# HW 5 - Due Tuesday October 4, 2016 in moodle and hardcopy in class.

# Upload R file to Moodle with name: HW5\_490IDS\_YourClassID.R

# Do Not remove any of the comments. These are marked by #

# Please ensure that no identifying information (other than yur class ID)

# is on your paper copy, including your name

#For this problem we will start with a simulation in order to find out how large n needs

#to be for the binomial distribution to be approximated by the normal

#distribution.

#We will take m samples from the binomial distribution for some n and p.

#1.(4pts.) Let's let p=1/2, use the rbinom function to generate the sample of size m.

#Add normal curves to all of the plots.

#Use 3 values for n, 10, 30, and 50. Display the histograms as well as your

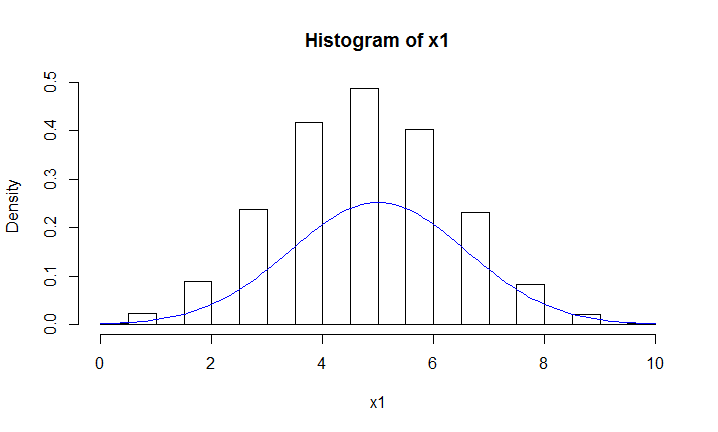
#code below.

> m = 10000

> x1 = rbinom(m, size = 10, prob = 0.5)

> hist(x1, freq = FALSE, breaks = 20)

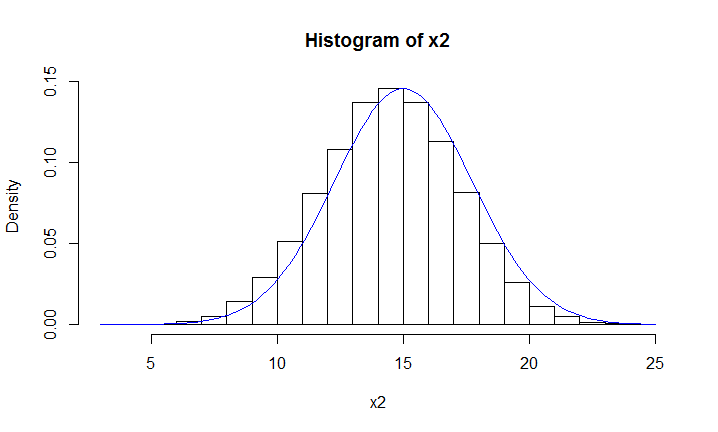
> curve(dnorm(x, 10\*0.5, sqrt(10\*0.5\*0.5)), col="blue", lwd=1, add=TRUE)



> x2 = rbinom(m, size = 30, prob = 0.5)

> hist(x2, freq = FALSE, breaks = 20)

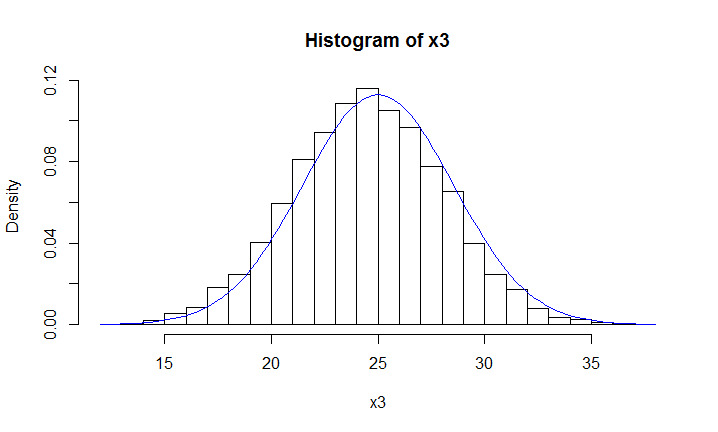
> curve(dnorm(x, 30\*0.5, sqrt(30\*0.5\*0.5)), col="blue", lwd=1, add=TRUE)



