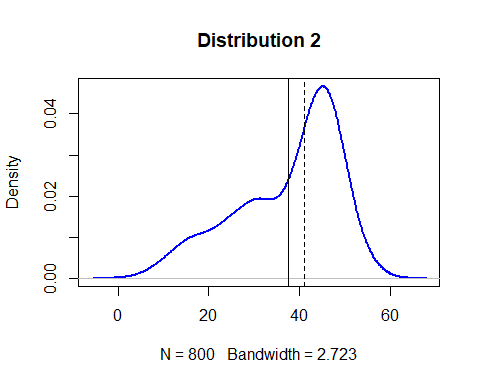
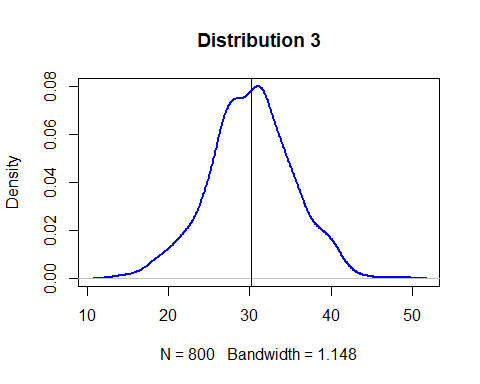
BACS-hw2.R

2022-04-11

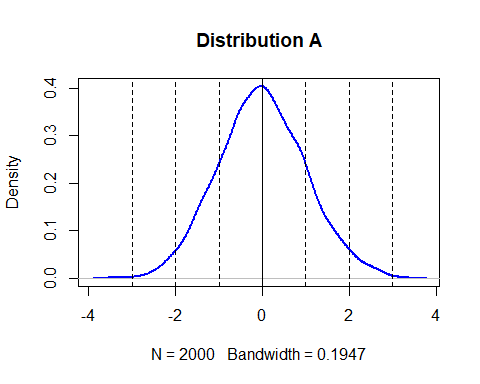
#Question 1  
#a  
# Three normally distributed data sets  
d1 <- rnorm(n=100, mean=15, sd=5)  
d2 <- rnorm(n=200, mean=30, sd=5)  
d3 <- rnorm(n=500, mean=45, sd=5)  
# Combining them into a composite dataset  
d123 <- c(d1, d2, d3)  
# Let’s plot the density function of d123  
plot(density(d123), col="blue", lwd=2, main = "Distribution 2")  
# Add vertical lines showing mean and median  
abline(v=mean(d123))  
abline(v=median(d123), lty="dashed")



#b  
d4 <- rnorm(800, mean=30, sd=5)  
plot(density(d4), col="blue", lwd=2, main = "Distribution 3")  
abline(v=mean(d4))  
abline(v=median(d4), lty="dashed")

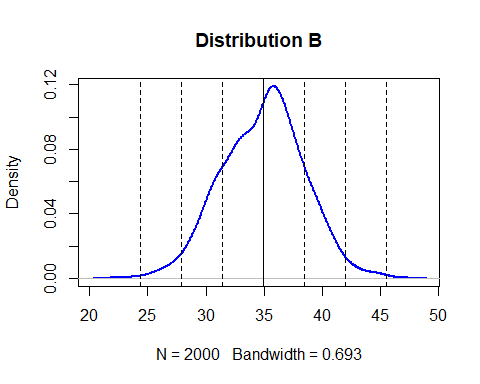


#c  
mean. Mean values include all data; hence, the value will change when there are extreme outliers. However, median value only calculates the middle number.  
  
#Question 2  
quartiles\_vs\_sd <- function(distr, title){  
 #plot data distribution, mean + standard deviations lines  
 plot(density(distr), col="blue", lwd=2, main = title)  
 abline(v=mean(distr))  
 abline(v=mean(distr)+(-3:3)\*sd(distr), lty="dashed")  
   
 #return the distance of each quartile from the mean  
 q = quantile(distr, c(0.25, 0.5, 0.75))  
 return((q-mean(distr))/sd(distr))  
}  
#a,b  
rdata <- rnorm(2000, 0, 1)  
quartiles\_vs\_sd(rdata, "Distribution A")



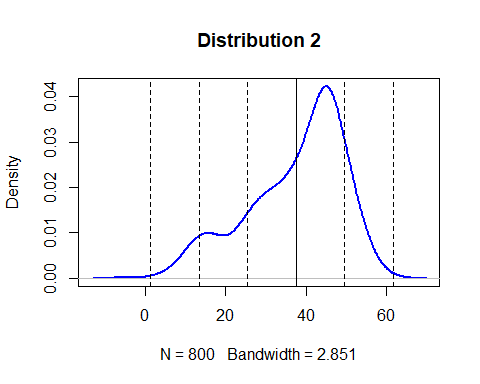
## 25% 50% 75%   
## -0.65747338 -0.01593631 0.67113705

#C  
rdata1 <- rnorm(2000, 35, 3.5)  
quartiles\_vs\_sd(rdata1, "Distribution B")



## 25% 50% 75%   
## -0.69864131 0.06553258 0.66734491

#d  
d1 <- rnorm(n=100, mean=15, sd=5)  
d2 <- rnorm(n=200, mean=30, sd=5)  
d3 <- rnorm(n=500, mean=45, sd=5)  
d123 <- c(d1, d2, d3)  
quartiles\_vs\_sd(d123, "Distribution 2")



