HW5_490IDS_17

17

October 3, 2016

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.

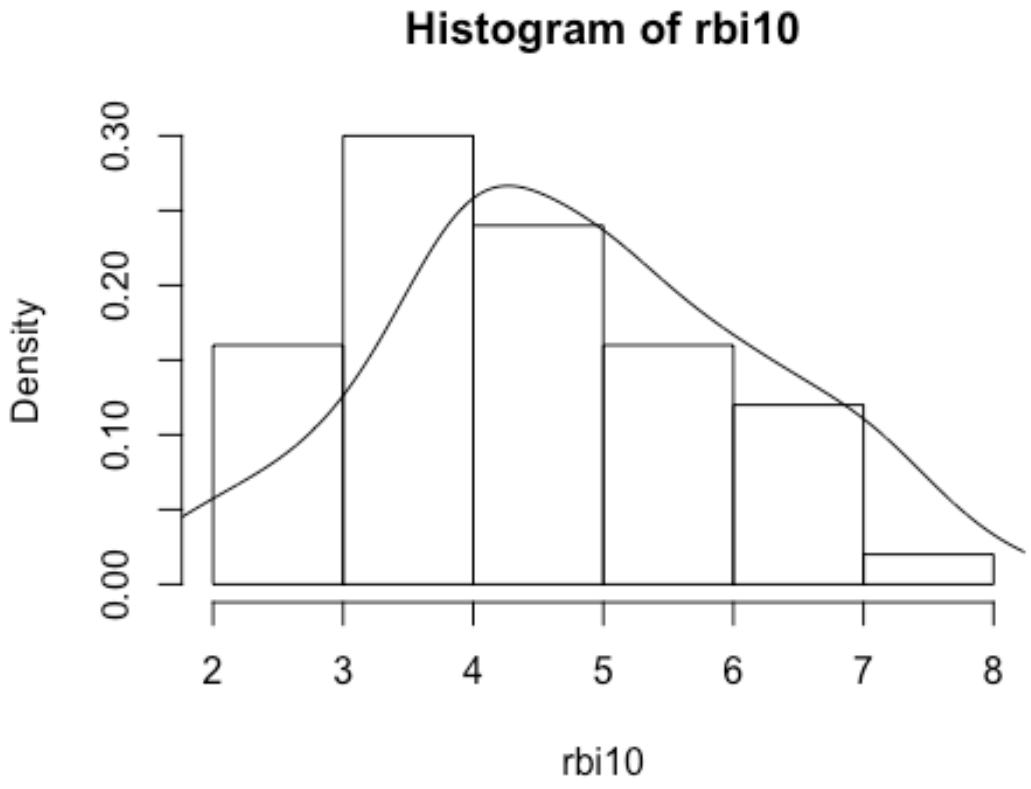
1.(4pts.) Let's let p=1/2, use the rbinom function to generate the sample of size m.

Add normal curves to all of the plots.