> x3 = rbinom(m, size = 50, prob = 0.5)

> hist(x3, freq = FALSE, breaks = 20)

> curve(dnorm(x, 50\*0.5, sqrt(50\*0.5\*0.5)), col="blue", lwd=1, add=TRUE)



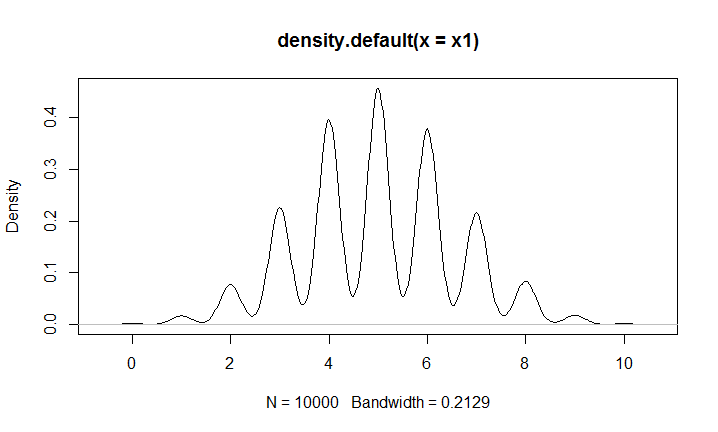
#1b.)(3pts.) Now use the techniques described in class to improve graphs.

# Explain each step you choose including why you are making the change. You

# might consider creating density plots, changing color, axes, labeling, legend, and others for example.

# Creating density plot to get a smoothed histogram makes it easier to compare with the normal curve.

plot(density(x1))



# The default bandwidth is small. To make it bigger can show the outline clearly.

> plot(density(x1, bw = 0.5), ylim = c(0,0.25), xlab = "x1", main = "density plot of binomial distribution(n = 10, p = 0.5)")

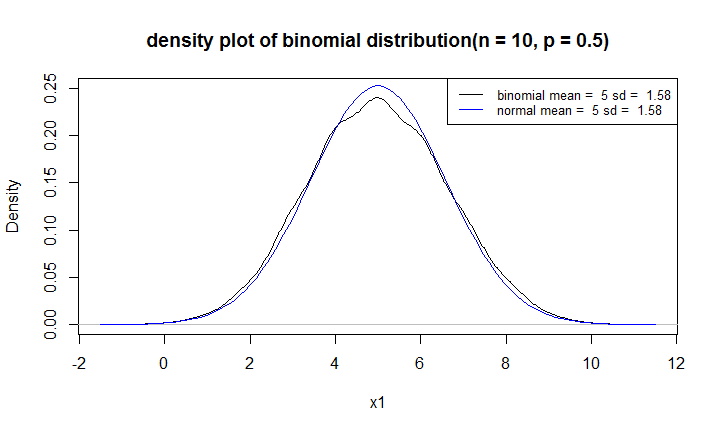
> curve(dnorm(x, 10\*0.5, sqrt(10\*0.5\*0.5)), col="blue", lwd=1, add=TRUE)

# The normal curve has a higher height than that of density plot, so I enlarge the range for the scale of y axis.

# Setting the normal curve a different color to make comparison with the binomial distribution.

> legend("topright", legend = paste("", c("binomial","normal"), "mean = ", c(5,5), "sd = ", c(1.58,1.58)), lwd=1, col = c("black", "blue"), cex=0.8, text.font = 1.5)

