STA610 HW01

Hun Kang

2024-09-14

3.

a.

$$
\begin{aligned}
y_{ij} &= \mu + a_j + \epsilon_{ij},\ i \in [n],\ j \in [m] \\
a_j &\sim N(0, \tau^2) \\
\epsilon_{ij} &\sim N(0, \sigma^2)
\end{aligned}
$$

It is straightforward to verify that $V(y_{ij}) = \sigma^2 + \tau^2$, $V(\bar{y}_j) = \sigma^2/n + \tau^2$, and $V(\bar{y}) = \sigma^2/(nm) + \tau^2/m$. Since $\bar{y} \sim N(\mu, \sigma^2/(nm) + \tau^2/m)$, the width of the interval is

$$
4 \times SD(\bar{y}) = 4\sqrt{\sigma^2/(nm) + \tau^2/m}
$$

b.

$$
4\sqrt{\sigma^2/(nm) + \tau^2/m} < 1/2
$$

after rearranging, it becomes

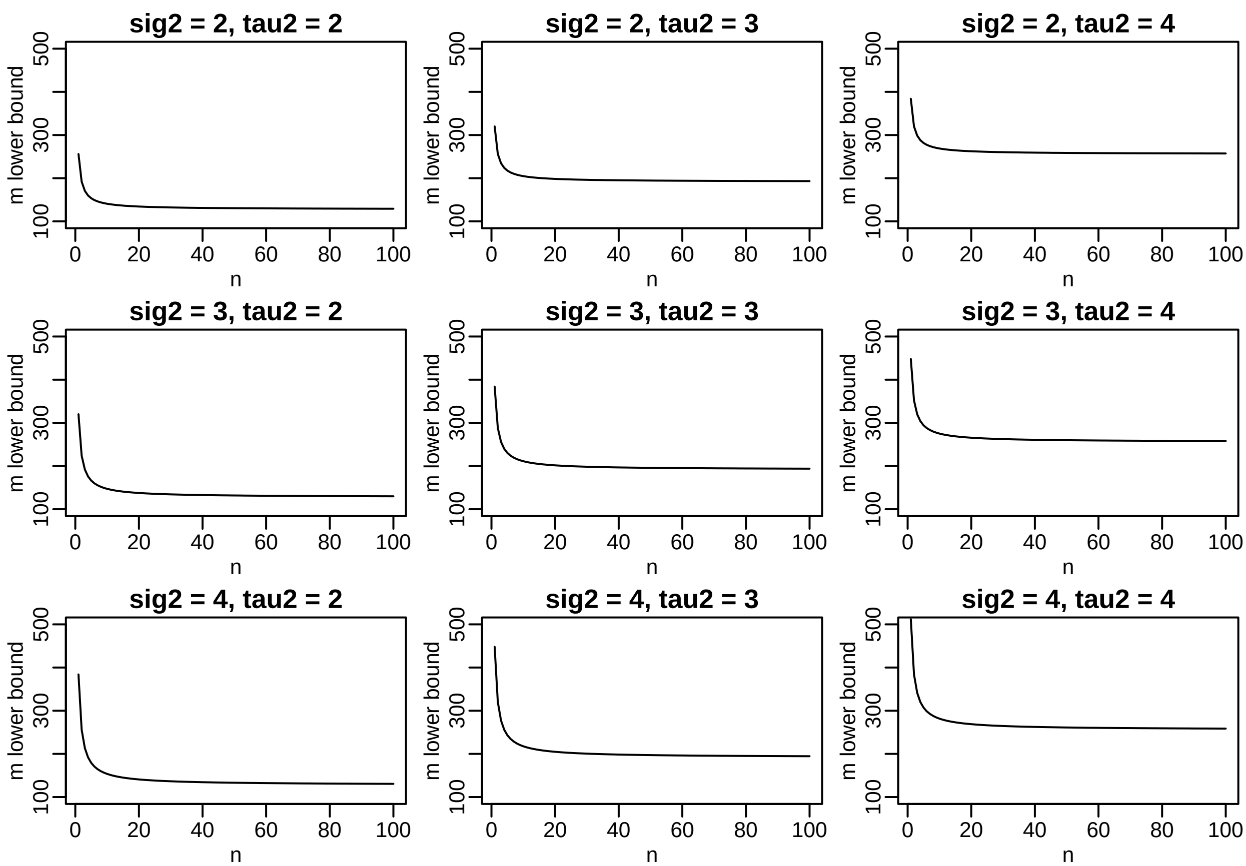
$$
m > 64(\sigma^2/n + \tau^2)
$$

c.

