Stat 361 - Recitation 11

Probability Density Estimation

Orçun Oltulu

14 - 15 / 05 / 2020

Probability Density Estimation

Density estimation is a collection of methods for constructing an estimate of a probability density, as a function of an observed sample of data. Namely, we observe $X_1, ..., X_n$ and we want to recover the underlying probability density function generating our dataset.

A density estimation problem requires a nonparametric approach if we have no information about the target distribution other than the observed data.

It is unlikely that the probability density function for a random sample of data is known. As such, the probability density must be approximated using a process known as probability density estimation.

It is useful to know the probability density function for a sample of data in order to know whether a given observation is unlikely, or so unlikely as to be considered an outlier or anomaly and whether it should be removed. It is also helpful in order to choose appropriate learning methods that require input data to have a specific probability distribution.

Probability Density Estimation comes handy in many outlier detection algorithms where we seek to estimate "true" distribution based on sample observations and then classify some of existing or new observations as outlier or not.

Univariate Density Estimation

Histograms:

Histogram plots provide a fast and reliable way to visualize the probability density of a data sample.

A histogram is a plot that involves first grouping the observations into bins and counting the number of events that fall into each bin. The counts, or frequencies of observations, in each bin are then plotted as a bar graph with the bins on the x-axis and the frequency on the y-axis.

The choice of the number of bins is important as it controls the coarseness of the distribution (number of bars) and, in turn, how well the density of the observations is plotted. It is a good idea to experiment with different bin sizes for a given data sample to get multiple perspectives or views on the same data.

We split the sample space up into bins, count how many samples fall into each bin, and then divide the counts by the total number of samples.

If we hold the bins fixed and take more and more data, then by the law of large numbers we anticipate that the relative frequency for each bin will converge on the bin's probability.

Sturges' Rule:

Although Sturges' rule tends to oversmooth the data and either Scott's rule or FD are generally preferable, Sturges' rule is the default in many statistical packages.

Please run this code; $'?hist()'$, you will see that break option is set to 'Sturges' as default.

According to Sturges, the optimal width of class intervals is given by

$$\frac{R}{1+\log_2(n)}, \qquad \text{where R is the sample range}$$

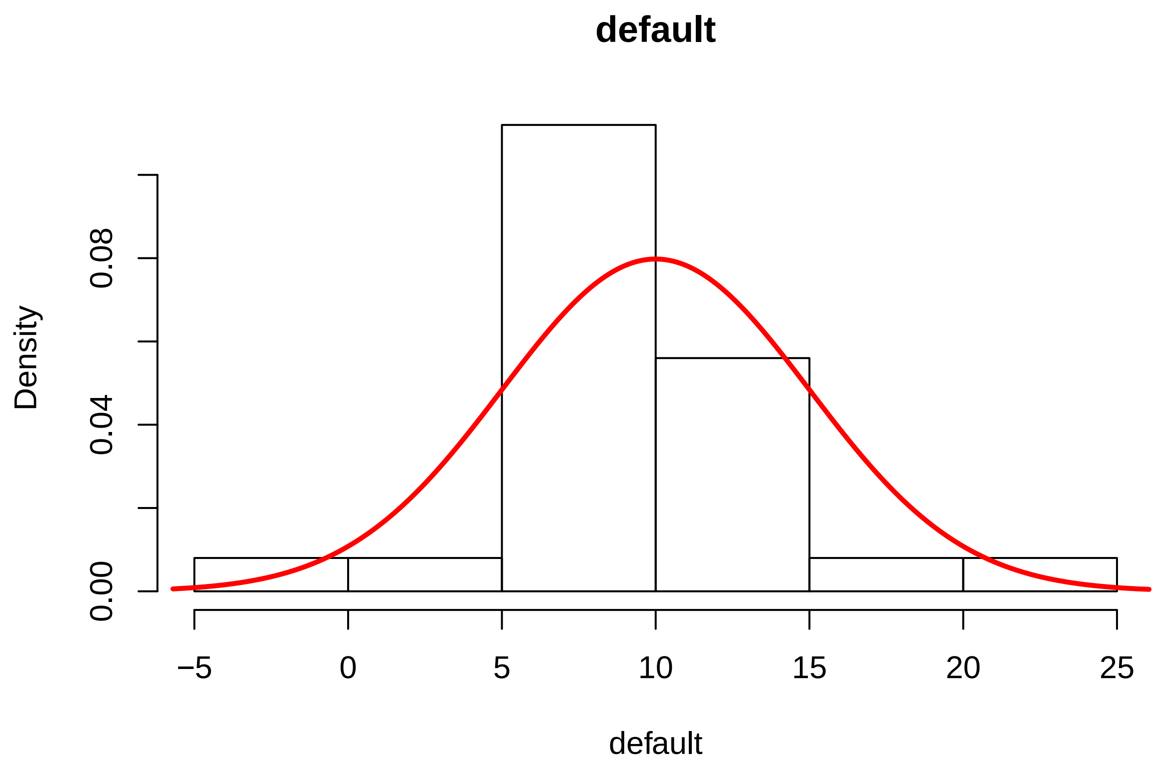
Note that This choice of class interval is designed for data sampled from symmetric, unimodal populations, but is not a good choice for skewed distributions or distributions with more than one mode.