# A legend with mean and sd can make it clear to identify the two curves.

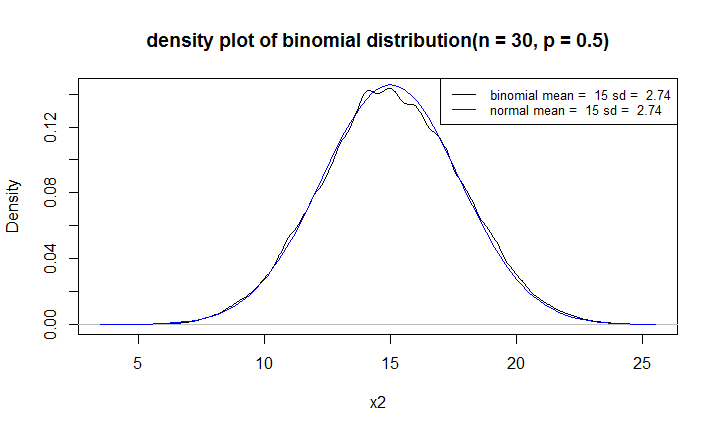


> plot(density(x2, bw = 0.5), xlab = "x2", main = "density plot of binomial distribution(n = 30, p = 0.5)")

> curve(dnorm(x, 30\*0.5, sqrt(30\*0.5\*0.5)), col="blue", lwd=1, add=TRUE)

# The normal curve has an approximate height with that of density plot, so there is no need to change the range of y axis.

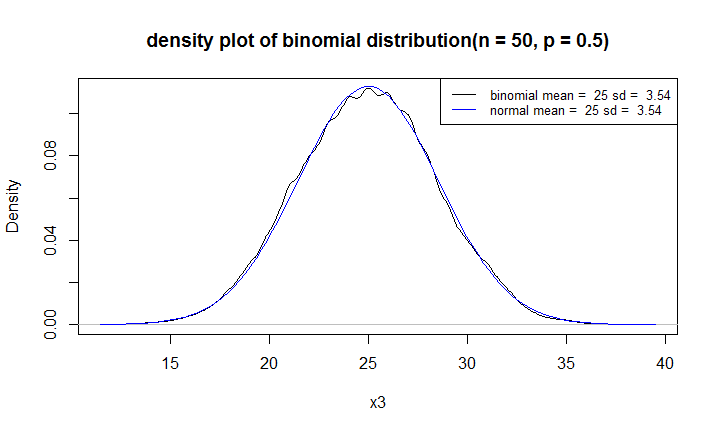
> legend("topright", legend = paste("", c("binomial","normal"), "mean = ", c(15,15), "sd = ", c(2.74,2.74)), lwd=1, col = c("black", "blue"), cex=0.8, text.font = 1.5)



> plot(density(x3, bw = 0.5), xlab = "x3", main = "density plot of binomial distribution(n = 50, p = 0.5)")

> curve(dnorm(x, 50\*0.5, sqrt(50\*0.5\*0.5)), col="blue", lwd=1, add=TRUE)

> legend("topright", legend = paste("", c("binomial","normal"), "mean = ", c(25,25), "sd = ", c(3.54,3.54)), lwd=1, col = c("black", "blue"), cex=0.8, text.font = 1.5)



#Q2.) (2pts.)

#Why do you think the Data Life Cycle is crucial to understanding the opportunities

#and challenges of making the most of digital data? Give two examples.

# Such opportunities and difficulties lie in data capture, storage, searching, sharing, analysis,

# and visualization, which are key elements in the Data Life Cycle.

# Information is increasing at an exponential rate, but information processing methods are improving relatively slowly.

# Currently, a limited number of tools are available to completely address the issues in Big Data analysis.

# For example, Hadoop cannot solve the real problems of storage, searching, sharing, visualization,

# and real-time analysis ideally in terms of Data Life Cycle. For large-scale data analysis, SAS, R, and Matlab are unsuitable.

# Graph lab provides a framework that calculates graph-based algorithms but it does not manage data effectively.

# Therefore, proper tools to adequately exploit Big Data are still lacking. (Khan et al., 2014)

# Data Life Cycle is crucial to develop proper tools and to solve data processing problems

# based on the before and after steps during the cycle.

# Also, challenges in Big Data analysis include "data inconsistency and incompleteness, scalability, timeliness, and security".

