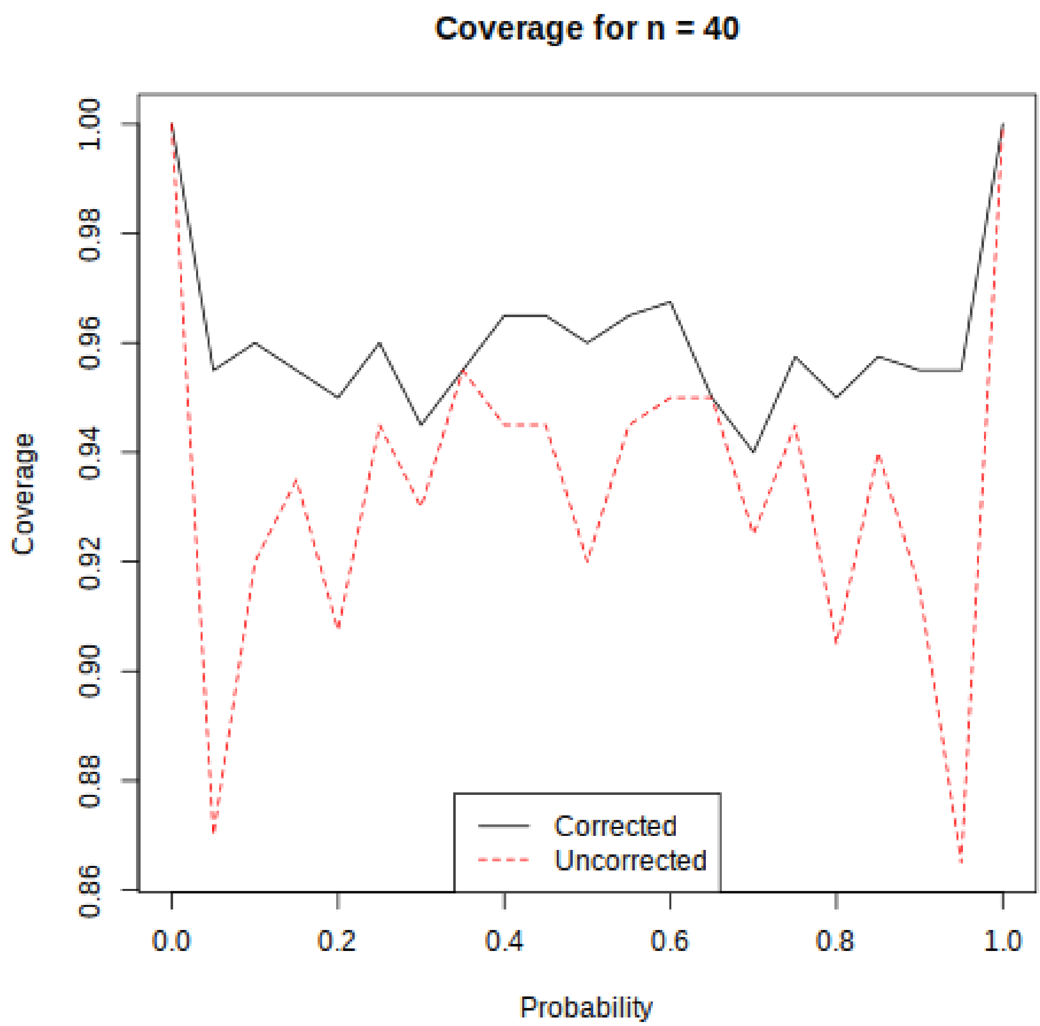
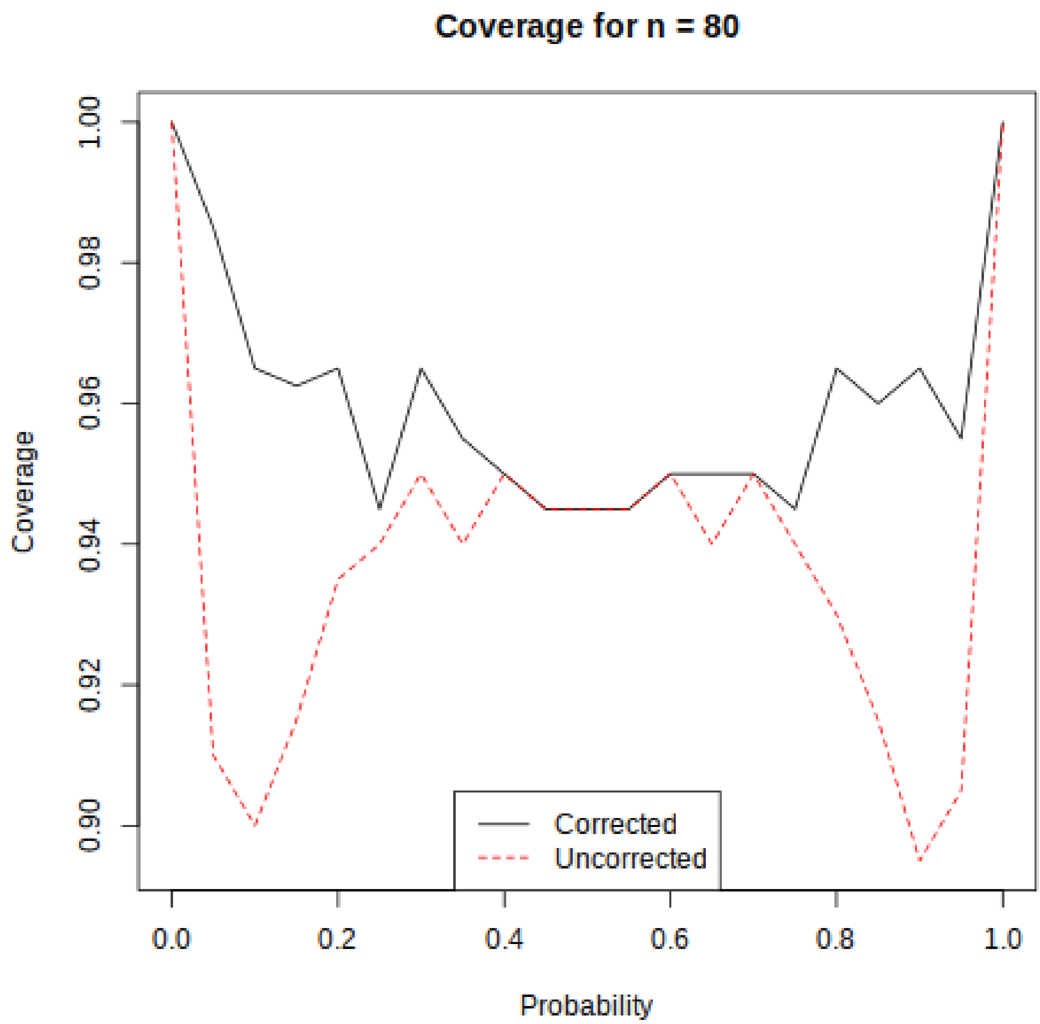
STA 032 R Extra Credit