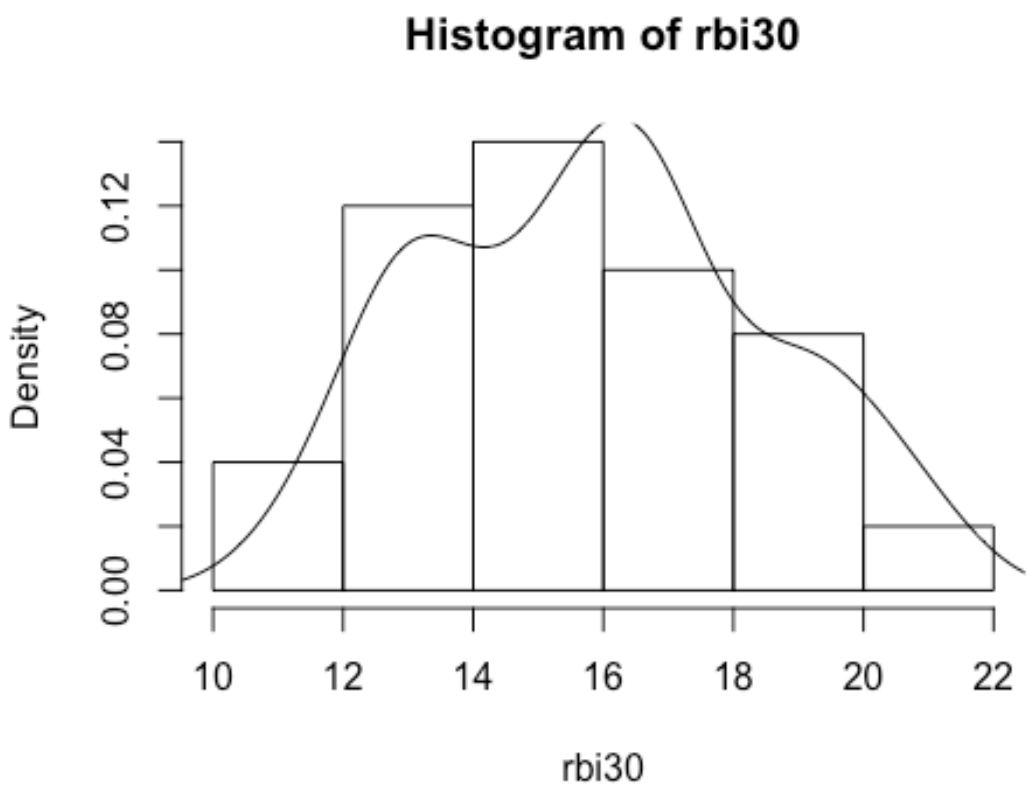
Use 3 values for n, 10, 30, and 50. Display the histograms as well as your

code below.

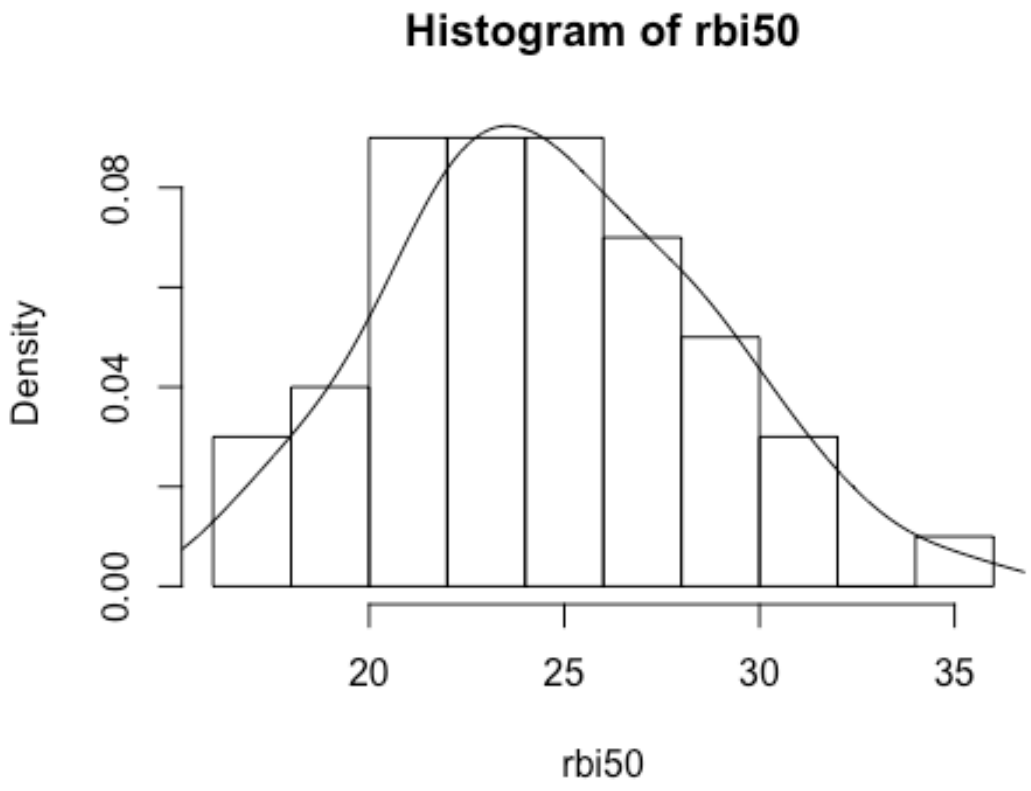
```
m = 50
rbi10 = rbinom(m, 10,  0.5)
rbi30 = rbinom(m, 30,  0.5)
rbi50 = rbinom(m, 50,  0.5)
hist(rbi10, prob = TRUE)
lines(density(rbi10))
```




```r
hist(rbi30, prob = TRUE)
lines(density(rbi30))
```




```r
hist(rbi50, prob = TRUE)
lines(density(rbi50))
```

