Lab 3 Distributions

P. Johnson, L. Shaw, B. Rogers

Thursday, February 05, 2015

Stata Workshop

Feb. 13 Visualizing and Reporting Regression Results Using Stata

Speaker: Jacob Fowles, School of Public Affairs and Administration, KU

3pm-5pm Watson Library 455

URL: http://crmda.ku.edu/fowles-20150213

Lab 3 Topics

- Importing data of various types
- Central Limit Theorem demonstration
- What kind of distribution do you have?

What is your working directory?

- Good practice: check your working directory at the start of your session
- Change it if necessary

```
getwd()
```

```
## [1] "D:/Users/l076s857/Dropbox/GTA706"
```

```
# See ?setwd or lab 1 if you need help with this function
setwd("Labs")
```

Reading data in

- When working with real data, it is rare that it will be delivered to you in the format you like to work with
- Fortunate for you, R is able to read in many different types of file formats (with the help of packages)
- Last week in lab we read in a tab delimited file and a comma delimited file
- This week will import a text file, a Stata data set, and an SPSS data set

Text files

- Recall from lab 2, read.table and its variations can process a variety of text-based file formats
- In this example we are reading in a file and storing it in a dataframe called dat1

```
dat1 <- read.table("student-1.txt", header=TRUE)
```

- What does header = TRUE refer to?

```
dim(dat1) # what is the dimension of the data set?
```

```
## [1] 200   6
```

```
names(dat1) # What are the column names in the data set?
```

```
## [1] "x1" "y1" "x2" "x3" "x4" "x5"
```

Stata data sets

- When you install R, you get a number of different packages
- Unlike 'rockchalk', you do not need to use install.package, but instead just call it and then use functions inside of it

```
library(foreign)
dat2 <- read.dta("student-2.dta")
dim(dat2)
```

```
## [1] 200   6
```

```
names(dat2)
```

```
## [1] "x1" "y1" "x2" "x3" "x4" "x5"
```

SPSS data sets

- We are also going to use foreign to read in SPSS data
- In addition to Stata and SPSS, foreign can be used to import a number of other file types including SAS, Minitab, Systat, S, dBase and more

```
# library(foreign)
dat3 <- read.spss("bank.sav", to.data.frame = TRUE)
dim(dat3)
```

```
## [1] 474  11
```

```
names(dat3)
```

```
##  [1] "ID"       "SALBEG"   "SEX"      "TIME"     "AGE"      "SALNOW"
##  [7] "EDLEVEL"  "WORK"     "JOBCAT"   "MINORITY" "SEXRACE"
```

- What function would I use to look at the first rows of the data set?

Proving the Central Limit Theorem to Yourself

- A statistic is a value derived from a random sample.
- The distribution of a statistic is called sampling distribution.
- The sampling distribution of mean for large sample size is normal.

Create a sampling distribution

Create a numeric vector with 1000 elements, all equal 0.

```
x <- numeric(1000)
x[1:10]
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

Many ways to populate that vector

- You can either draw a random number, 1000 times (HOW TEDIOUS!)

