Coefficients and bias

Jacob LaRiviere

June 18, 2017, 2017

## R Markdown

This is an R Markdown document. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(plyr)

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

library(tidyr)  
library(ggplot2)  
library(reshape2)

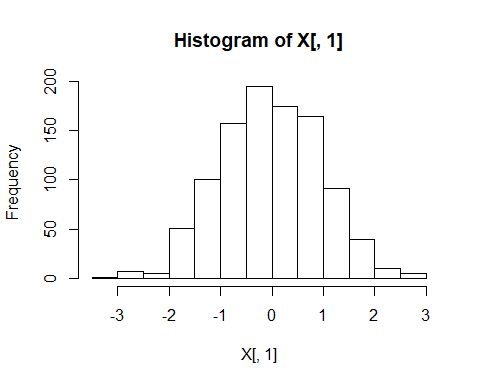
##   
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':  
##   
## smiths

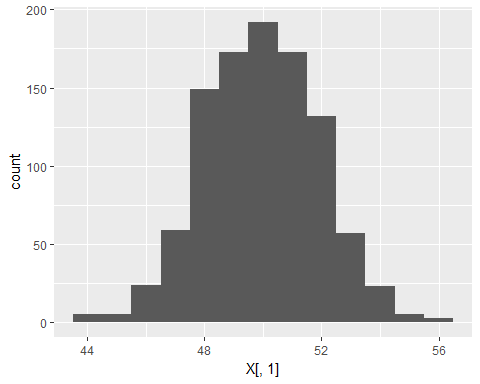
#setwd("C:/Users/jlariv/OneDrive/Econ 404/")  
#mydata <- read.csv("oj.csv")  
#colnames(mydata)[1-17]  
X <- NULL  
# Create a variable n which is the number of observations we'll look at.   
n = 1000  
  
#The variable p will be the width of that variable.   
p = 2  
  
#rnorm generates a random draw from a standard normal distribution.  
X = matrix(rnorm(n\*p), n, p)  
  
# This lets us look at the data  
head(X)

## [,1] [,2]  
## [1,] -0.9298356 -1.2153367  
## [2,] 0.3972522 1.6676900  
## [3,] 2.1368256 1.4732654  
## [4,] -1.5473415 0.5488630  
## [5,] 0.7460600 0.4183436  
## [6,] -0.6770107 1.2682192

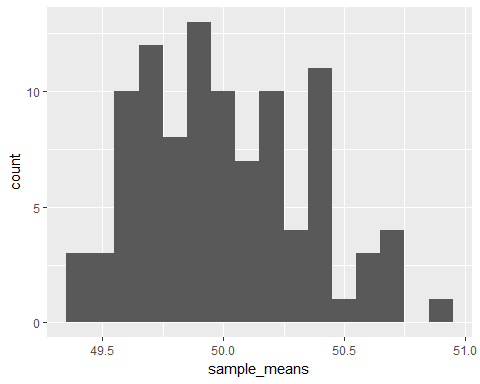
# Histogram of the first column. We can see that it looks like a bell curve  
hist(X[,1])



#NOTE: if you wanted to draw from something with a non-zero mean and more interesting variance that's: rnorm(n\*p)\*st\_dev + mu  
st\_dev <- 2  
mu <- 50  
  
X[,1] <- X[,1]\*st\_dev + mu  
X <-data.frame(X)  
ggplot(X,aes(X[,1])) + geom\_histogram(binwidth=1)



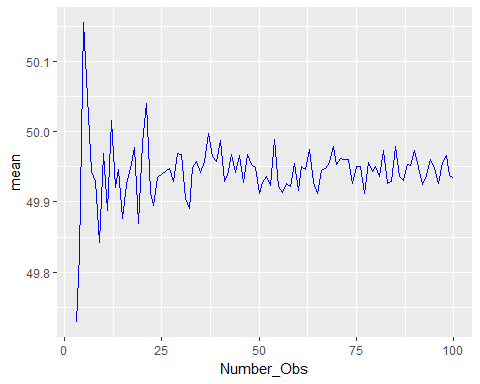
# Base R version: hist(X[,1])  
  
# So Jacob made this data up. We know that the mean is 50. There is a really nice function in R called "sample" which lets us take a random sub sample of our data say I'd like to take a sample of 25 observations   
obs = 25  
total\_samps = 100  
#sample\_means = matrix(0, total\_samps, 1)  
sample\_means = rep(0,total\_samps)  
  
for (i in 1:total\_samps) {  
subsample <- c(sample(X[,1], obs))  
sample\_means[i] = mean(subsample)  
}  
  
sample\_means <-data.frame(sample\_means)  
ggplot(sample\_means,aes(sample\_means)) + geom\_histogram(binwidth=.1)



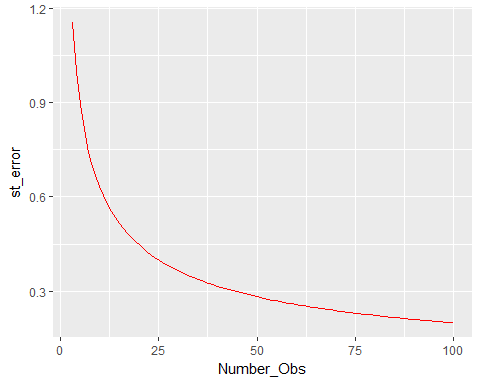
st\_err <- st\_dev/(obs^.5)  
  
#hist(sample\_means)  
summary(sample\_means)

## sample\_means   
## Min. :49.38   
## 1st Qu.:49.72   
## Median :49.96   
## Mean :50.00   
## 3rd Qu.:50.24   
## Max. :50.92

#Create a vector which increments up from an initial value to 100 by one  
obs\_0 = 3  
obs\_max = 100  
obs\_vec <- seq(obs\_0, obs\_max, 1)  
obs\_count <- length(obs\_vec)  
  
#Create dataframe to store our observations  
sim\_power = matrix(0, obs\_count, 3)  
sim\_power[,1]<-obs\_vec  
sim\_power <- data.frame(sim\_power)  
colnames(sim\_power) <- c("Number\_Obs","mean","st\_error")  
#Reset sample\_means  
sample\_means <- NULL  
sample\_means = rep(0,total\_samps)  
  
for (j in 1:obs\_count) {  
 for (i in 1:total\_samps) {  
 subsample <- c(sample(X[,1], sim\_power[j,1]))  
 sample\_means[i] = mean(subsample)  
 }  
 sim\_power[j,2] <- mean(sample\_means)  
 sim\_power[j,3] <- st\_dev/(sim\_power[j,1]^.5)  
}  
  
  
ggplot(sim\_power, aes(Number\_Obs)) + geom\_line(aes(y=mean),colour = "blue")



ggplot(sim\_power, aes(Number\_Obs)) + geom\_line(aes(y=st\_error),colour = "red")



# Question: Take a normal distribution with mu = 10 and variance of 9 (e.g., standard deviation of 3). Simulate a population of 10,000. What is the sample size needed to make the standard error of the sample mean sufficiently small so that no more than 5% of the sample means are less 9?

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.