Stat 361 - hw5

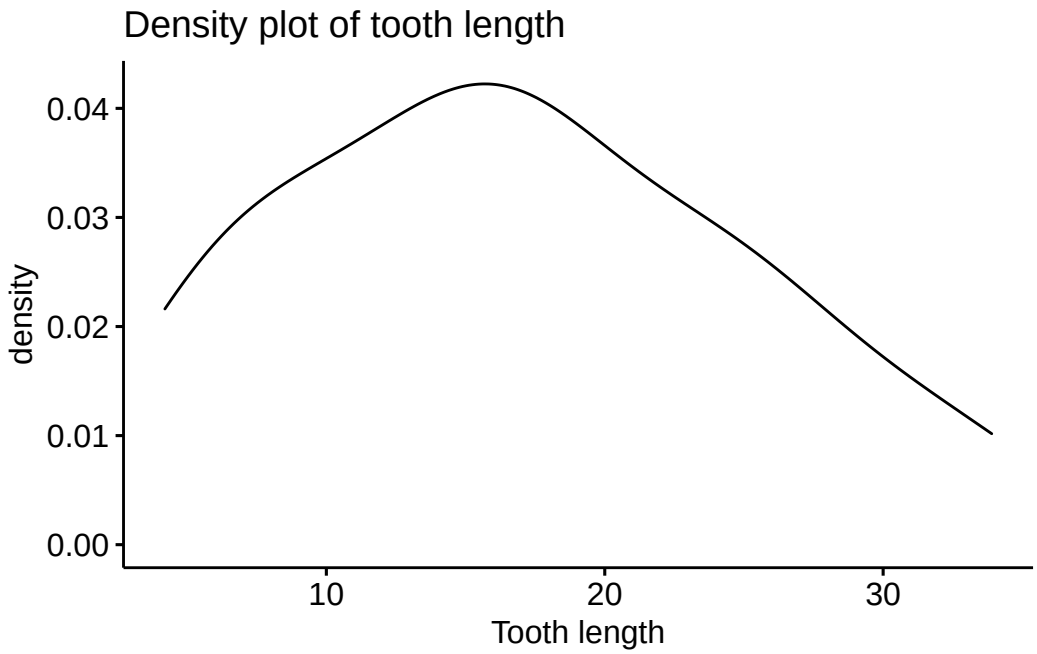
Mert Göksel Bilge Özkır

Q1

```
df <- sample(ToothGrowth)
x <- df %>% filter(supp == "VC") %>% select(len) %>% unlist()
```

Part A

```
#For the Sturges Rule, we need to check whether is data sampled from symmetric,
#unimodal populations.

library("ggpubr")
ggdensity(x, main = "Density plot of tooth length",
          xlab = "Tooth length") #Our data is symmetric, so it is unimodal.
```




```
#But, when we check the mode of our data;
find_mode <- function(x) {
  u <- unique(x)
  tab <- tabulate(match(x, u))
  u[tab == max(tab)]
}

find_mode(x) # Sturges Rule's not a good choice for the distribution
```

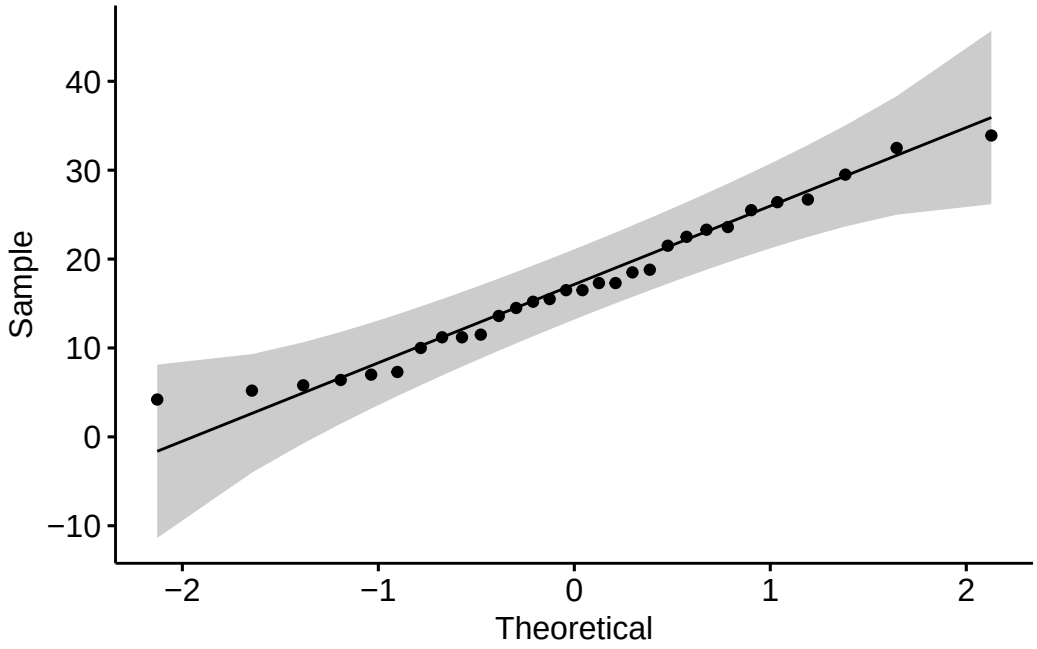
```
[1] 11.2 16.5 17.3
```

```
#with more than one mode.

#And we found 3 different modes.

# On the other hand, Scott's normal reference rule is optimal for random
#samples of normally distributed data, in the sense that it minimizes the
#integrated mean squared error of the density estimate. When we check the data
#for normality;
```