```
x[1] <- mean(rnorm(453, mean=95, sd=23))
x[2] <- mean(rnorm(453, mean=95, sd=23))
x[3] <- mean(rnorm(453, mean=95, sd=23))
x[4] <- mean(rnorm(453, mean=95, sd=23))
# ....................
x[1000] <- mean(rnorm(453, mean=95, sd=23))
```

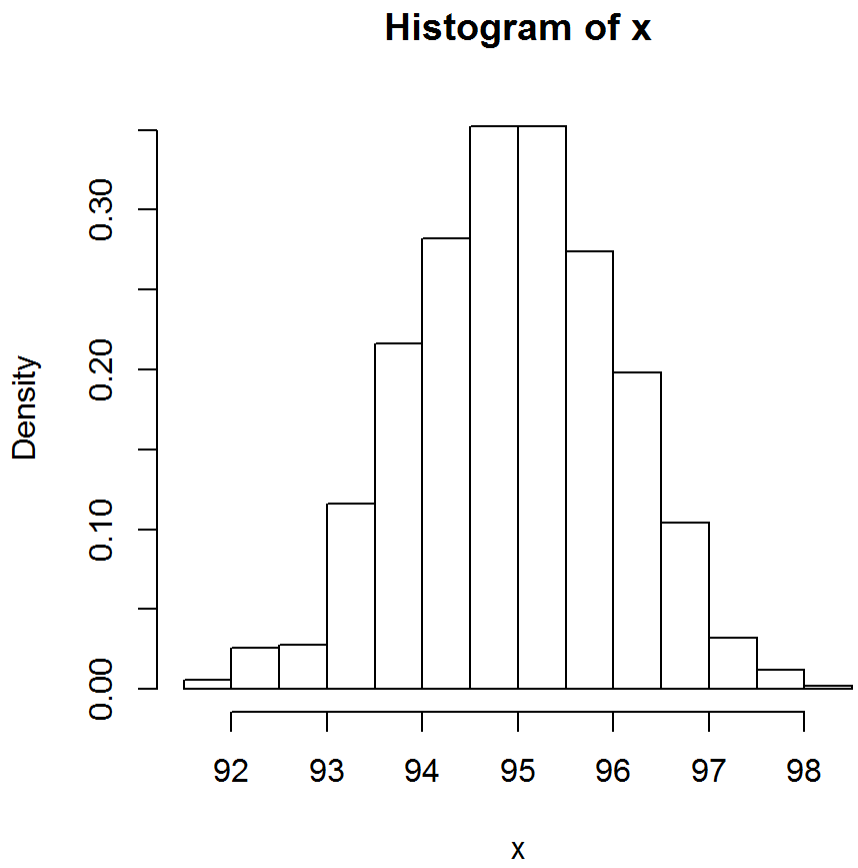
Use a for loop to populate the vector

- This method is less prone to mistakes than what we saw on the last slide though still a little slow
- We are using rnorm and mean functions

```
x <- numeric(1000)
for (i in 1:1000){
  x[i] <- mean(rnorm(453, mean=95, sd=23)) # mu is 95 and sigma is 23
}
```

View histogram of data we created with for loop

```
hist(x, prob=TRUE)
```



Functions and sapply

- In the previous slide we created a vector of means
- In R there is always more than one way to accomplish the same thing

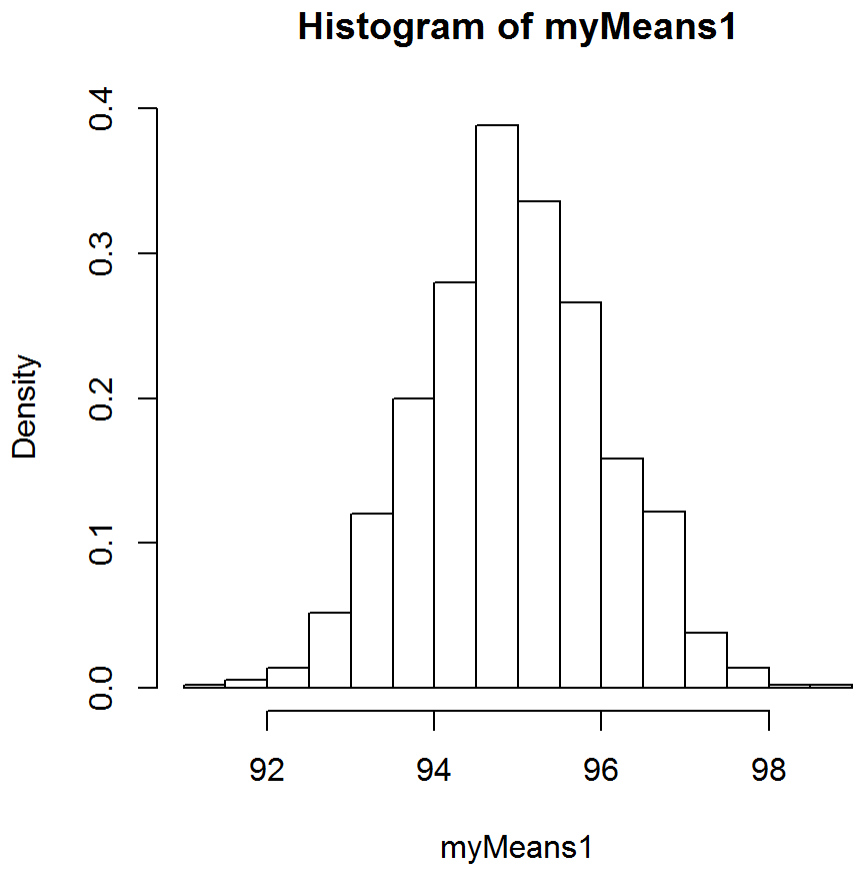
```
# Create a function to return a single mean
getNormalMean <- function(mu){
    e <- rnorm(453, mean = mu, sd = 23)
    mean(e)
}