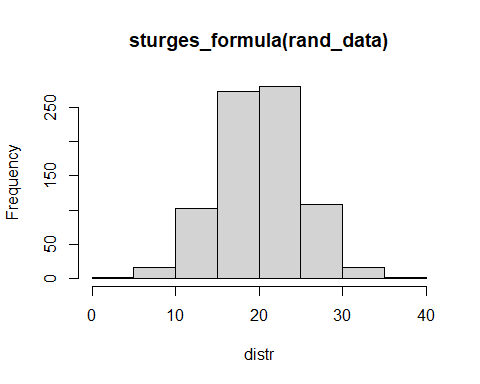
## 25% 50% 75%   
## -0.6142023 0.2586607 0.7381402

#Question 3  
#a  
h=2\*IQR\*n^(1/3)

k=(max-min)/h

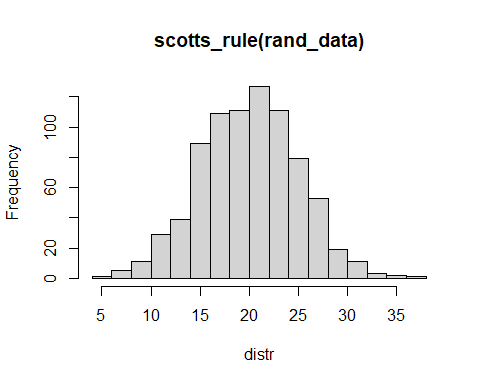
It replaces 3.5sd of Scott’s rule with 2 IQR, which is less sensitive than the standard deviation to outliers in data

#b  
#Sturges' Rule: Log length for number of bins  
sturges\_formula <- function(distr, title) {  
 k = ceiling(log2(length(distr))) + 1  
 h = (max(distr) - min(distr))/k  
 hist(distr, breaks = k, main=title)  
 return(data.frame(k, h))  
}  
  
#Scott’s Rule: Standard deviation for bin size  
scotts\_rule <- function(distr, title) {  
 h <- 3.5\*sd(distr) / (length(distr)^(1/3))  
 k = ceiling((max(distr) - min(distr))/h)  
 hist(distr, breaks = k, main=title)  
 return(data.frame(k, h))  
}  
  
#Freedman-Diaconis Choice: IQR for bin size  
fd\_choice <- function(distr, title) {  
 h <- 2\*IQR(distr) / (length(distr)^(1/3))  
 k <- ceiling((max(distr) - min(distr))/h)  
 hist(distr, breaks = k, main=title)  
 return(data.frame(k, h))  
}  
  
rand\_data <- rnorm(800, mean=20, sd=5)  
sturges\_formula(rand\_data, "sturges\_formula(rand\_data)")



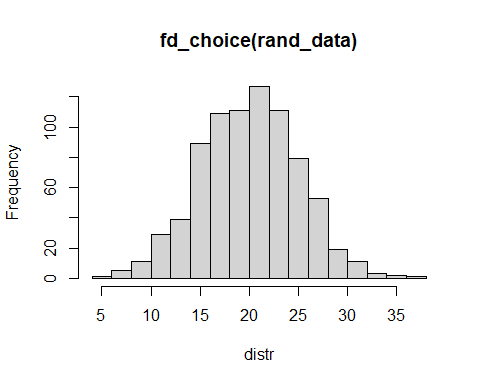
## k h  
## 1 11 2.958156

scotts\_rule(rand\_data, "scotts\_rule(rand\_data)")



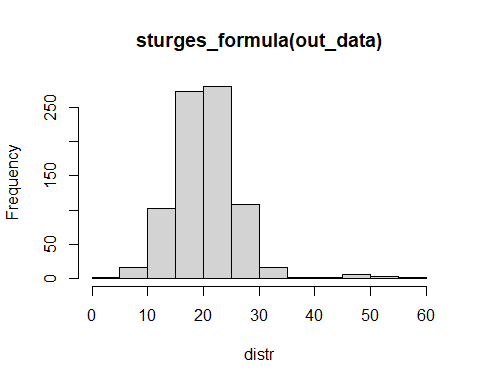
## k h  
## 1 18 1.863892

fd\_choice(rand\_data, "fd\_choice(rand\_data)")



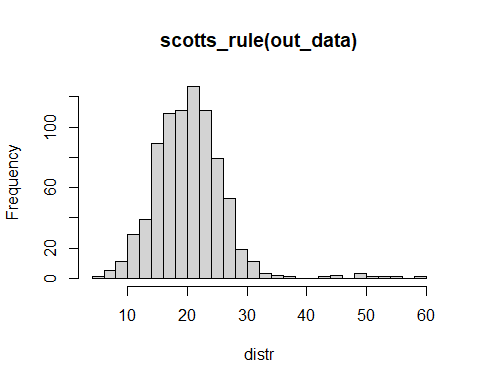
## k h  
## 1 23 1.470768

#c  
out\_data <- c(rand\_data, runif(10, min=40, max=60))  
sturges\_formula(out\_data, "sturges\_formula(out\_data)")



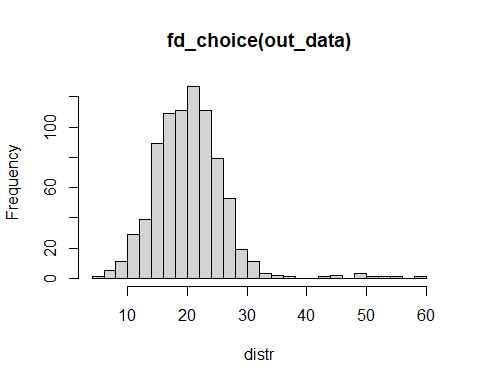
## k h  
## 1 11 4.91102

scotts\_rule(out\_data, "scotts\_rule(out\_data)")



## k h  
## 1 25 2.230429

fd\_choice(out\_data, "fd\_choice(out\_data)")



## k h  
## 1 37 1.479007

Freedman-Diaconis’s choice change the least. Using quartiles (IQR) makes bin size (h) insensitive to the presence of outliers