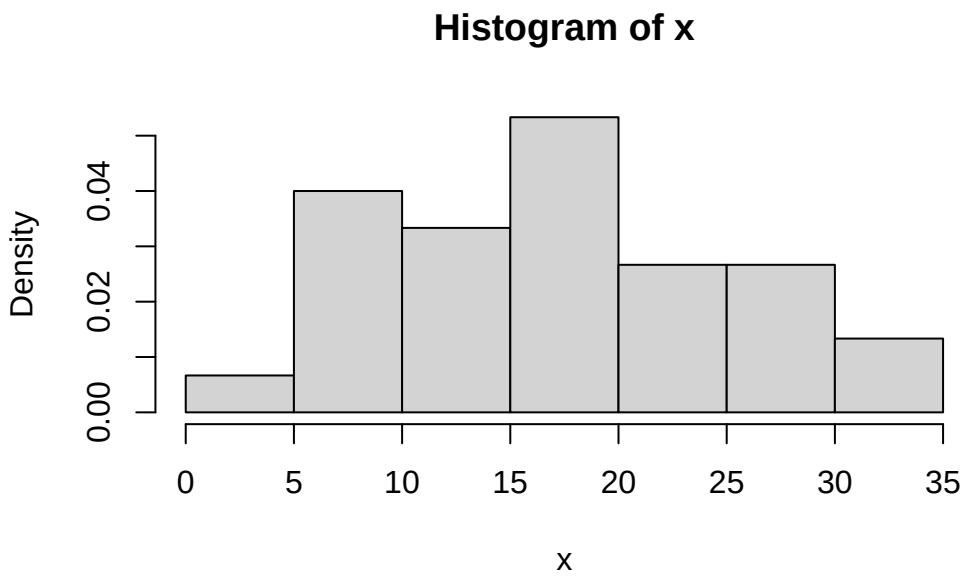
```r
shapiro.test(x) # Our p value >0.05, so our data is normally distributed.
```

```
    Shapiro-Wilk normality test

data:  x
W = 0.96567, p-value = 0.4284
```

```r
ggqqplot(x) #We can see in the Q-q plot this normality.
```



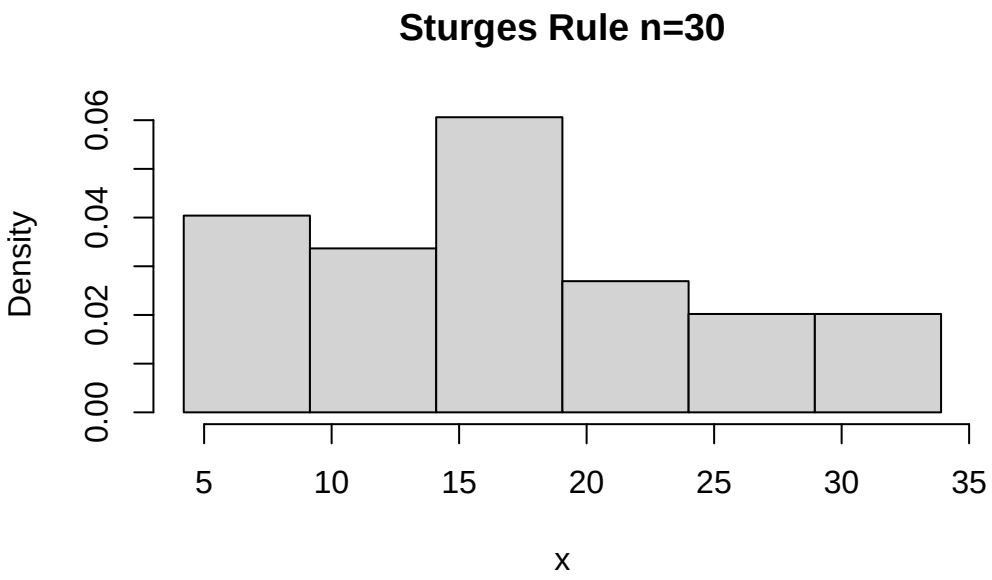

```r
#For the Freedman - Diaconis, similar to Scott's rule
#because distribution symmetric.
hist(x, breaks = 6, probability = T)
```




```
#Finally, other rules conditions also provided. Methods for density
#estimation is suitable for tooth length sample except for Sturges Rule.
```