Hardy Jones
999397426
Professor Melcon
Winter 2015

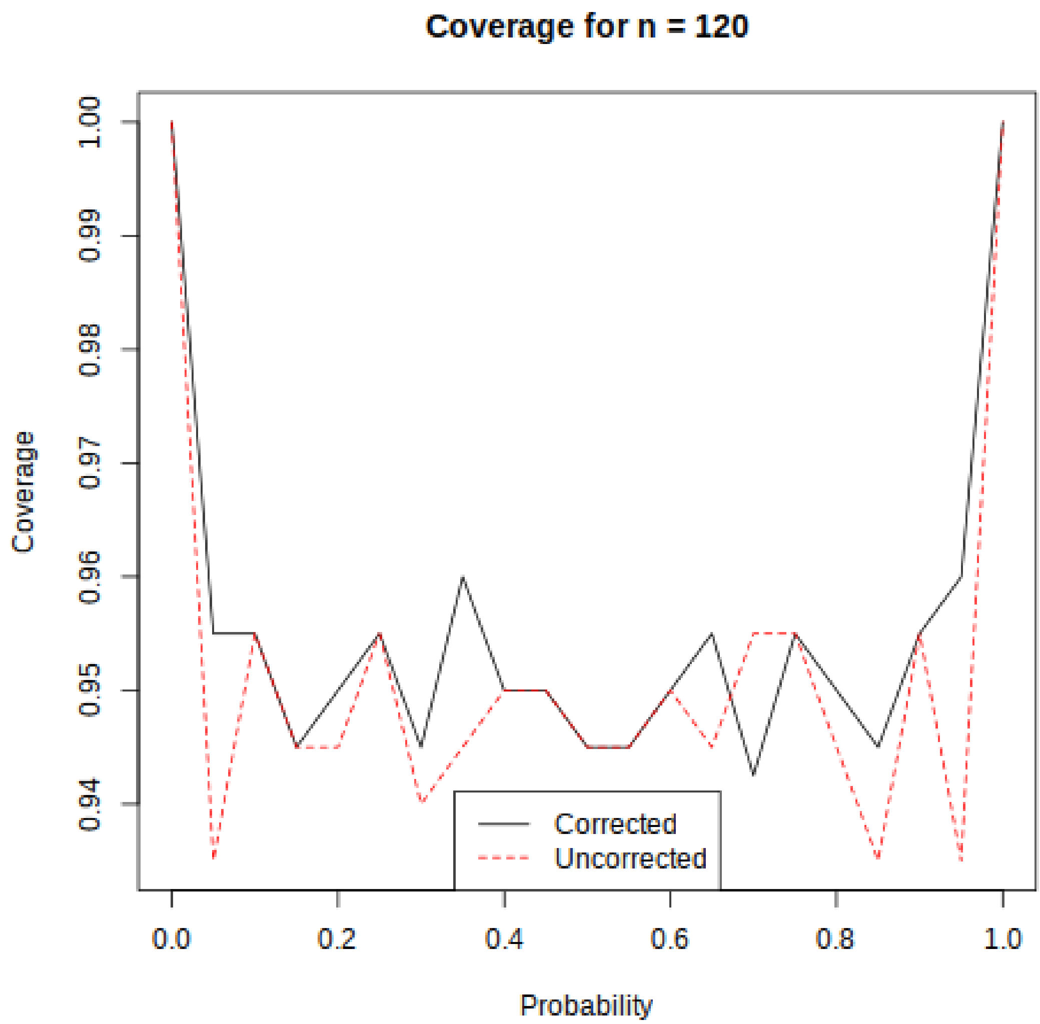


1. (a)