Question 1:

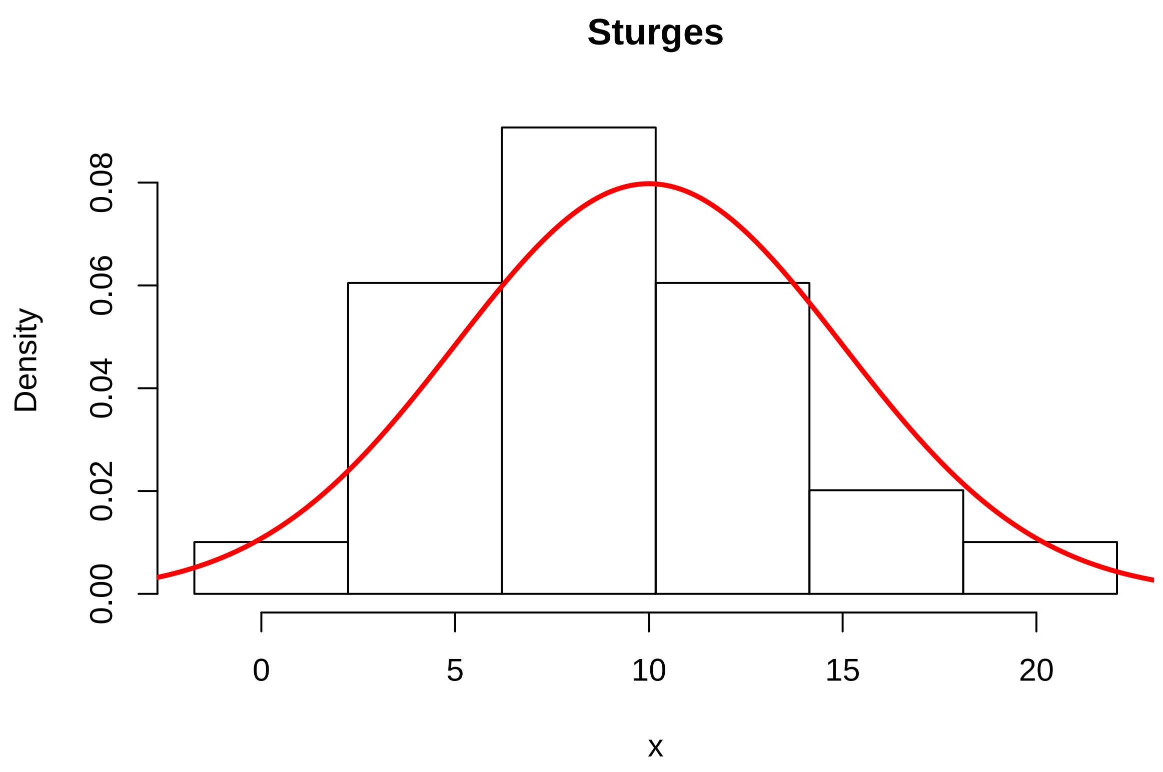
Generate random sample of size n = 25 from Normal(10,25) and using Sturges' Rule plot a histogram and comment on the histogram.

```r
# Sturges' Rule

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))
}


set.seed(1234)
x <- rnorm(25, 10, 5)
```

```r
z <- seq(min(x) - sturges(x)$cwidth, max(x) +  sturges(x)$cwidth, 0.01)

h.default <- hist(x, freq = FALSE,
                  xlab = "default", main = "default")
lines(z, dnorm(z, 10, 5),col = "red", lwd = 2.5)
```



```r
#sturges'
h.sturges <- hist(x, breaks = sturges(x)$breaks,
                  prob = TRUE, main = "Sturges")
lines(z, dnorm(z, 10, 5),col = "red", lwd = 2.5)
```



```r
# breaks
h.default$breaks
```

```
## [1] -5  0  5 10 15 20 25
```

```r
round(h.sturges$breaks,1)
```

```
## [1] -1.7  2.2  6.2 10.2 14.1 18.1 22.1
```

```r
# counts
h.default$counts
```

```
## [1]  1  1 14  7  1  1
```

```r
h.sturges$counts
```

