Final Project

Huiyue Li, Jiajie Shen, Weijia Mai<- Huiyue Li

```
# load the packages we need to use
library(knitr)
library(here)
library(Hmisc)
library(tidyverse)
```

Part 1

A function that produces the simulated data, estimates the coverage probabilities, and estimates the width of the intervals

```
# remove all the objects in workspace
rm(list=ls())
# simulation function
sim.fc=function(seed,no.runs,n.samps,theta){
  CI1.low=matrix(NA,no.runs,length(n.samps)) # set empty matrices
  CI1.up=matrix(NA,no.runs,length(n.samps))
  CI2.low=matrix(NA,no.runs,length(n.samps))
  CI2.up=matrix(NA,no.runs,length(n.samps))
  CI3.low=matrix(NA,no.runs,length(n.samps))
  CI3.up=matrix(NA,no.runs,length(n.samps))
  CP=matrix(NA,3,length(n.samps))
  width=matrix(NA,3,length(n.samps))
  col.names = rep(NA,length(n.samps))   # set empty vector
  row.names=c("Asymptotic Interval","Wilson Interval","Exact Binomial I
nterval")
  set.seed(seed)  # set seed

# create a FOR loop to get the CIs, then gain CP
  for (i in 1:length(n.samps)) {
    col.names[i] <- paste('Sample Size= ',n.samps[i],sep='')

    samp=rbinom(no.runs,n.samps[i],theta)
    CI1.low[,i]=binconf(samp,n.samps[i],.05,method = "asymptotic")[,2]
    CI1.up[,i]=binconf(samp,n.samps[i],.05,method = "asymptotic")[,3]
    CI2.low[,i]=binconf(samp,n.samps[i],.05,method = "wilson")[,2]
    CI2.up[,i]=binconf(samp,n.samps[i],.05,method = "wilson")[,3]
    CI3.low[,i]=binconf(samp,n.samps[i],.05,method = "exact")[,2]
    CI3.up[,i]=binconf(samp,n.samps[i],.05,method = "exact")[,3]

    CP[1,i]=sum(CI1.low[,i]<=theta & CI1.up[,i]>=theta)/no.runs
    CP[2,i]=sum(CI2.low[,i]<=theta & CI2.up[,i]>=theta)/no.runs
```

```r
    CP[3,i]=sum(CI3.low[,i]<=theta & CI3.up[,i]>=theta)/no.runs

  }

# set the colname and rowname
 colnames(CP)=col.names
 colnames(width)=col.names
 rownames(CP)=row.names
 rownames(width)=row.names

# using apply to calculate the CI width
 width[1,]=apply(CI1.up-CI1.low, 2, mean)
 width[2,]=apply(CI2.up-CI2.low, 2, mean)
 width[3,]=apply(CI3.up-CI3.low, 2, mean)

 return(list(CP=CP,width=width))
}
# generate seeds
set.seed(601)
seeds<-round(runif(2,100,1000),0)

# set simulation parameters that will not change across
n.samps<-c(20,40,100)
no.runs<-20000

# simulation scenario 1: theta=0.05
fit.theta05<-sim.fc(seeds[1],no.runs,n.samps,0.05)
# simulation scenario 1: theta=0.15
fit.theta15<-sim.fc(seeds[2],no.runs,n.samps,0.15)
```

Tables that summarize the results (one for coverage probability and one for width)

```
# create  multiple tables via kable(list()) for CP
kable(list(cbind(rep(0.05,3),round(fit.theta05$CP,2)),
           cbind(rep(0.15,3),round(fit.theta15$CP,2))),
           align  = "c", caption="Table 1: Coverage Probability for Dif
ferent 95% Confidence Intervals",col.name = c("**Theta**","**Sample Siz
e=20**","**Sample Size=40**","**Sample Size=100**"))
```

Table 1: Coverage Probability for Different 95% Confidence Intervals

	Theta	Sample Size=20	Sample Size=40	Sample Size=100
Asymptotic Interval	0.05	0.64	0.86	0.88
Wilson Interval	0.05	0.92	0.95	0.97
Exact Binomial Interval	0.05	0.98	0.99	0.98
	Theta	**Sample Size=20**	**Sample Size=40**	**Sample Size=100**
Asymptotic Interval	0.15	0.82	0.94	0.93
Wilson Interval	0.15	0.98	0.96	0.93
Exact Binomial Interval	0.15	0.98	0.98	0.96

```
# create multiple tables for width
kable(list(cbind(data.frame(rep(0.05,3)),round(fit.theta05$width,2)),
           cbind(data.frame(rep(0.15,3)),round(fit.theta15$width,2))),
           align  = "c", caption="Table 2: Width for Different 95% Confidence Intervals",col.name = c("**Theta**","**Sample Size=20**","**Sample Size=40**","**Sample Size=100**"))
```