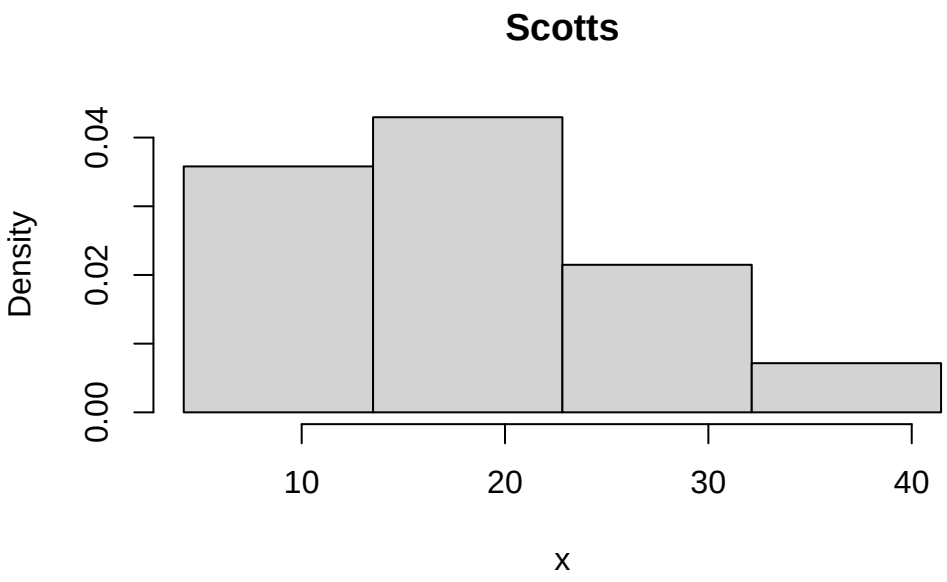
Part B

```
sturges <- function(x){
    n <- length(x)
    nclass <- ceiling(1 + log2(n))
    cwidth <- diff(range(x) / nclass)
    breaks <- min(x) + cwidth * 0:nclass
    return(list(nclass = nclass, cwidth = cwidth, breaks = breaks))
}

z1 <- seq(min(x) - sturges(x)$cwidth,
          max(x) + sturges(x)$cwidth, 0.01)

h.sturges <- hist(x, breaks = sturges(x)$breaks,
                  prob = T, main = "Sturges Rule n=30")
```



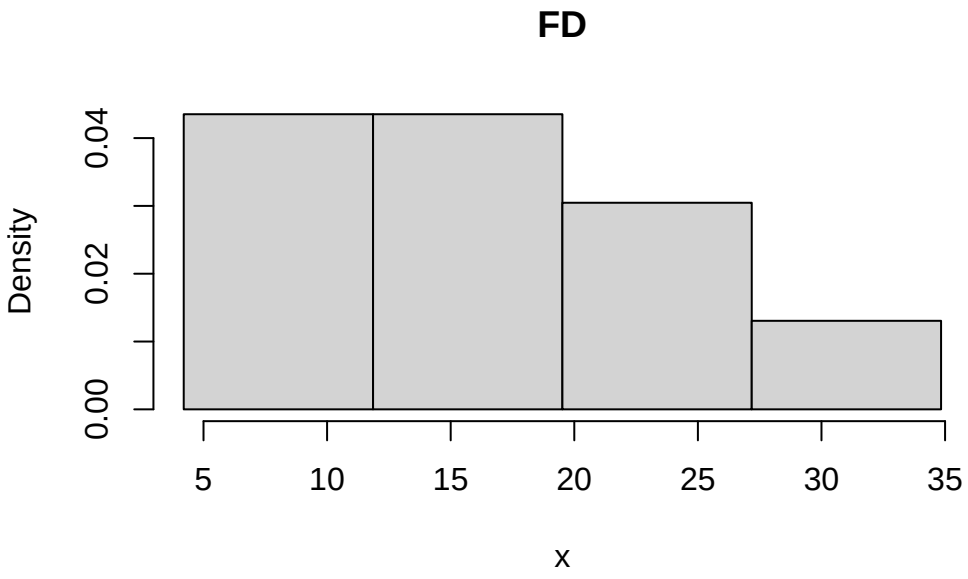
Part C

```
scotts <- function(x){
    n <- length(x)
    h <- 3.5 * sd(x) * n^(-1/3)
    nclass <- ceiling(diff(range(x)) / h)
    breaks <- min(x) + h * 0:nclass
    return(list(nclass = nclass, h = h, breaks = breaks))
}

h.scotts <- hist(x, breaks = scotts(x)$breaks,
                 prob = T, main = "Scotts")
```