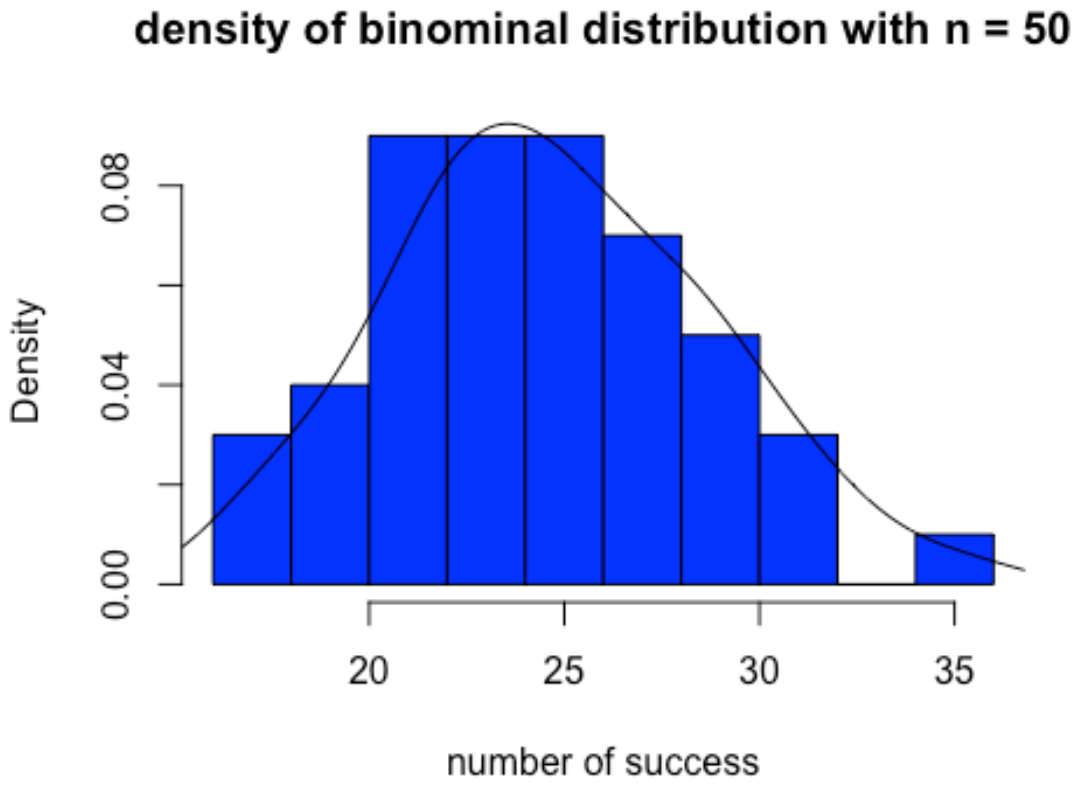


1b.)(3pts.) Now use the techniques described in class to improve graphs.

