HW3.R

Frank

Tue Sep 27 17:29:01 2016

```
#
# Homework 3         Dingjue Ji (dj333)                      J
#
# WORKED WITH:  ( list collaborators here please )
#
# STAPLE!  Now worth real points.
#
# DUE at 1 PM, Wednesday September 28.  Not at 2:15.
#
################################################################
# Part A: Work with data in EP_partlyclean
files <- dir('EP_partlyclean/')
pdf('HW3_plots.pdf')
options(warn = -1)
par(mfrow=c(2,2))
for(i in 1:length(files)){
  file <- paste('EP_partlyclean/', files[i], sep = '')
  state <- gsub("(.+)\\.csv", "\\1", files[i])
  state <- gsub("^.", toupper(substr(state, 1, 1)), state)
  csv <- read.csv(file, as.is = TRUE, header = FALSE, skip = 1)
  csv <- csv[, !apply(csv, 2, function(x) all(is.na(x)))]
  colnames(csv) <- c('Firm', 'Dates', 'Sample', 'Clinton', 'Trump', 'Spread')
  csv[, 'Sample'] <- gsub("[^\\.0-9]*([\\.0-9]+)[^\\.0-9]*", "\\1", csv[, 'Sample'])
  csv[, 'Sample'] <- as.numeric(csv[, 'Sample'])
  total <- csv[, 'Clinton'] + csv[, 'Trump']
  csv[, c('Clinton', 'Trump')] <- 100 / total * csv[, c('Clinton', 'Trump')]
  #Just for cleaning out the confusion
  csv <- csv[!is.na(csv$Sample), ]
  #Cumulate the vote counts during the poll and plot it.
  Clinton.Votes <- csv$Clinton*csv$Sample/sum(csv$Sample)
  Clinton.Votes <- cumsum(Clinton.Votes)
  Trump.Votes <- csv$Trump*csv$Sample/sum(csv$Sample)
  Trump.Votes <- cumsum(Trump.Votes)
  Polls <- 1:nrow(csv)
  plot(Clinton.Votes ~ Polls, main = paste('Cumulative Sample Votes in ', state, sep = ''),
       pch = 'C', col = 'red', ylim = range(c(Clinton.Votes, Trump.Votes), na.rm = TRUE),
       xlab = 'Polls (left side is most recent)', ylab = 'Sample Votes (%)')
  points(Trump.Votes ~ Polls, pch = 'T', col = 'blue')
  ans <- loess(Clinton.Votes ~ Polls, span = 1)
  lines(predict(ans) ~ Polls, col = "red", lwd = 2, lty = 2)
  ans <- loess(Trump.Votes ~ Polls, span = 1)
  lines(predict(ans) ~ Polls, col = 'blue', lwd = 2, lty = 2)
}
dev.off()
```

```
## pdf
##   2
```

```r
##############################################
# Part B: Expanding our previous simulation

B <- 10000
n <- 1950
p <- 0.7
x <- data.frame(Y1 = rbinom(B, n, p), Y2 = rbinom(B, n, p))

x$phat1 <- x$Y1 / n
x$phat2 <- x$Y2 / n
x$phatdiff <- x$phat1 - x$phat2
x$pvalue <- NA; x$confL <- NA; x$confU <- NA

for (i in 1:B) {
  ans <- prop.test(c(x$Y1[i], x$Y2[i]), n=c(n, n))
  x$pvalue[i] <- ans$p.value
  x[i, 7:8] <- ans$conf.int
}
x$confOK <- x$confL<=0 & x$confU>=0

p.test <- function(x, n, p, p.value = FALSE, conf = FALSE) {
  test <- prop.test(x, n, p)
  if(p.value) return(test$p.value)
  if(conf) return(as.numeric(test$conf.int))
}

Y1.conf <- sapply(x[, 'Y1'], p.test, n = 1950, p = 0.7, conf =TRUE)
Y1.conf <- t(Y1.conf)
Y2.conf <- sapply(x[, 'Y2'], p.test, n = 1950, p = 0.7, conf =TRUE)
Y2.conf <- t(Y2.conf)

confsoverlap <- apply(Y1.conf - Y2.conf[, c(2,1)], 1, prod) < 0
table(x$confOK, confsoverlap)
```

```
##        confsoverlap
##         FALSE TRUE
##   FALSE    47  382
##   TRUE      0 9571
```

```r
table(x$pvalue > 0.05, confsoverlap)
```

```
##        confsoverlap
##         FALSE TRUE
##   FALSE    47  382
##   TRUE      0 9571
```

```r
# Summary:
# Only when the confOK is true, the two different confidence intervals will
# definitely overlap with each other. Because the original one is 95% CI, so
# p.values > 0.05 will give the same anwser if there is no bad luck.
```