The lower bound of m is more affected by τ^2 than σ^2: more samples are needed for each group for larger τ^2.

```r
lb = function(n, sig2, tau2){
  64*(sig2/n + tau2)
}
n = 1:1e2
params = list(
  c(2,2),
  c(2,3),
  c(2,4),
  c(3,2),
  c(3,3),
  c(3,4),
  c(4,2),
  c(4,3),
  c(4,4)
)
opars = par(no.readonly = T)
```

```r
par(mfrow=c(3,3))
par(mar=c(2.5, 2.5, 1.5, 0.5), mgp=c(1.5,0.5,0), oma = c(0,0,0,0))
for(i in seq_along(params)){
  param = params[[i]]
  plot(n, lb(n, param[1], param[2]), type="l", ylim = c(100,500),
       xlab = "n", ylab = "m lower bound",
       main = paste0("sig2 = ", param[1], ", tau2 = ",param[2]))
}
```




```r
par(opars)
```

4.

```r
nels_math_ses = dget("https://www2.stat.duke.edu/~pdh10/Teaching/610/Homework/nels_math_ses")
head(nels_math_ses)
```

```
##   school mathdeg mathscore   ses
## 1   1011       4     52.11 -0.25
## 2   1011       4     57.65  0.58
## 3   1011       6     66.44 -0.85
## 4   1011       4     44.68 -0.80
## 5   1011       6     40.57 -1.41
## 6   1011       4     35.04 -1.07
```

a.

Since the sample size of each school is all different, the grand mean is NOT the sample mean of group means. Therefore, sample variance of the group means is NOT the variance of the group means around the grand mean.

```
# grand mean
ybar = mean(nels_math_ses$mathscore)
ybar
```

```
## [1] 48.07446
```

```
# variance of the group means around the grand mean
ybars = aggregate(mathscore ~ as.factor(school), data = nels_math_ses, mean)[,2]
var(ybars) # wrong
```

```
## [1] 30.99446
```