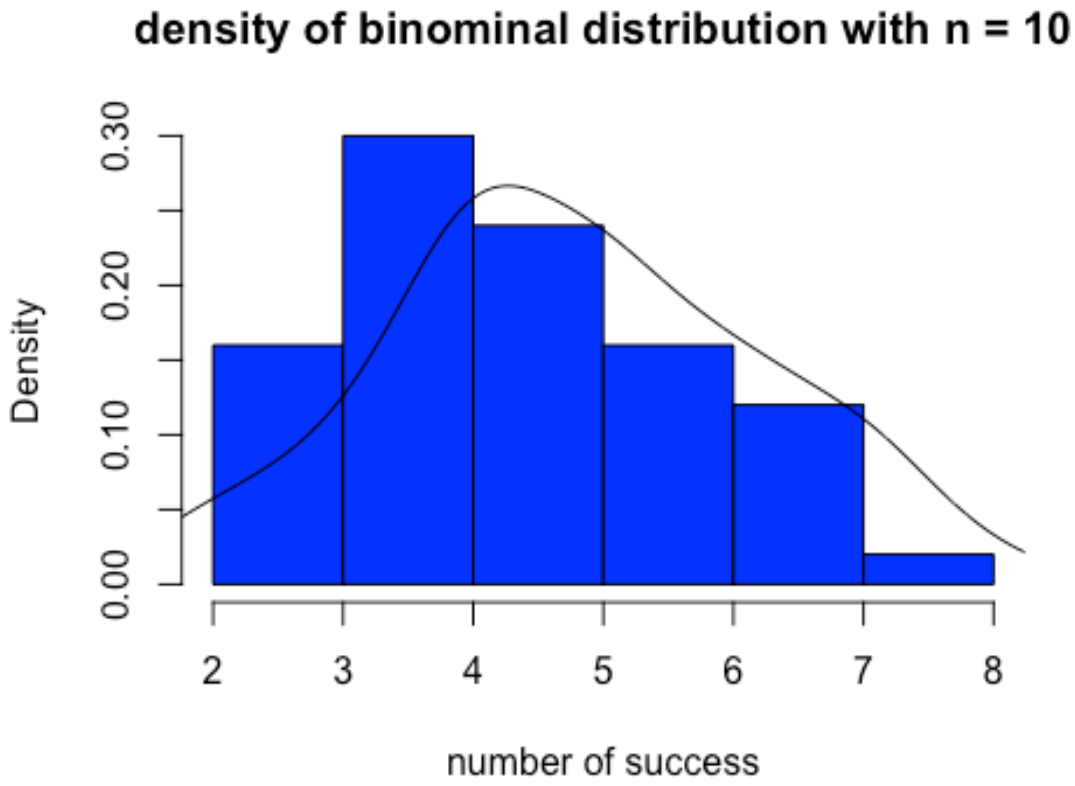
Explain each step you choose including why you are making the change. You