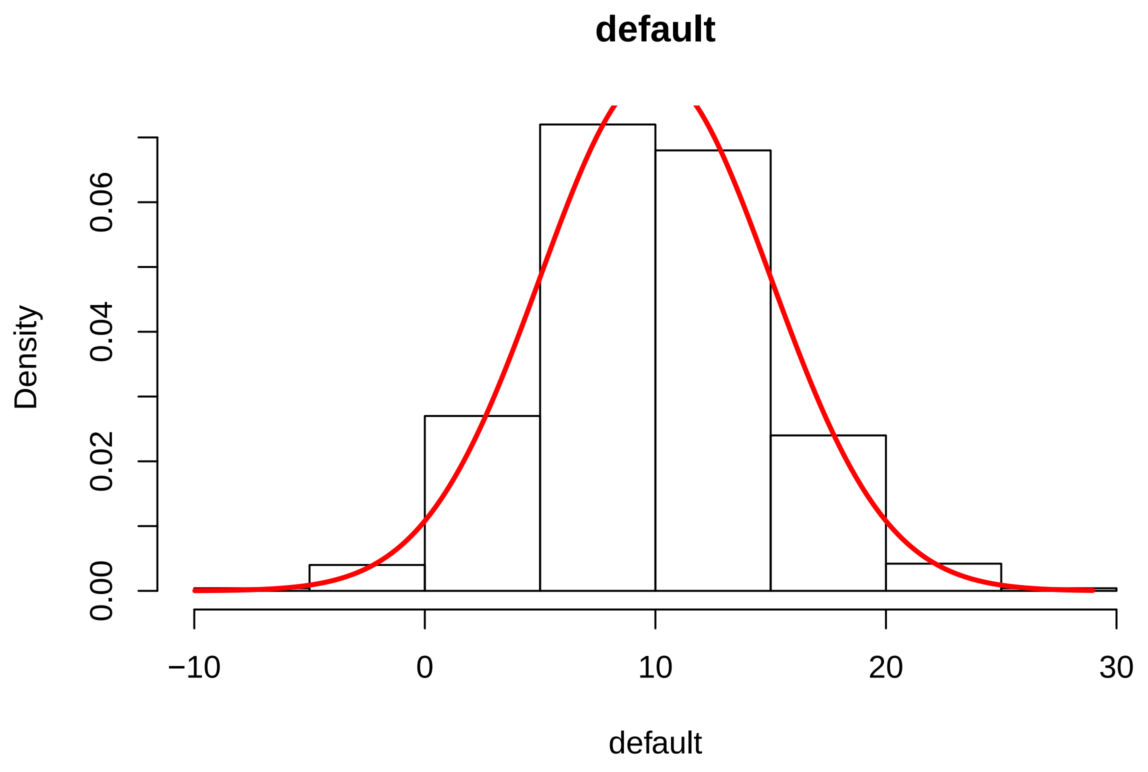
```
## [1] 1 6 9 6 2 1
```

```r
sturges(x)$cwidth
```