Coverage for n = 80
Coverage
1.00
0.98
0.96
0.94
0.92
0.90
Corrected
Uncorrected
0.0
0.2
0.4
0.6
0.8
1.0
Probability

(b)



(c)

(d) It seems that in each simulation, the extremes of 0 and 1 give a coverage of 100%. It also looks like as n increases, the uncorrected confidence interval becomes better. The uncorrected confidence interval produces similar results for $n = 120$. It also seems like when p is near 0.5, the differences between the two methods is negligible.

2. (a) This student will get a score of 87.3% for a grade of B.

   (b) This student will get a score of 74.32% for a grade of C.

   (c) This student will get a score of 84.03% for a grade of B.

   (d) This student needs at least 97 points on the final for an overall score of at least 83%.

Appendix A R code

Problem 1

```
source("../R_final/prob3.R")

Coverages <- function(alpha, n, N) function(p) {
  sims <- replicate(N, rbinom(n, 1, p))
  confs <- apply(sims, 2, Conf(alpha))

  corrected.low    <- confs[1, ]
  corrected.high   <- confs[3, ]
  uncorrected.low  <- confs[2, ]
  uncorrected.high <- confs[4, ]

  covered.mean <- covered(p)

  corrected.covered <- mapply(covered.mean, corrected.low, corrected.high)
  uncorrected.covered <- mapply(covered.mean, uncorrected.low, uncorrected.high)

  c(mean(corrected.covered), mean(uncorrected.covered))
}

MedianM <- function(alpha, n, N, M) function(p) {
  sims <- replicate(M, Coverages(alpha, n, N)(p))
  c(median(sims[1, ]), median(sims[2, ]))
}

ManyP <- function(alpha, n, N, M, many.p) {
  sims <- sapply(many.p, MedianM(alpha, n, N, M))
  matrix(c(sims[1, ], sims[2, ]), length(many.p), 2)
}

x <- ManyP(0.05, 40, 200, 100, seq(0, 1, 0.05))

png("prob1a.png")

matplot(seq(0, 1, 0.05), x, type=c("l"), col = 1:2, xlab = "Probability",
        ylab = "Coverage")
title("Coverage for n = 40")
legend("bottom", legend = c("Corrected", "Uncorrected"), lty=1:2, col = 1:2)

dev.off()

x <- ManyP(0.05, 80, 200, 100, seq(0, 1, 0.05))

png("prob1b.png")

matplot(seq(0, 1, 0.05), x, type=c("l"), col = 1:2, xlab = "Probability",
        ylab = "Coverage")
title("Coverage for n = 80")
legend("bottom", legend = c("Corrected", "Uncorrected"), lty=1:2, col = 1:2)

dev.off()
```

```r
x <- ManyP(0.05, 120, 200, 100, seq(0, 1, 0.05))

png("prob1c.png")

matplot(seq(0, 1, 0.05), x, type=c("l"), col = 1:2, xlab = "Probability",
        ylab = "Coverage")
title("Coverage for n = 120")
legend("bottom", legend = c("Corrected", "Uncorrected"), lty=1:2, col = 1:2)

dev.off()
```

Problem 2

```r
CalcGrade <- function(weights, student) {
  hws         <- subset(student, Category=="HW")$Grade
  exams       <- subset(student, Category=="Exam")$Grade
  finals      <- subset(student, Category=="Final")$Grade

  hw.grade    <- sum(hws) / length(hws) * weights$HW
  exam.grade  <- sum(exams) / length(exams) * weights$Exam
  final.grade <- sum(finals) / length(finals) * weights$Final

  score <- round(sum(hw.grade, exam.grade, final.grade), 2)

  c(score = score, letter = CalcLetter(score))
}

CalcLetter <- function(n) {
  if      (n >= 90) "A"
  else if (n >= 80) "B"
  else if (n >= 70) "C"
  else if (n >= 60) "D"
  else              "F"
}

Min83 <- function(weights, student) {
  cleaned <- na.omit(student)
  possibles <- sapply(c(1:100), function(n) {
    possible <- rbind(cleaned, data.frame(Grade = n, Category = "Final"))
    grade <- CalcGrade(weights, possible)
    if (grade[1] >= 83) n else NA
  })
  min(possibles, na.rm = TRUE)
}
```