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

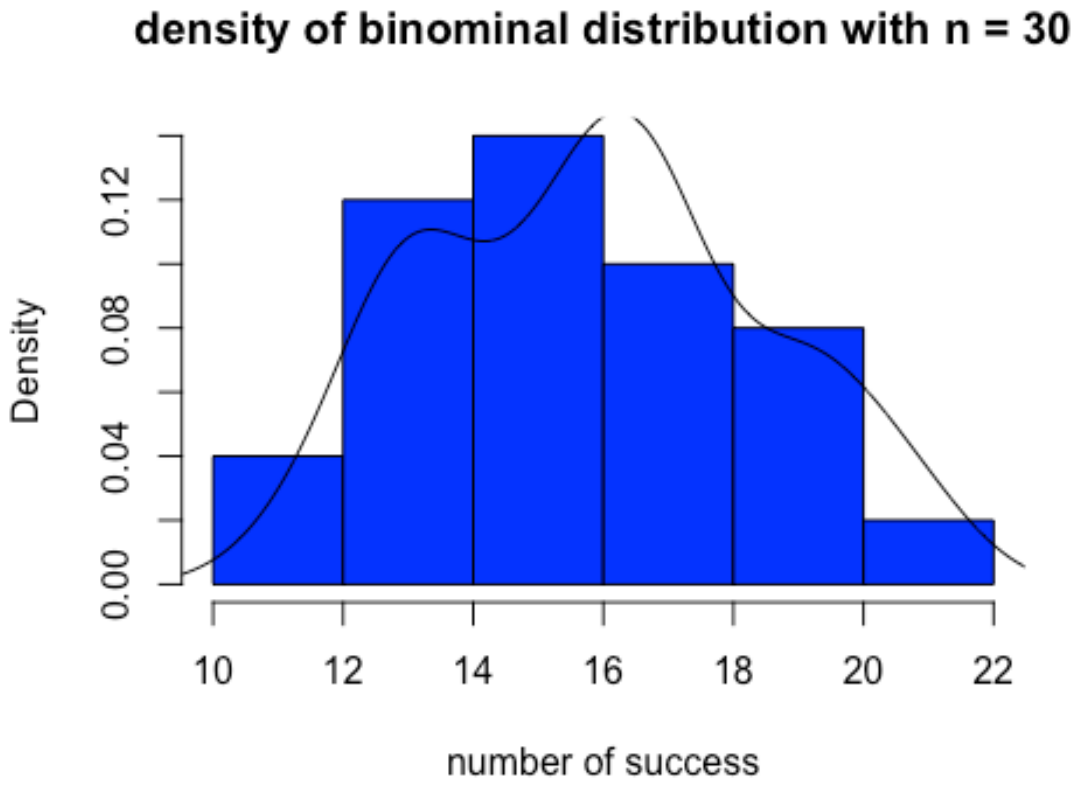
```
hist(rbi50, prob = TRUE, main = "density of binominal distribution with n =
50", xlab = "number of success", col = "blue")
lines(density(rbi50))
```




```r
hist(rbi10, prob = TRUE, main = "density of binominal distribution with n =
10", xlab = "number of success", col = "blue")
lines(density(rbi10))
```




```r
hist(rbi30, prob = TRUE, main = "density of binominal distribution with n = 30", xlab = "number of success", col = "blue")
lines(density(rbi30))
```




```
print("I added a title, added x axis label, changed color")
```

```
## [1] "I added a title, added x axis label, changed color"
```

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.

```
print("1. Consider you have made a successful product and it is very popular. But when will you need to make the next generation of this product? How soon are customers done with the old ones? We need to collect and analyze data to find out the answer and make smart decisions for maximizing the gain. And since the data will be huge and there are so many factors, data life cycle is important.")
```

```
## [1] "1. Consider you have made a successful product and it is very
popular. But when will you need to make the next generation of this product?
How soon are customers done with the old ones? We need to collect and analyze
data to find out the answer and make smart decisions for maximizing the gain.
And since the data will be huge and there are so many factors, data life
cycle is important."
```

```
print("2. In a president campaign, how much effort will be needed in each
state? This question can be predicted by analyzing datas. Also the past data
could be useful. That's why we emphasize on preserving the data smartly.")
```

```
## [1] "2. In a president campaign, how much effort will be needed in each
state? This question can be predicted by analyzing datas. Also the past data
could be useful. That's why we emphasize on preserving the data smartly."
```

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

```
lapply(housing, class)
```

```
## $county
## [1] "factor"
##
## $city
## [1] "factor"
##
## $zip
## [1] "factor"
##
## $street
## [1] "character"
```

```
##
## $price
## [1] "numeric"
##
## $br
## [1] "integer"
##
## $lsqft
## [1] "numeric"
##
## $bsqft
## [1] "integer"
##
## $year
## [1] "integer"
##
## $date
## [1] "POSIXt"  "POSIXct"
##
## $long
## [1] "numeric"
##
## $lat
## [1] "numeric"
##
## $quality
## [1] "factor"
##
## $match
## [1] "factor"
##
## $wk
## [1] "Date"
```

Your answer

```
print("county:factor ;city[1] factor;$zip[1] factor;$street[1]
character;$price[1] numeric; $br[1] integer;$lsqft[1] numeric;$bsqft[1]
integer; $year[1] integer; $date[1] POSIXt  POSIXct;$long[1] numeric;$lat[1]
numeric;$quality[1] factor;$match[1] factor;$wk[1] Date")
```

```
## [1] "county:factor ;city[1] factor;$zip[1] factor;$street[1]
character;$price[1] numeric; $br[1] integer;$lsqft[1] numeric;$bsqft[1]
integer; $year[1] integer; $date[1] POSIXt  POSIXct;$long[1] numeric;$lat[1]
numeric;$quality[1] factor;$match[1] factor;$wk[1] Date"
```

What are the names of the vectors in housing?