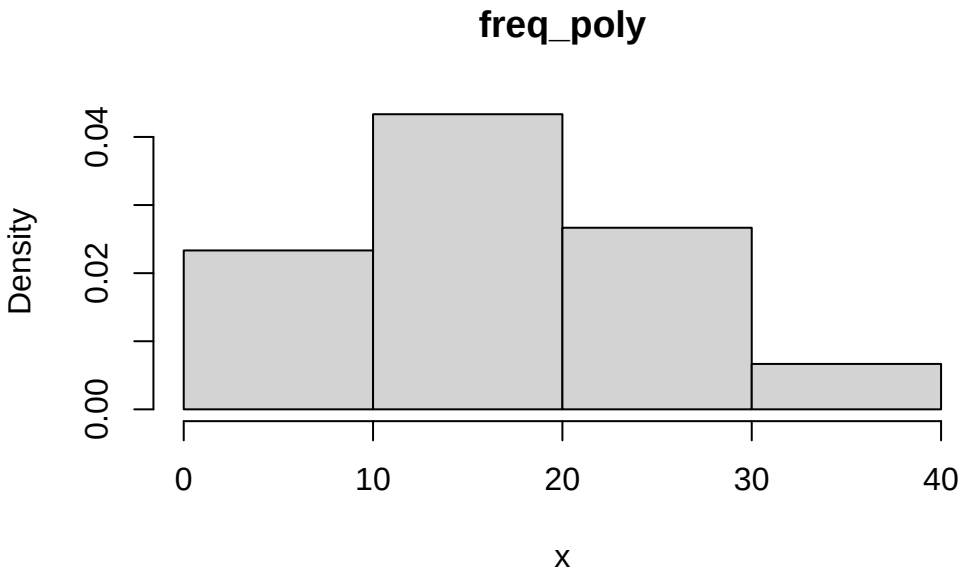
Part D

```
FD <- function(x){
    n <- length(x)
    h <- 2 * IQR(x) * n^(-1/3)
    nclass <- ceiling(diff(range(x)) / h)
    breaks <- min(x) + h * 0:nclass
    return(list(nclass = nclass, h = h, breaks = breaks))
}

h.df <- hist(x, breaks = FD(x)$breaks,
                  prob = T, main = "FD")
```



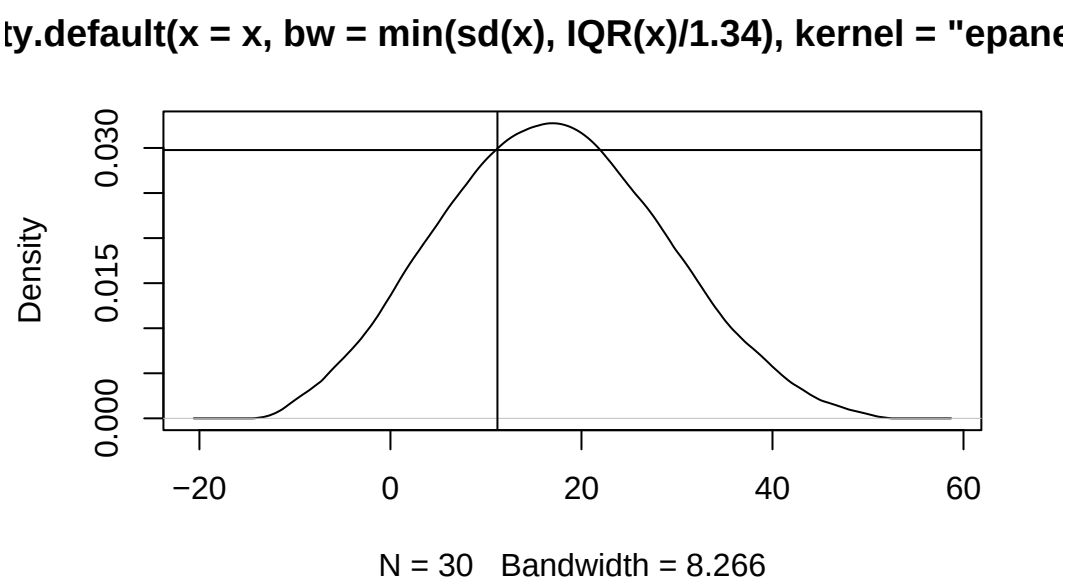
Part E

```
freq_poly <- function(x){
    n <- length(x)
    h <- 2.15 * sd(x) * n^(-1/5)
    br <- pretty(x, diff(range(x)) / h)
    brplus <- c(min(br)-h, max(br)+h)
    return(list(brplus = brplus, h = h, breaks = br))
}

h.freqpoly <- hist(x, breaks = freq_poly(x)$breaks,
                   prob = T, main = "freq_poly")
```