# Get one mean value with a mean = 95
getNormalMean(95)
```

```
## [1] 94.55671
```

```
# Use sapply() to obtain a vector of means at once
myMu <- rep(95, 1000)  ## create 1000 mus
myMeans1 <- sapply(myMu, getNormalMean) # First argument is "mu" from myMu.
```

View histogram of data we created with sapply

```
hist(myMeans1, prob=TRUE)
```

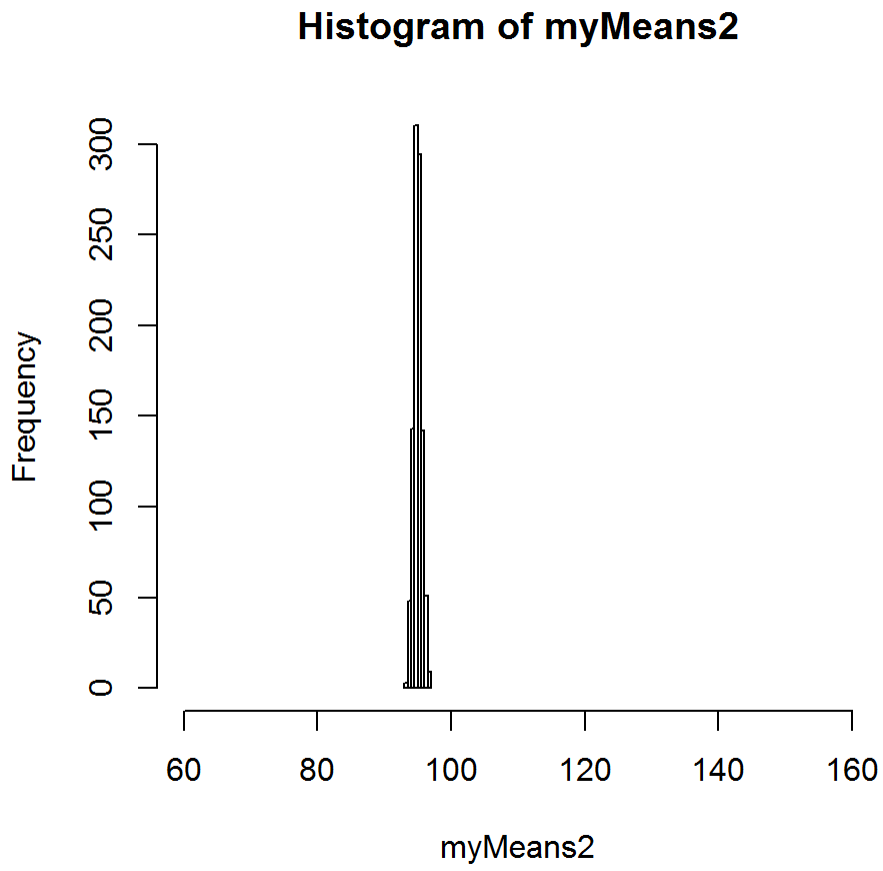


Play around with N