Your code below

```
colnames(housing)
```

```
##  [1] "county"  "city"    "zip"     "street"  "price"   "br"      "lsqft"
##  [8] "bsqft"   "year"    "date"    "long"    "lat"     "quality" "match"
## [15] "wk"
```

Your answer here

```
print("county  city zip    street  price   br      lsqft   bsqft   year
date   long   lat  quality match   wk ")
```

```
## [1] "county  city zip    street  price   br      lsqft   bsqft   year
date   long   lat  quality match   wk "
```

How many observations are in housing?

Your code below

```
dim(housing)
```

```
## [1] 281506     15
```

Your answer here

```
print("281506 observations")
```

```
## [1] "281506 observations"
```

Explore the data using the summary function.

```
summary(housing)
```

```
##                  county                  city             zip
##  Santa Clara County :70424   Oakland      : 14730   94565  :  4595
##  Alameda County     :60410   Santa Rosa   :  9917   94509  :  4302
##  Contra Costa County:59381   Fremont      :  9414   95123  :  4023
##  Solano County      :23404   San Francisco:  8137   95687  :  3652
##  San Mateo County   :22558   Evergreen    :  7947   94533  :  3472
##  Sonoma County      :21676   Antioch      :  7726   (Other):261457
##  (Other)            :23653   (Other)      :223635   NA's   :     5
##     street              price                br            lsqft
##  Length:281506      Min.   :   22000   Min.   :1.000   Min.   :       19
##  Class :character   1st Qu.:  400000   1st Qu.:2.000   1st Qu.:     4000
##  Mode  :character   Median :  530000   Median :3.000   Median :     5760
##                     Mean   :  602000   Mean   :3.024   Mean   :    65939
##                     3rd Qu.:  700000   3rd Qu.:4.000   3rd Qu.:     7701
##                     Max.   :20000000   Max.   :8.000   Max.   :418611600
##                                                        NA's   :21687
```

```
##      bsqft              year           date
##  Min.   :    122   Min.   :   0   Min.   :2003-04-27 02:00:00
##  1st Qu.:   1121   1st Qu.:1954   1st Qu.:2004-02-08 02:00:00
##  Median :   1430   Median :1971   Median :2004-10-24 02:00:00
##  Mean   :   1624   Mean   :1966   Mean   :2004-11-01 18:06:12
##  3rd Qu.:   1882   3rd Qu.:1985   3rd Qu.:2005-07-24 02:00:00
##  Max.   :1868120   Max.   :3894   Max.   :2006-06-04 02:00:00
##  NA's   :426       NA's   :9202
##       long             lat
##  Min.   :-123.6   Min.   :36.98
##  1st Qu.:-122.3   1st Qu.:37.50
##  Median :-122.1   Median :37.77
##  Mean   :-122.1   Mean   :37.78
##  3rd Qu.:-121.9   3rd Qu.:38.00
##  Max.   :-121.5   Max.   :38.85
##  NA's   :23316    NA's   :23316
##                                          quality
##  QUALITY_ADDRESS_RANGE_INTERPOLATION      :170719
##  gpsvisualizer                            : 31084
##  QUALITY_CITY_CENTROID                    : 20473
##  QUALITY_EXACT_PARCEL_CENTROID            : 17208
##  QUALITY_ZIP_CODE_TABULATION_AREA_CENTROID: 14980
##  (Other)                                  :  3726
##  NA's                                     : 23316
##                match              wk
##  Exact            :197044   Min.   :2003-04-21
##  Relaxed          : 30570   1st Qu.:2004-02-01
##  Relaxed; Soundex: 23338   Median :2004-10-18
##  Soundex          :  2573   Mean   :2004-10-26
##  1                :  2244   3rd Qu.:2005-07-18
##  (Other)          :  2421   Max.   :2006-05-29
##  NA's             : 23316
```

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

Write your response here

```
print("1.The maximum price seems to be too big")
```

```
## [1] "1.The maximum price seems to be too big"
```

```
print("2.Lots of data are missing. There's a lot of NA's in the dataset")
```

```
## [1] "2.Lots of data are missing. There's a lot of NA's in the dataset"
```

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.

```
new_housing = subset(housing,housing$city %in% c("Albany", "Berkeley",
"Piedmont", "Emeryville"))
BerkArea = new_housing[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.