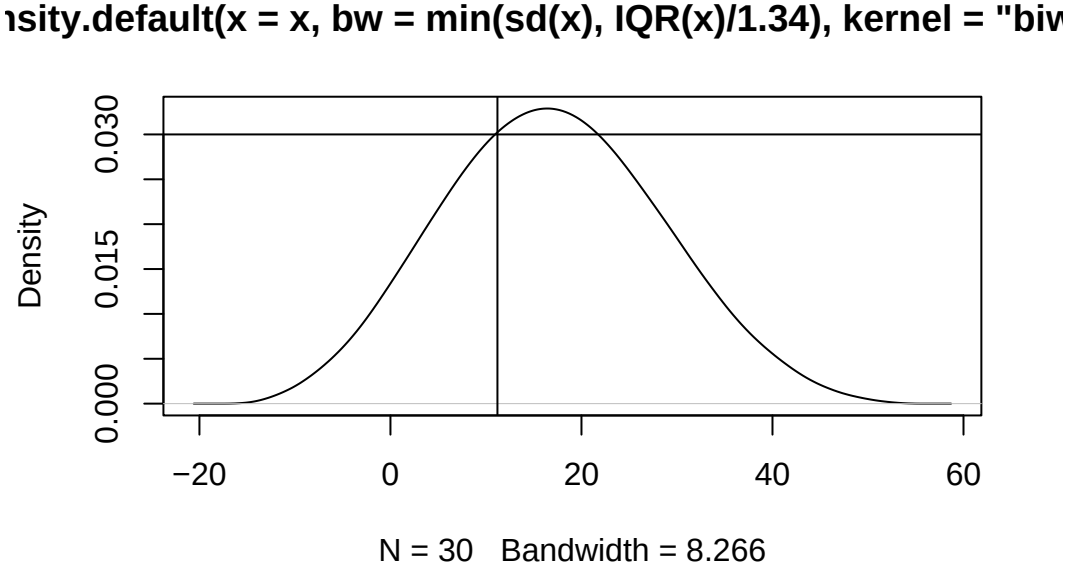


Part F

```
plot(density(x, bw=min(sd(x), IQR(x)/1.34), kernel='epanechnikov'))
abline(v=11.2, h=0.02977) #approx 0.02977
```

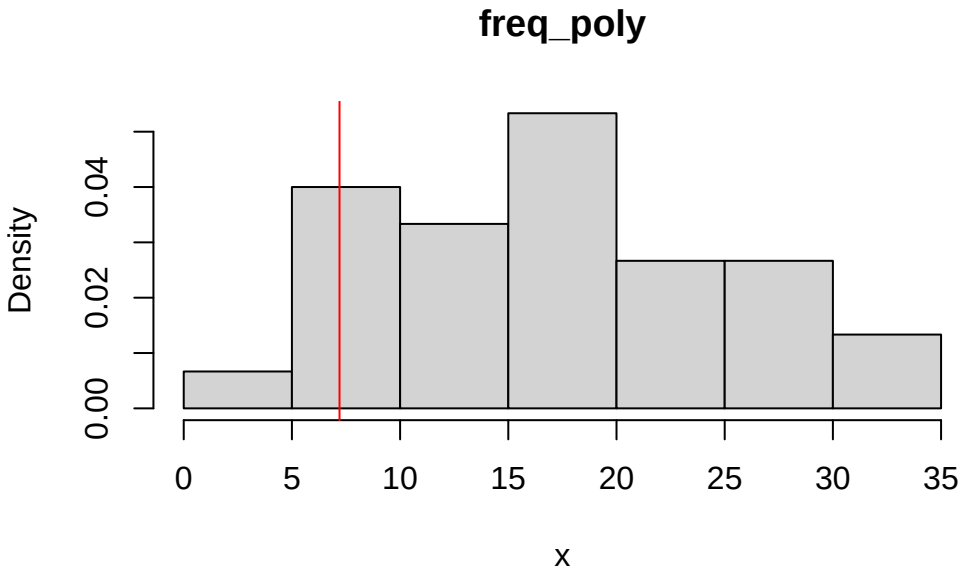



```
plot(density(x, bw=min(sd(x), IQR(x)/1.34), kernel='biweight'))
abline(v=11.2, h=0.03) # approx 0.03
```