# (Khan et al., 2014). Prior to data analysis, data must be well constructed based on the Data Life Cycle.

# However, considering the variety of datasets, the efficient representation, access, and analysis

# of unstructured or semistructured data are still challenging (Khan et al., 2014).

# Therefore, numerous data preprocessing techniques, including data cleaning, integration,

# transformation, and reduction, should be applied to remove noise and correct inconsistencies.

# Data Life Cycle is crucial to the achievement of data consistency and completeness considering

# the bond between different segments in the cycle.

Khan, N., Yaqoob, I., Hashem, I. A. T., Inayat, Z., Mahmoud Ali, W. K., Alam, M., … Gani, A. (2014). Big Data: Survey, Technologies, Opportunities, and Challenges. The Scientific World Journal, 2014, 712826. <http://doi.org/10.1155/2014/712826>

###Part 2###

#3.) San Francisco Housing Data

#

# Load the data into R.

load(url("http://www.stanford.edu/~vcs/StatData/SFHousing.rda"))

# (2 pts.)

# What is the name and class of each object you have loaded into your workspace?

### Your code below

objects()

class(cities)

class(housing)

### Your answer here

# > objects()

# [1] "cities" "housing"

# > class(cities)

# [1] "data.frame"

# > class(housing)

# [1] "data.frame"

# There are two data frame class objects. One is "cities", and the other is "housing".

# What are the names of the vectors in housing?

### Your code below

names(housing)

### Your answer here

# > names(housing)

# [1] "county" "city" "zip" "street" "price" "br" "lsqft" "bsqft" "year"

# [10] "date" "long" "lat" "quality" "match" "wk"

# How many observations are in housing?

### Your code below

dim(housing)

### Your answer here

# There are 281506 observations.

# Explore the data using the summary function.

# Describe in words two problems that you see with the data.

#### Write your response here

summary(cities)

summary(housing)

# In the cities object, there are NAs in the longitude and latitude variables. The names of

# the cities are not shown as a variable (column) in the object.

# In the housing object, there are many NAs in zip, lsqft, bsqft, year, long, lat, quality and match variables.

# The Max of year is 3894, which is unreasonable. The Min is 0, which is inappropriate.

# The calculation of min, 1st Qu, median, etc seems inappropriate to date and wk.

# Q5. (2 pts.)

# We will work the houses in Albany, Berkeley, Piedmont, and Emeryville only.

# Subset the data frame so that we have only houses in these cities

# and keep only the variables city, zip, price, br, bsqft, and year

# Call this new data frame BerkArea. This data frame should have 4059 observations

# and 6 variables.

> BerkArea = housing[housing$city %in% c("Albany","Berkeley","Piedmont","Emeryville"), c("city","zip","price","br","bsqft","year")]



# Q6. (2 pts.)

# We are interested in making plots of price and size of house, but before we do this

# we will further subset the data frame to remove the unusually large values.

# Use the quantile function to determine the 99th percentile of price and bsqft

# and eliminate all of those houses that are above either of these 99th percentiles

# Call this new data frame BerkArea, as well. It should have 3999 observations.

> pricelimit = quantile(BerkArea$price, 0.99)

> bsqftlimit = quantile(BerkArea$bsqft, 0.99, na.rm = TRUE)

> pricelimit

99%

2285500

> bsqftlimit

99%

4035.76

> BerkArea = BerkArea[BerkArea$price<=pricelimit & BerkArea$bsqft<=bsqftlimit,]



# Q7 (2 pts.)

# Create a new vector that is called pricepsqft by dividing the sale price by the square footage

# Add this new variable to the data frame.

> pricepsqft = BerkArea$price / BerkArea$bsqft

> BerkArea["pricepsqft"] = pricepsqft

# Q8 (2 pts.)

# Create a vector called br5 that is the number of bedrooms in the house, except

# if this number is greater than 5, it is set to 5. That is, if a house has 5 or more

# bedrooms then br5 will be 5. Otherwise it will be the number of bedrooms.

> br5 = ifelse(BerkArea$br>5, 5, BerkArea$br)