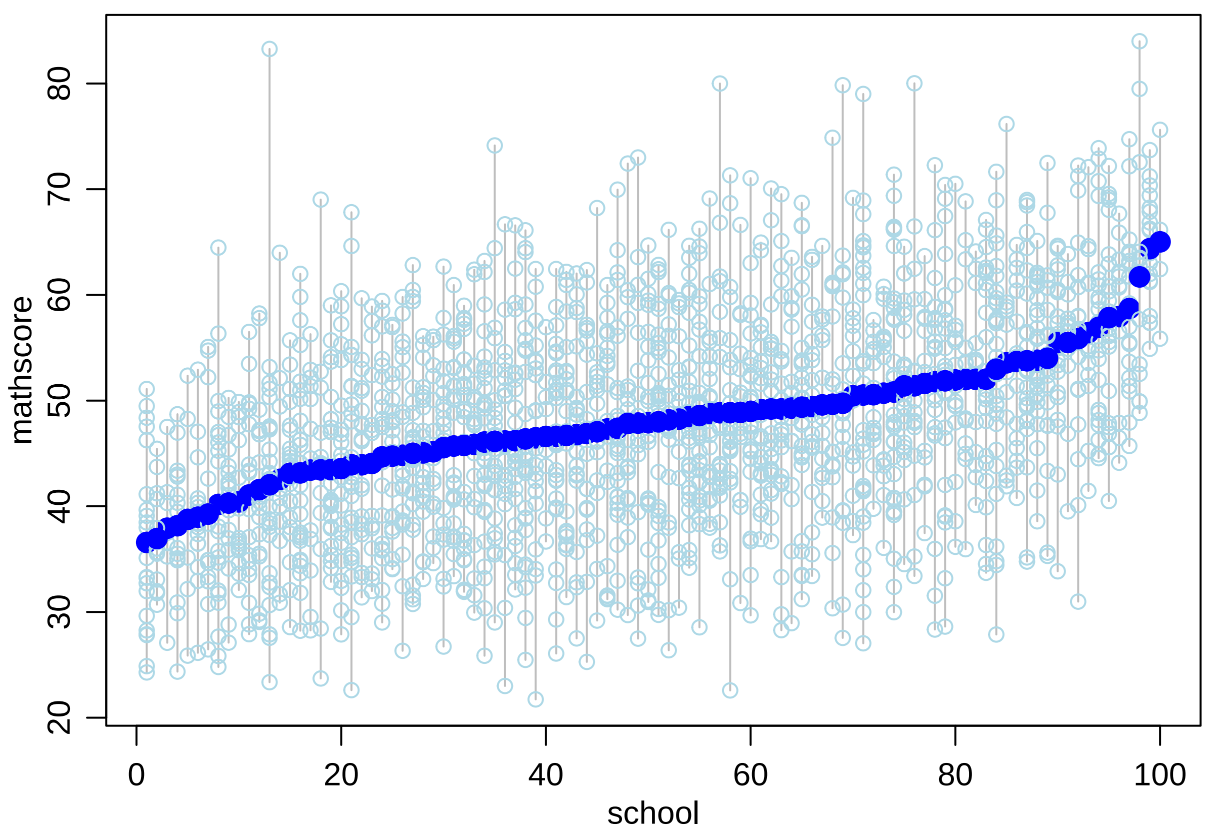
```
sum((ybars - ybar)^2 / (length(ybars)-1)) # correct
```

```
## [1] 30.99751
```

```
gdotplot<-function(y,g,xlab="group",ylab="response",mcol="blue",
                   ocol="lightblue",sortgroups=TRUE,...)
{
  m<-length(unique(g))
  rg<-rank( tapply(y,g,mean),ties.method="first")
  if(sortgroups==FALSE){ rg<-1:m ; names(rg)<-unique(g)}
  plot(c(1,m),range(y),type="n",xlab=xlab,ylab=ylab)

  for(j in unique(g))
  {
    yj<-y[g==j]
    rj<-rg[ match(as.character(j),names(rg)) ]
    nj<-length(yj)
    segments(rj ,max(yj),rj,min(yj),col="gray")
    points( rep(rj,nj), yj,col=ocol, ...)
    points(rj,mean(yj),pch=16,cex=1.5,col=mcol)
  }
}

par(mar=c(3,3,1,1), mgp=c(1.75,.75,0))
gdotplot(nels_math_ses$mathscore,
         nels_math_ses$school,
         xlab="school", ylab="mathscore")
```