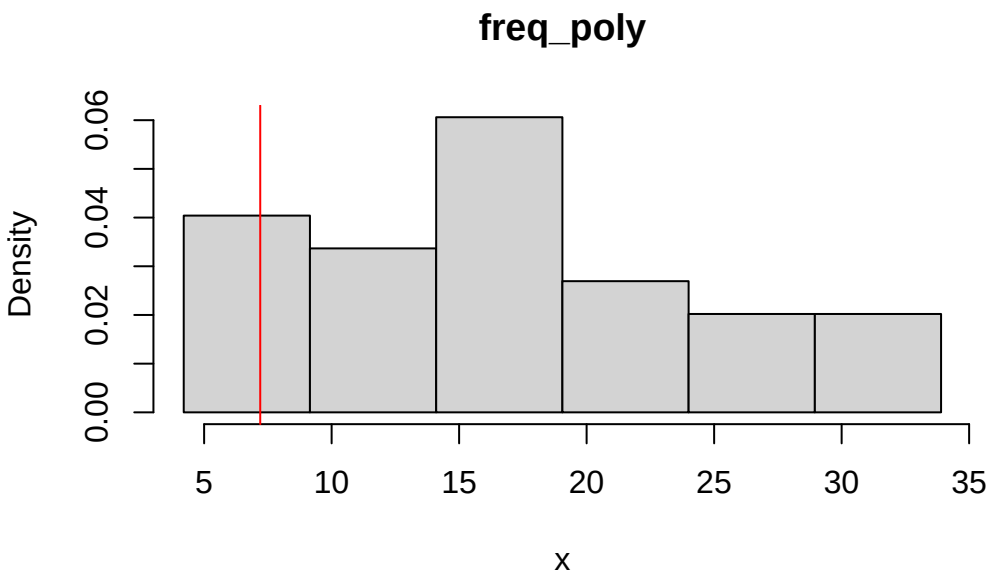


Part G

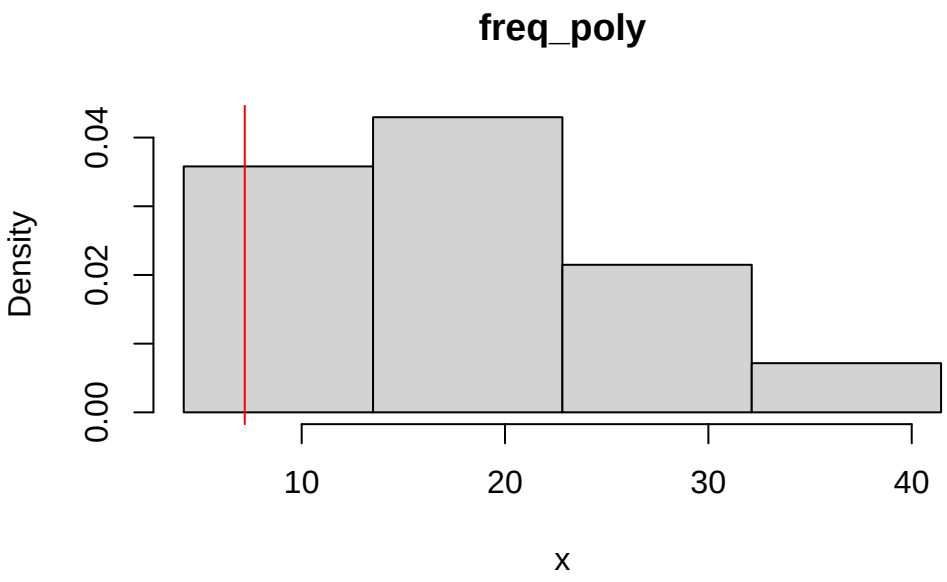
```
hist(x, prob = T, main = "freq_poly")
abline(v=7.2, col='red') #2nd
```



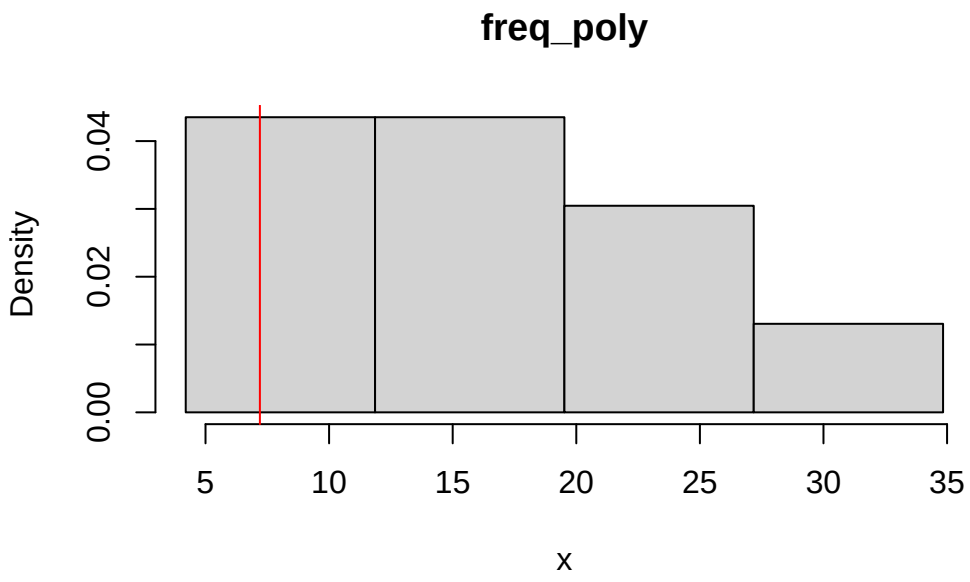

```
hist(x, breaks = sturges(x)$breaks,
                  prob = T, main = "freq_poly")
abline(v=7.2, col='red') #1st
```




```
hist(x, breaks = scotts(x)$breaks,
                  prob = T, main = "freq_poly")
abline(v=7.2, col='red') #1st
```




```
hist(x, breaks = FD(x)$breaks,
                  prob = T, main = "freq_poly")
abline(v=7.2, col='red') #1st
```