```
# Modify the function vary N
getNormalMean <- function(mu, N){
    e <- rnorm(N, mean = mu, sd = 23)
    mean(e)
}

myMu <- rep(95, 1000)
# Then specify sigma and N in the sapply() function
myMeans2 <- sapply(myMu, getNormalMean, N = 1500) # what if N=1500?
```

View histogram we created with N = 1500

```
hist(myMeans2, xlim=c(60,160))
```

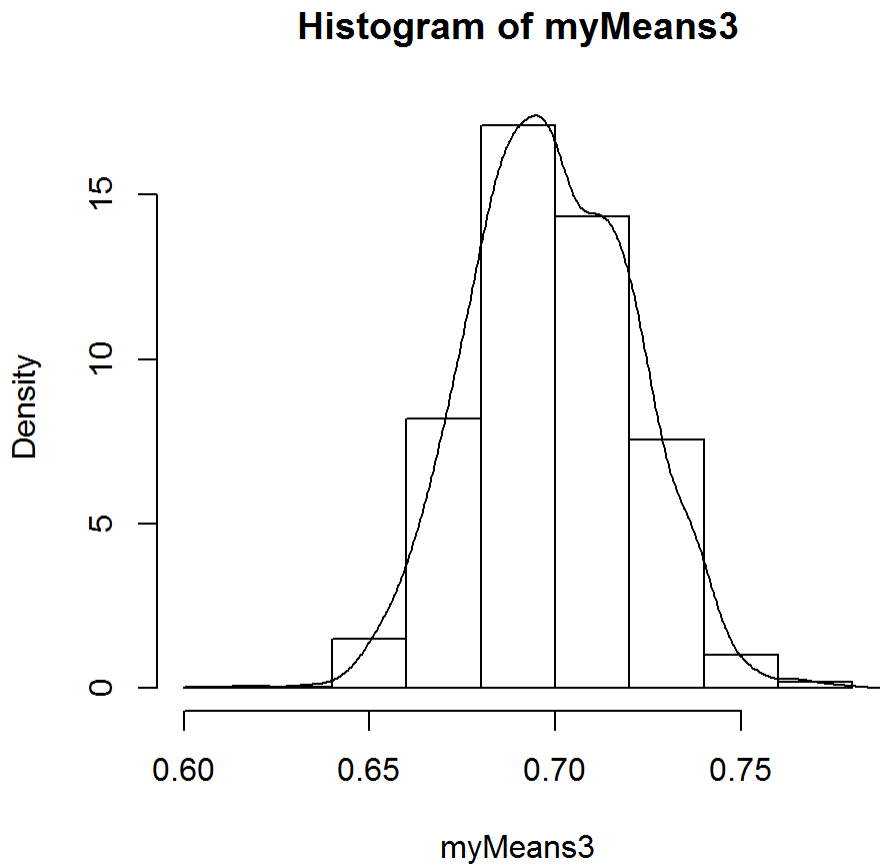


Try another distribution

```
# Change function to randomize data for a Poisson distribution
getPoissonMean <- function(lambda, N){
    e <- rpois(N, lambda)
    mean(e)
}

myLambda <- rep(0.7, 1000) ## create 1000 lambdas
myMeans3 <- sapply(myLambda, getPoissonMean, N = 1500) # what if N=1500?
```

View histogram of Poisson distributed data