Table 2: Width for Different 95% Confidence Intervals

	Theta	Sample Size=20	Sample Size=40	Sample Size=100
Asymptotic Interval	0.05	0.15	0.12	0.08
Wilson Interval	0.05	0.22	0.15	0.09
Exact Binomial Interval	0.05	0.24	0.16	0.09

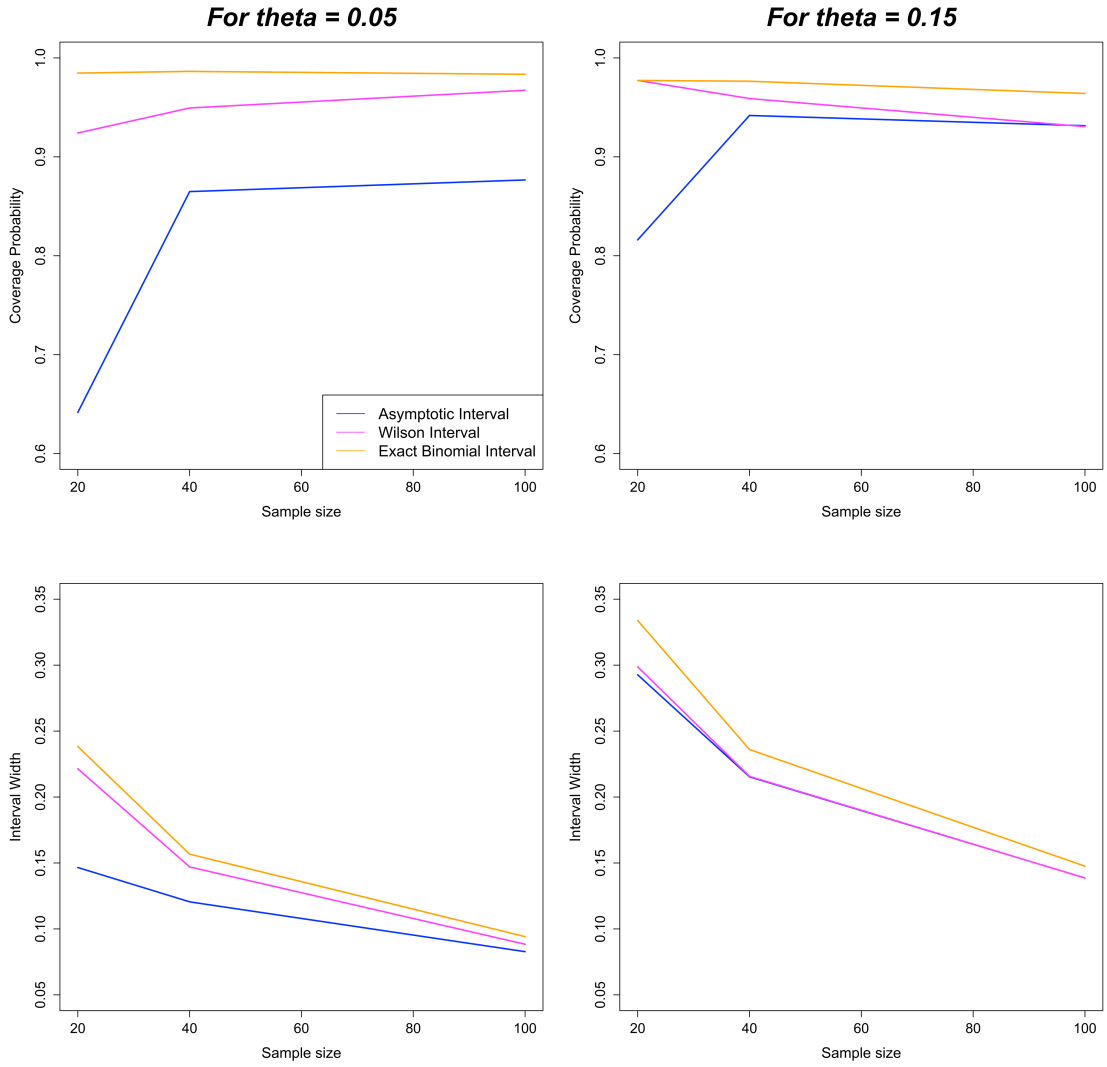
	Theta	Sample Size=20	Sample Size=40	Sample Size=100
Asymptotic Interval	0.15	0.29	0.22	0.14
Wilson Interval	0.15	0.30	0.22	0.14
Exact Binomial Interval	0.15	0.33	0.24	0.15