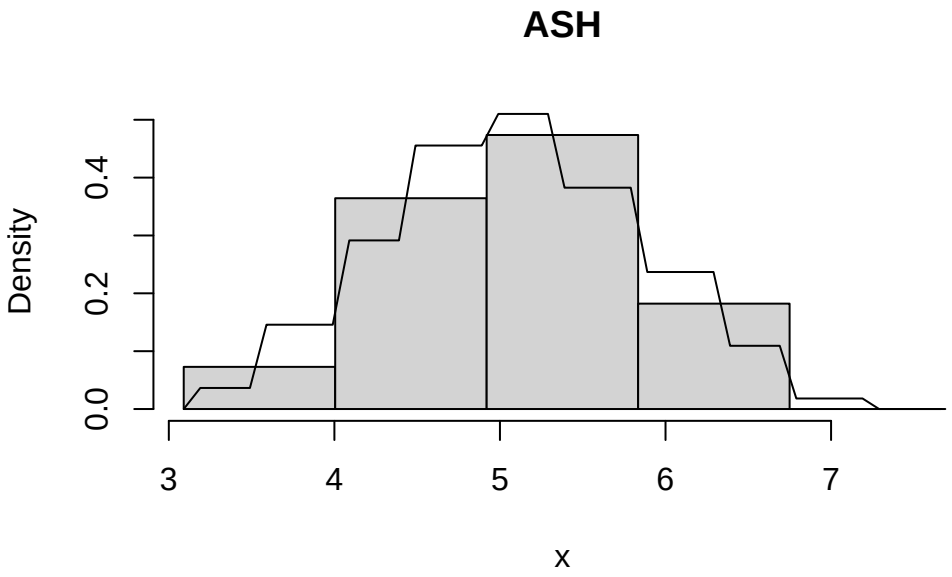


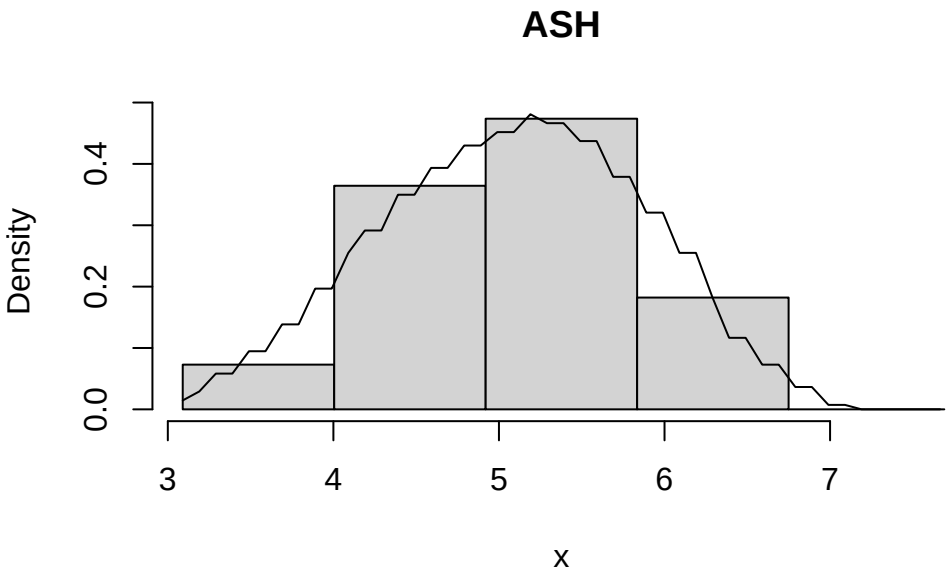
Q2

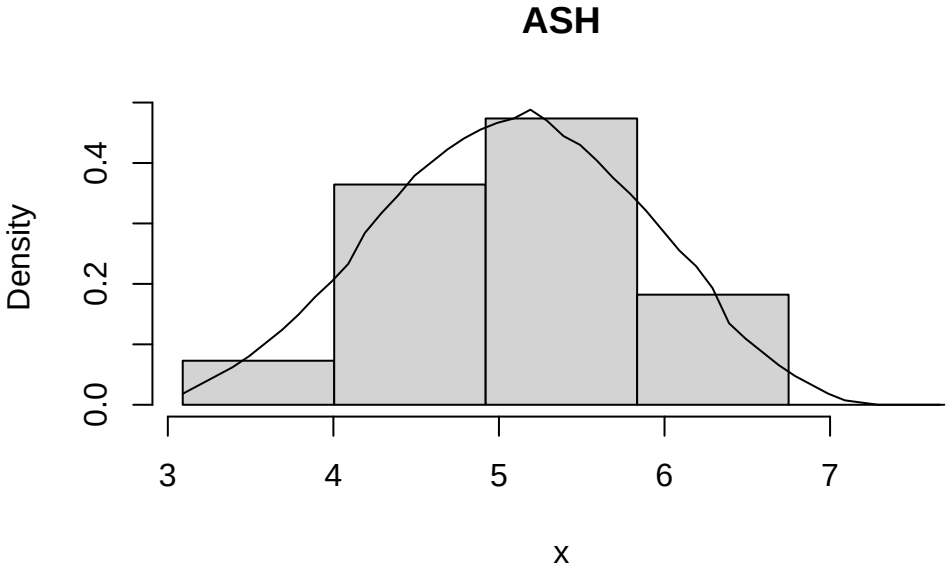
Part A

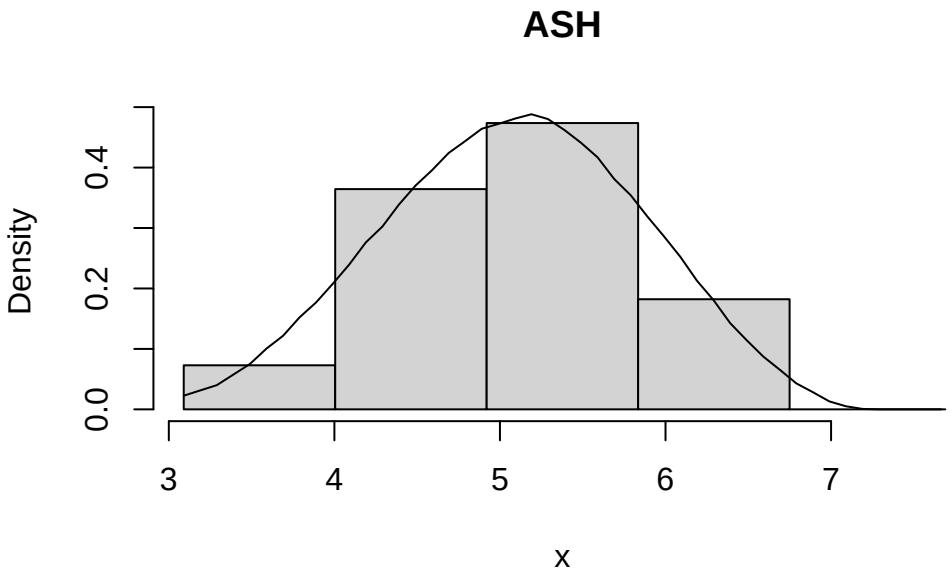
```
x <- PlantGrowth$weight
n <- length(x)
h <- 2.576 * sd(x) * n^(-1/5) #bin with formula
a <- min(x) - 0.5
b <- max(x) + 0.5
m <- c(2,5,10,50,100)

fhat <- function(x,j){
      i <- max(which(x > breaks))
      k <- (i - m[j] + 1):(i + m[j] - 1)
      vk <- nk[k]
      sum((1 - K / m[j]) * vk) / (n * h)
}

for(i in 1:length(m)){
  delta <- h / m[i]
```

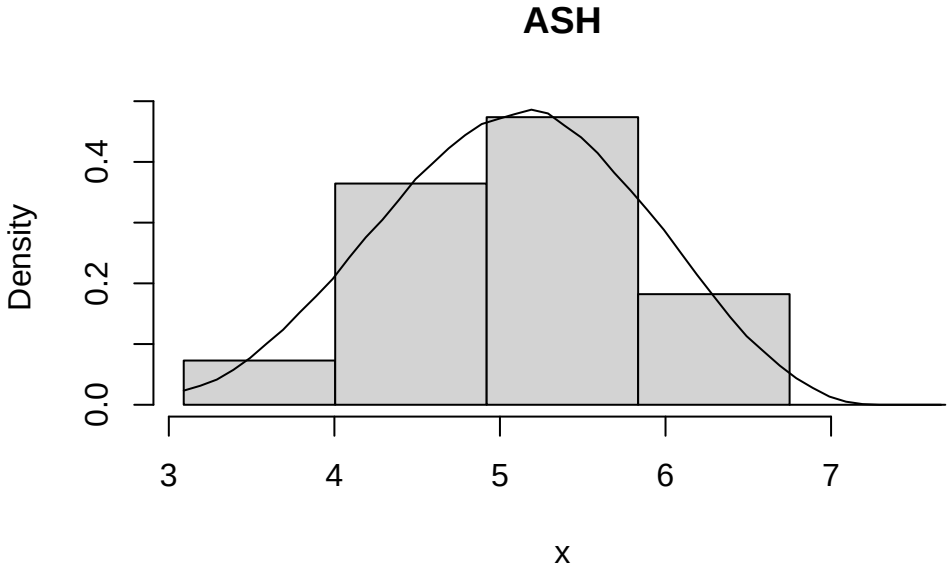
```r
  breaks <- seq(a - h, b + 2*h, delta)
  hist.ash <- hist(x, breaks = breaks, plot = FALSE)
  nk <- hist.ash$counts
  K <- abs((1-m[i]):(m[i]-1))
  z_ash <- as.matrix(seq(a, b + h, 0.1))
  f.ash <- apply(z_ash, 1, fhat, j=i)
  breaks2 <- seq(a, b + h, h)
  hist(x, breaks = breaks2, freq = FALSE, main = "ASH", ylim = c(0, max(f.ash)))
  lines(z_ash, f.ash, xlab = "x")
}
```




ASH
Density
0.0
