
# 0.5286842
```

```
#########################
# R Code for Question 1(C)
#########################

# import sql library
library(sqldf)


# Reading the bodytem-heartrate.csv file
data <- read.csv.sql(file="/users/psprao/downloads/stats/datasets/bodytemp-
    heartrate.csv")


# Scatterplot for heart rate vs body temp for whole sample
```

```
hrate<-data[,3]
temp<-data[,1]
par(mfrow=c(1,1))
plot(hrate,temp,xlab="heart rate",ylab="body temp",main="Scatterplot: body
     temperature vs heart rate")
abline(lm(temp~hrate))


# correlation coefficient for heart rate and body temperature
# > cor(hrate,temp)
# [1] 0.2536564



# Scatterplot for body temp vs heart rate for males
male.hrate<-sqldf("select * from data where gender=1")[,3]
male.temp<-sqldf("select * from data where gender=1")[,1]
plot(male.hrate,male.temp,xlab="heart rate",ylab="body temp",main="Scatterplot
    : body
     temperature vs heart rate - males")
abline(lm(male.temp~male.hrate))



# correlation coefficient for heart rate and body temperature males
# > cor(male.hrate,male.temp)
# [1]  0.1955894


# Scatterplot for body temp vs heart rate for females in sample
female.hrate<-sqldf("select * from data where gender=2")[,3]
female.temp<-sqldf("select * from data where gender=2")[,1]
plot(female.hrate,female.temp,xlab="heart rate",ylab="body temp",main="
    Scatterplot: body
     temperature vs heart rate - females")
abline(lm(female.temp~female.hrate))


# correlation coefficient for heart rate and body temperature (females)
# > cor(hrate,temp)
# [1]  0.2869312
```

```
##############################
# R code for question 2
##############################

# function to get large -sample CI
ci.norm<-function(n,lambda)
{
```

```
  alpha<-0.05

  # generate a sample
  x <- rexp(n,lambda)

  # get a (1-alpha)% CI
  ci <- mean(x) + c(-1,1) * qnorm(1-alpha/2) * sd(x)/sqrt(n)

  return(ci)
}


# function to compute one resample and its mean
mean.star<-function(x)
{
  n<-length(x)

  # getting lambda value
  xbar<-mean(x)
  lambda<-1/xbar

  # resample
  xstar<-rexp(n,rate=lambda)

  # compute mean of resample
  xstar.mean<-mean(xstar)

  return(xstar.mean)
}



# Function to get CI using parametric bootstrap
par.boot.ci<-function(n,lambda)
{
  # generate a sample
  x<-rexp(n,lambda)

  # Generate nboot resamples
  nboot<-1000
  mean.boot.dist<-replicate(nboot,mean.star(x))

  # get a 95% percentile bootstrap CI
  mean.ci<-sort(mean.boot.dist)[c(25, 975)]

  return(mean.ci)
}
```

```
# function to get large-sample and boostrap  coverage probabilty for given (n,
   lambda)
cover.probs<-function(n,lambda,nsim)
{
  # value of mean
  mu<-1/lambda

  # Generate nsim large-sample CIs
  ci.mat<-replicate(nsim,ci.norm(n,lambda))

  # get large-sample coverage probability
  cp.norm<-mean((mu>=ci.mat[1,])*(mu<=ci.mat[2,]))

  # generate nsim bootstrap CIs
  ci.mat<-replicate(nsim,par.boot.ci(n,lambda))

  # get boostrap interval coverage probability
  cp.boot<-mean((mu>=ci.mat[1,])*(mu<=ci.mat[2,]))

  return(c(cp.norm,cp.boot))

}

# Confidence intervals computed using using large-sample and bootstrap
# first value in result is the large-sample coverage probability
# second value is the percentile bootstrap coverage probability

# > cover.probs(30,0.1,5000)
# [1] 0.9136 0.9360

# > cover.probs(5,0.01,5000)
# [1] 0.8098 0.8968

# > cover.probs(5,0.1,5000)
# [1] 0.8204 0.8970

# > cover.probs(5,1,5000)
# [1] 0.7974 0.8984

# > cover.probs(5,10,5000)
# [1] 0.8124 0.9006

#> cover.probs(10,0.01,5000)
#[1] 0.8652 0.9222

# > cover.probs(10,0.1,5000)
# [1] 0.8696 0.9152
```

```
# > cover.probs(10,1,5000)
# [1] 0.8732 0.920

# > cover.probs(10,10,5000)
# [1] 0.8748 0.9156

# > cover.probs(30,0.01,5000)
# [1] 0.9152 0.9366

# > cover.probs(30,1,5000)
# [1] 0.9166 0.9366

# > cover.probs(30,10,5000)
# [1] 0.915 0.9434

# > cover.probs(100,0.01,5000)
# [1] 0.9412 0.9442

# > cover.probs(100,0.1,5000)
# [1] 0.9368 0.9420

# > cover.probs(100,1,5000)
# [1]  0.9434 0.9476

# > cover.probs(100,10,5000)
# [1] 0.9422 0.9430
```