Plots that show the measures of performance by the varying conditions

```
# plot function
plot.func=function(n.samps,t1.data,t2.data,t3.data,
                       main.title,y.lab,y.lim=NULL,
                       legend=FALSE){
  if (is.null(y.lim)) {y.lim <- range(t1.data,t2.data,t3.data)}

  plot(n.samps,t1.data,
       ylim=y.lim,
       ylab=y.lab, main=main.title,
       xlab='Sample size',
       type='l',col='blue',lwd=2)
  lines(n.samps,t2.data,col='magenta',lwd=2)
  lines(n.samps,t3.data,col='orange',lwd=2)
  if(legend==TRUE) {
    legend("bottomright",
           legend=c("Asymptotic Interval","Wilson Interval","Exact Binomial Interval"),
           lty=rep(1,3),
           col=c('blue','magenta','orange'),cex=1.5)}
  }

# plot results
par(mfcol=c(2,2),oma=c(0,0,3,0),cex.lab=1.42,cex.axis=1.35,cex.main=2.5,
```

```
font.main=4)

# CP for theta=0.05
plot.func(n.samps<-c(20,40,100),
          t1.data=fit.theta05$CP[1,],
          t2.data=fit.theta05$CP[2,],
          t3.data=fit.theta05$CP[3,],
          main.title="For theta = 0.05",
          y.lab="Coverage Probability",
          y.lim=c(0.60,1.00),
          legend=TRUE)

# width for theta=0.05
plot.func(n.samps<-c(20,40,100),
          t1.data=fit.theta05$width[1,],
          t2.data=fit.theta05$width[2,],
          t3.data=fit.theta05$width[3,],
          main.title="",
          y.lab="Interval Width",
          y.lim=c(0.05,0.35))

# CP for theta=0.15
plot.func(n.samps<-c(20,40,100),
          t1.data=fit.theta15$CP[1,],
          t2.data=fit.theta15$CP[2,],
          t3.data=fit.theta15$CP[3,],
          main.title = "For theta = 0.15",
          y.lab="Coverage Probability",
          y.lim=c(0.60,1.00))

# width for theta=0.15
plot.func(n.samps<-c(20,40,100),
          t1.data=fit.theta15$width[1,],
          t2.data=fit.theta15$width[2,],
          t3.data=fit.theta15$width[3,],
          main.title="",
          y.lab="Interval Width",
          y.lim=c(0.05,0.35))

# add the title for all these figures
mtext("Figure 1: The Measures of Performance by the Varying Conditions",
outer = T,cex=2.7,font = 2)
```

Figure 1: The Measures of Performance by the Varying Conditions



Part 2

Based on the simulation results from Part 1, I would like to use the **Wilson** interval estimation method. For the following reasons:

(1) Concerning the coverage probability, we could find that the coverage probability of asymptotic interval method is much lower than which of Wilson interval and exact binomial interval methods given that the true mortality percentages are *0.05* and *0.15*, especially when the sample size is small. Therefore, we do not use the asymptotic interval method.

(2) Concerning the CI width, we could find that the Wilson interval and exact binomial interval have relatively similar coverage probabilities given that the range of coverage probabilities of the three methods with a fixed sample size.

Whereas, the CI widths of Wilson interval and exact binomial interval have more differences given that the range of CI width of the three methods with a fixed sample size, especially when the sample size is small. And the CI width of Wilson interval estimation method is smaller than which of exact binomial interval in all cases. Thus, we would like to select Wilson interval estimation method based on the coverage probability and CI width.

Part 3

```
# Read in the data
mort=read_csv(here::here("mortality_data.csv"))

# Review the data and deal with any missing values or cleaning issues
glimpse(mort,width = 60) # review

## Rows: 62
## Columns: 3
## $ id    <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,…
## $ death <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ sex   <chr> "F", "F", "F", "F", "M", "M", "M", "F", "F…

mort=mort%>%mutate(sex=na_if(sex,-999)) # handle NA

# Compute a confidence interval around the proportion of deaths in the
full sample
CI=binconf(sum(mort$death),length(mort$id),0.05,method = "wilson")

# Compute confidence intervals around the proportion of deaths in your
full sample for each sex group
CI_male=binconf(mort%>%filter(sex=="M"&death==1)%>%nrow,mort%>%filter(s
ex=="M")%>%nrow,0.05,method = "wilson") # for male
CI_female=binconf(mort%>%filter(sex=="F"&death==1)%>%nrow,mort%>%filter
(sex=="F")%>%nrow,0.05,method = "wilson") # for female

# Create a table showing the results (Note: One patient is missing for
the sex variable,full sample not equals to male+female here)
table=cbind(c(length(mort$id),mort%>%filter(sex=="M")%>%nrow,mort%>%fil
ter(sex=="F")%>%nrow),rbind(CI,CI_male,CI_female))

rownames(table)=c("Full Sample","Male Group","Female Group")
colnames(table)=c("**Sample Size**","**Point Estimation**","**Lower Bou
nd**","**Upper Bound**")
```

```
kable(table,digits=2,caption ="Table 3: 95% Confidence Intervals around
 the Proportion of Deaths",align="c")
```

Table 3: 95% Confidence Intervals around the Proportion of Deaths

	Sample Size	Point Estimation	Lower Bound	Upper Bound
Full Sample	62	0.06	0.03	0.15
Male Group	25	0.04	0.00	0.20
Female Group	36	0.08	0.03	0.22