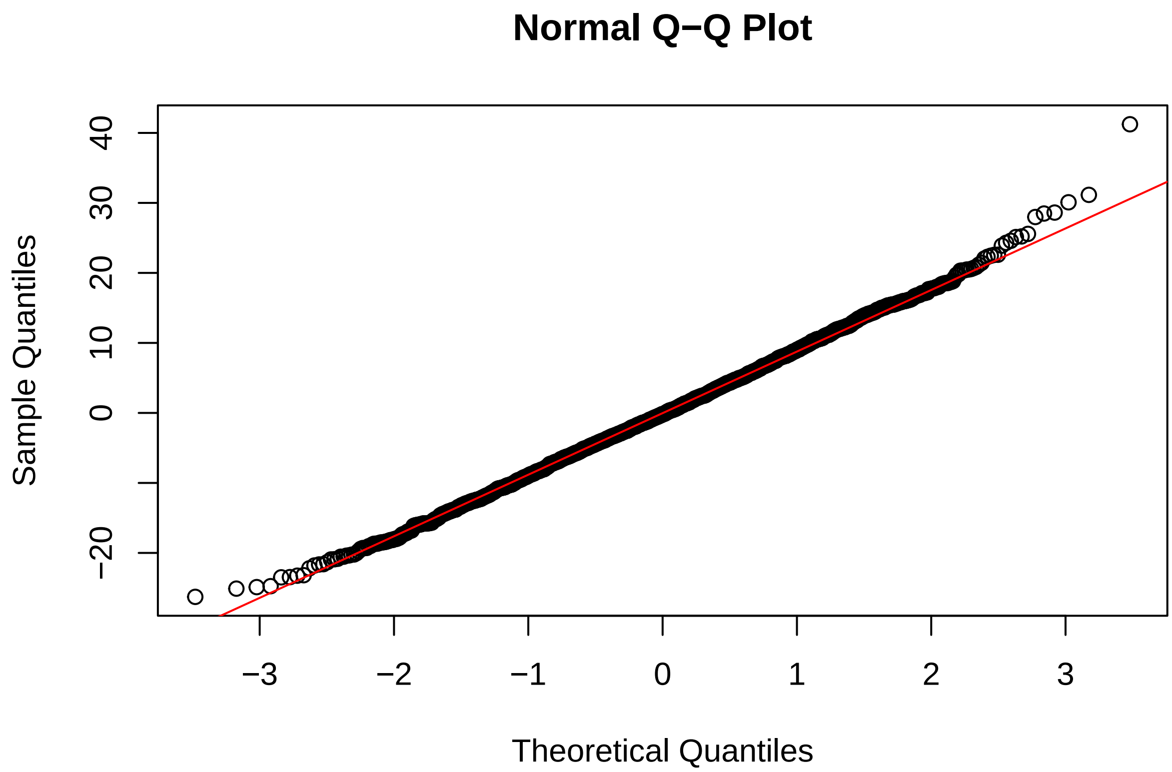
b.

```
mod = lm(mathscore ~ as.factor(school), data = nels_math_ses)
anova(mod)
```

```
## Analysis of Variance Table
##
## Response: mathscore
##                     Df Sum Sq Mean Sq F value    Pr(>F)
## as.factor(school)   99  48825  493.18   5.834 < 2.2e-16 ***
## Residuals         1893 160024   84.53
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

c.

```
qqnorm(resid(mod))
qqline(resid(mod), col="red")
```



d.

The problem did not specify which variance estimate to use. Therefore, you can either use pooled estimate of σ^2 as below, or an estimate using only one group.

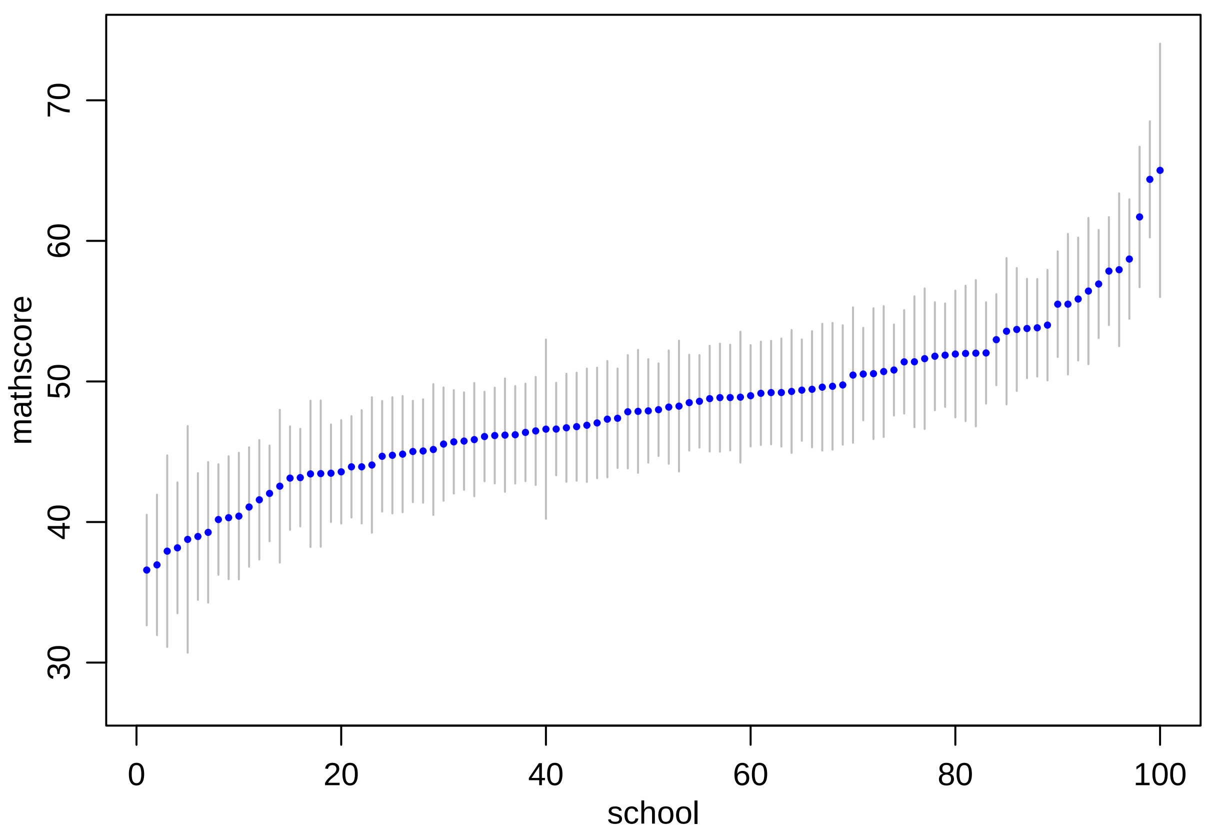
$$\bar{y}_j \pm \frac{t_{m(n-1),1-\alpha/2}}{\sqrt{n_j/\hat{\sigma}^2}}$$