0.2
0.4
3
4
5
6
7
x


ASH
Density
0.0
0.2
0.4
3
4
5
6
7
x


ASH
Density
0.0
0.2
0.4
3
4
5
6
7
x


ASH
Density
0.0
0.2
0.4
3
4
5
6
7
x

Part B

```
n <- length(x)
h <- 2.576 * sd(x) * n^(-1/5) #bin with formula
a <- min(x) - 0.5
b <- max(x) + 0.5
m <- 2
delta <- h / m
breaks <- seq(a - h, b + 2*h, delta)
hist.ash <- hist(x, breaks = breaks, plot = FALSE)
nk <- hist.ash$counts
K <- abs((1-m):(m-1))

fhat <- function(x){
  m <- 2
  i <- max(which(x > breaks))
  k <- (i - m + 1):(i + m - 1)
  vk <- nk[k]
  sum((1 - K / m) * vk) / (n * h)
}

x <- c(4.1,3.5,6.1,4.9,5.7)
for(i in x){
  print(paste("For x0 =", i, ", Our density estimate =", fhat(i)))
}
```

```
[1] "For x0 = 4.1 , Our density estimate = 0.291481012576231"
[1] "For x0 = 3.5 , Our density estimate = 0.0364351265720289"
[1] "For x0 = 6.1 , Our density estimate = 0.236828322718188"
[1] "For x0 = 4.9 , Our density estimate = 0.455439082150361"
[1] "For x0 = 5.7 , Our density estimate = 0.382568829006303"
```