```
hist(myMeans3, prob = TRUE)
lines(density(myMeans3))
```



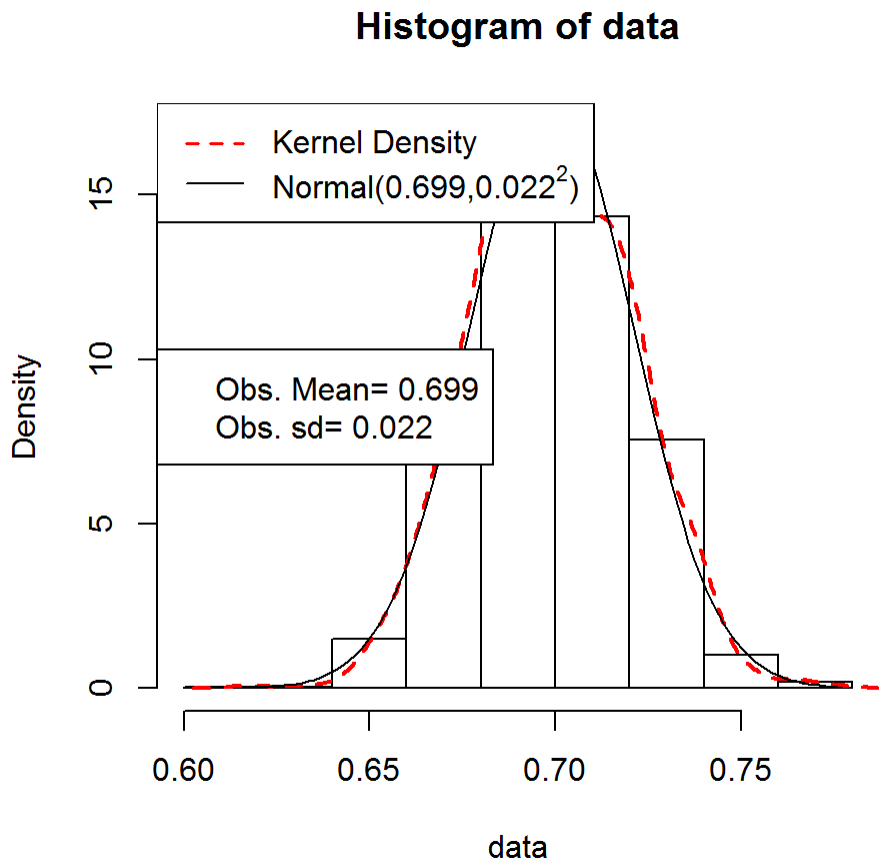
Draw normal and Kernel density line in same plot

```
drawHist <- function(data){
  hist(data, prob=T)
  lines(density(data), col="red", lty=2, lwd=2)
  cr <- range(data)
  ind <- seq(cr[1], cr[2], length.out=100)
  cm1 <- round(mean(data), 3)
  cs1 <- round(sd(data), 3)
  nprob <- dnorm(ind, m=cm1, s=cs1)
  lines(ind, nprob, lty=1, col="black")
  nlab <- bquote(paste("Normal(", .(round(cm1,3)),",", .(round(cs1,3))^2,")"))
  legend("topleft",legend=c("Kernel Density", as.expression(nlab)), lty=c(2,1),
    col=c("red","black"), lwd=c(1.5,1))
  legend("left", legend=c(paste("Obs. Mean=", cm1),paste("Obs. sd=",cs1)))
}
```

Placement of legends need some work

- My plots have all been sized to be 5 in x 5 in so function above needs some adjustment in order for it to work for me

```
drawHist(myMeans3)
```



What should be adjusted in the legend?