```
gdotplot<-function(y,g,xlab="group",ylab="response",mcol="blue",
                   ocol="lightblue",sortgroups=TRUE,...)
{
  m<-length(unique(g))
  rg<-rank( tapply(y,g,mean),ties.method="first")
  if(sortgroups==FALSE){ rg<-1:m ; names(rg)<-unique(g)}
  mod = lm(y~g)
  sig2 = (summary(mod)$sigma)^2
  ylims = c(range(predict(mod))[1] - 1*sqrt(sig2),
            range(predict(mod))[2] + 1*sqrt(sig2))
  plot(c(1,m),ylims,type="n",xlab=xlab,ylab=ylab)

  for(j in unique(g))
  {
    yj<-y[g==j]
    rj<-rg[ match(as.character(j),names(rg)) ]
    nj<-length(yj)
```

```r
    yjmean = mean(yj)
    ci = qt(1- 0.05/2, length(y) - m) / sqrt(nj / sig2)
    segments(rj, yjmean + ci,
             rj, yjmean - ci,
             col="gray")
    points(rj,mean(yj),pch=16,cex=.5,col=mcol)
    if(rj == min(rg))
      cat(paste0("lowest CI (",
                 signif(yjmean - ci,5), ", ", signif(yjmean + ci,5),
                 "), nj = ", nj, ", width = ", signif(2*ci,3), "\n"))
    if(rj == max(rg))
      cat(paste0("highest CI (",
                 signif(yjmean - ci,5), ", ", signif(yjmean + ci,5),
                 "), nj = ", nj, ", width = ", signif(2*ci,3), "\n"))
  }
}

par(mar=c(3,3,1,1), mgp=c(1.75,.75,0))
gdotplot(nels_math_ses$mathscore,
         as.factor(nels_math_ses$school),
         xlab="school", ylab="mathscore")
```




```
## lowest CI (32.648, 40.518), nj = 21, width = 7.87
## highest CI (56.002, 74.033), nj = 4, width = 18
```

e.

```
# standardize variables to put on the same scale
nels_math_ses.std = scale(nels_math_ses)
mathscore = aggregate(mathscore ~ as.factor(school), data = nels_math_ses.std, mean)[,2]
mathdeg = aggregate(mathdeg ~ as.factor(school), data = nels_math_ses.std, mean)[,2]
ses = aggregate(ses ~ as.factor(school), data = nels_math_ses.std, mean)[,2]
par(mfrow=c(1,2))
plot(mathdeg, mathscore, xlab = "std mathdeg", ylab = "std mathscore")
abline(reg = lm(mathscore ~ mathdeg), col="red")
plot(ses, mathscore, xlab = "std ses", ylab = "std mathscore")
abline(reg = lm(mathscore ~ ses), col="red")
```