```
BerkArea = subset(BerkArea, BerkArea$price < quantile(BerkArea$price, 0.99, na.rm = TRUE) & BerkArea$bsqft < quantile(BerkArea$bsqft, 0.99, na.rm = TRUE))
```

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.

```
BerkArea["pricepsqft"] = BerkArea$price/BerkArea$bsqft
```

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 = BerkArea$br
br5[br5 >5] <- 5
```

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.

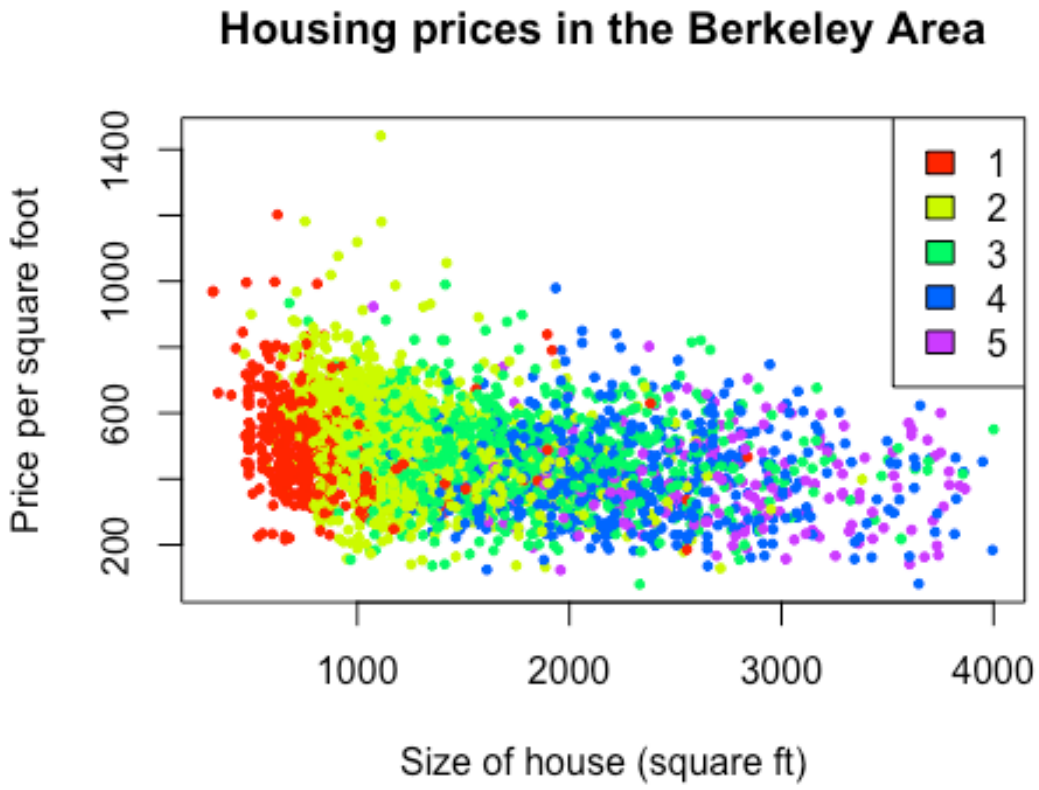
(2 pts.)

```
rCols = c(rainbow(5, s = 1, v = 1, start = 0, end = max(1, 5 - 1)/5, alpha =
1))
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?

```
print("The size of houses is proportional to # of br. As we can see on the
graph the colors are actually in blocks.")
```

```
## [1] "The size of houses is proportional to # of br. As we can see on the
graph the colors are actually in blocks."
```

(2 pts.)

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

```
boxplot(c(BerkArea$price) ~ as.character(BerkArea$city),las = 2, main =
"housing price in 4 cities")
```