```
drawHist <- function(data){
  hist(data, prob=T)
  lines(density(data), col="red", lty=2, lwd=2)
  cr <- range(data)
  ind <- seq(cr[1], cr[2], length.out=100)
  cm1 <- round(mean(data), 3)
  cs1 <- round(sd(data), 3)
  nprob <- dnorm(ind, m=cm1, s=cs1)
  lines(ind, nprob, lty=1, col="black")
  nlab <- bquote(paste("Normal(", .(round(cm1,3)),",", .(round(cs1,3))^2,")"))
  legend("topleft",legend=c("Kernel Density", as.expression(nlab)), lty=c(2,1),
    col=c("red","black"), lwd=c(1.5,1))
  legend("left", legend=c(paste("Obs. Mean=", cm1),paste("Obs. sd=",cs1)))
}
```

How would you modify this for a Gamma distribution?

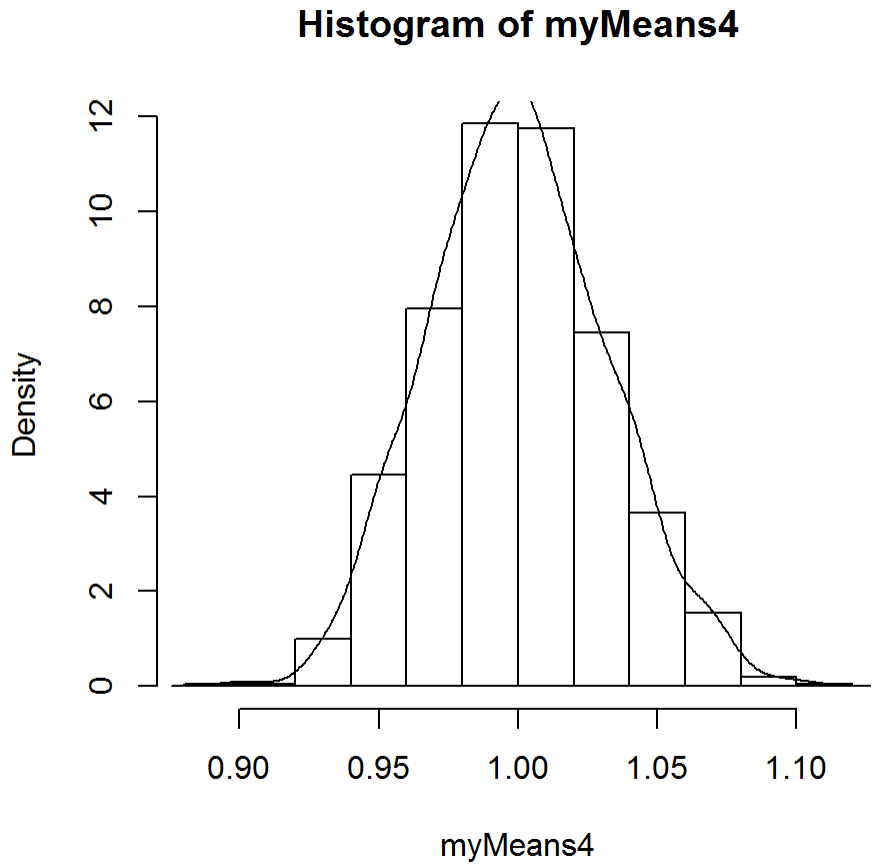
```
# Change function to randomize data for a Poisson distribution
getPoissonMean <- function(lambda, N){
    e <- rpois(N, lambda)
    mean(e)
}

myLambda <- rep(0.7, 1000) ## create 1000 lambdas
myMeans3 <- sapply(myLambda, getPoissonMean, N = 1500) # what if N=1500?
```

Modified for Gamma

```
getGammaMean <- function(shape, N){
    e <- rgamma(N, shape)
    mean(e)
}

myGamma <- rep(1, 1000)
myMeans4 <- sapply(myGamma, getGammaMean, N = 1000)
```

View histogram of Gamma distributed data

```
hist(myMeans4, prob = TRUE)
lines(density(myMeans4))
```



Lab exercise

- Download text data (.txt) and stata data (.dta)
- Please go to this webpage: http://pj.freefaculty.org/stat/ps706/student-Ex1/
- Read in data which should have 200 observations for 6 variables.
- Each column was randomly generated.
- Your task is to figure out what type of distribution you have for each column.

Steps

- What are the three basic steps Dr. Johnson led us through?