# Q9 (4 pts. 2 + 2 - see below)

# Use the rainbow function to create a vector of 5 colors, call this vector rCols.

# When you call this function, set the alpha argument to 0.25 (we will describe what this does later)

# Create a vector called brCols of 4059 colors where each element's

# color corresponds to the number of bedrooms in the br5.

# For example, if the element in br5 is 3 then the color will be the third color in rCols.

# (2 pts.)

rCols = rainbow(n = 5, alpha = 0.25)

brCols = rCols[br5]

######

# We are now ready to make a plot.

# Try out the following code

plot(pricepsqft ~ bsqft, data = BerkArea,

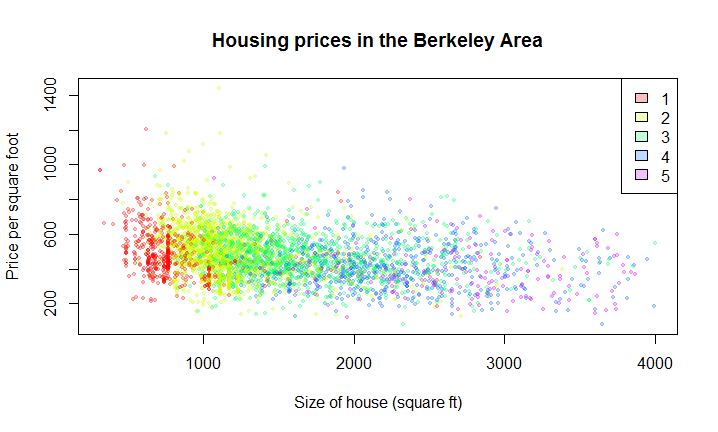
main = "Housing prices in the Berkeley Area",

xlab = "Size of house (square ft)",

ylab = "Price per square foot",

col = brCols, pch = 19, cex = 0.5)

legend(legend = 1:5, fill = rCols, "topright")



# (2 pts.)

### What interesting features do you see that you didn't know before making this plot?

# In general, prices per square foot slightly go down as the size of houses increases. For

# houses which have about 2000 square feet or so, the number of bedrooms vary from 3 to 5.

# Prices per square foot of house with one bedroom and two bedrooms have larger range than those

# of houses with more than 4 bedrooms.

# (2 pts.)

# Replicate the boxplots presented in class, with the boxplots sorted by median housing price (slide 45 of the lecture notes)

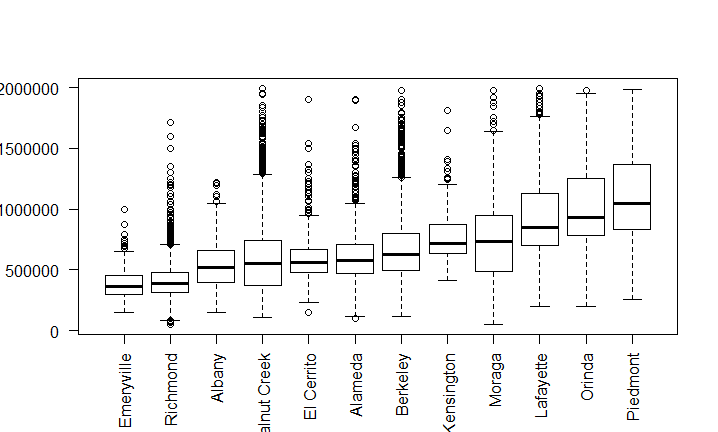
> someCities = c("Albany", "Berkeley", "El Cerrito", "Emeryville", "Piedmont", "Richmond", "Lafayette", "Walnut Creek", "Kensington","Alameda", "Orinda", "Moraga")

> shousing = housing[housing$city %in% someCities & housing$price < 2000000,]

> shousing$city = as.character(shousing$city)

> bymedian = with(shousing, reorder(city, price, median))

> boxplot(shousing$price ~ bymedian, las = 2)



# For BerkArea data frame:

> BerkArea$city = as.character(BerkArea$city)

> boxplot.median = with(BerkArea, reorder(city, price, median))

> boxplot(BerkArea$price ~ boxplot.median, las = 2)