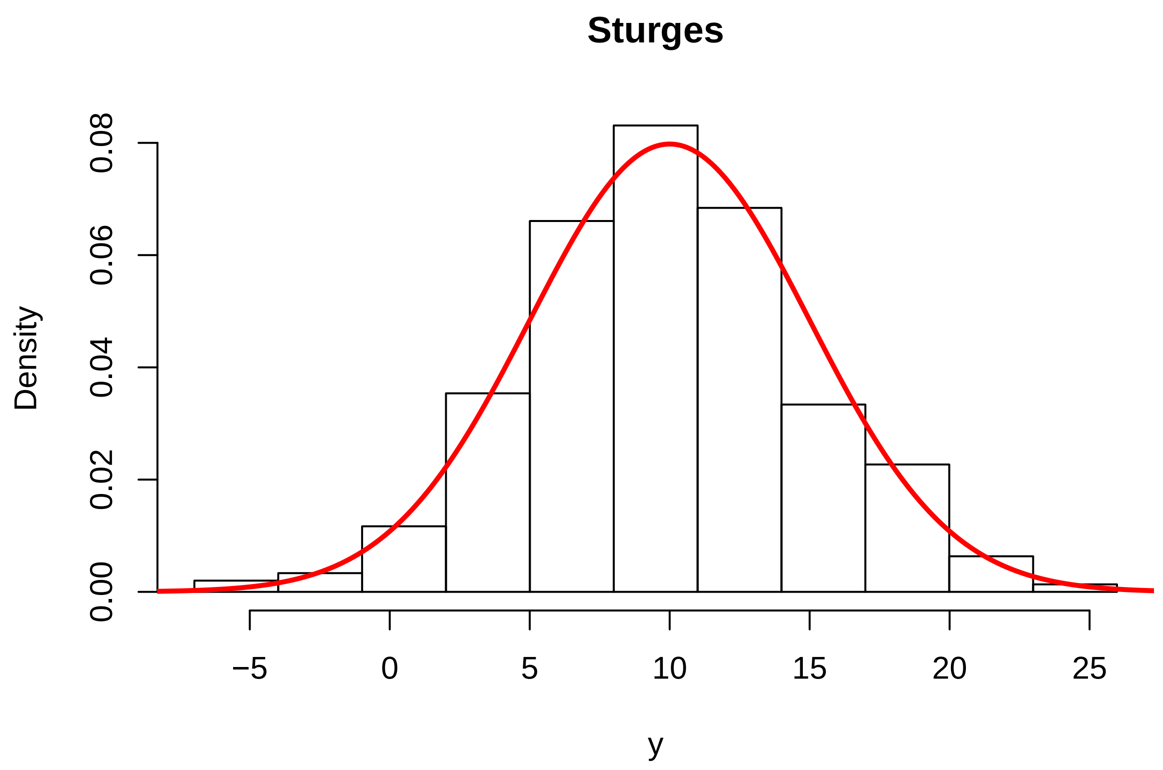
```
## [1] 3.967944
```

Now, this time increase the sample size to 1000, you must see that the shape looks exactly symmetric bell shape.

```
set.seed(1234)
y <- rnorm(1000, 10, 5)

z <- seq(min(y) - sturges(y)$cwidth, max(y) +  sturges(y)$cwidth, 0.01)

h.default2 <- hist(y, freq = FALSE,
                   xlab = "default", main = "default")
lines(z, dnorm(z, 10, 5),col = "red", lwd = 2.5)
```



```
#sturges'
h.sturges2 <- hist(y, breaks = sturges(y)$breaks,
                   prob = TRUE, main = "Sturges")
lines(z, dnorm(z, 10, 5),col = "red", lwd = 2.5)
```



Now, let $x_0 = 12$, locate the bin containing the point x_0 and then compute the relative frequency. Compare with exact value (dnorm(12,10,5))

```
x0 <- 12
b <- which.min(h.default2$breaks <= x0) - 1
print(c(b, h.default2$density[b]))
```

```
## [1] 5.000 0.068
```

```
b <- which.min(h.sturges2$breaks <= x0) - 1
print(c(b, h.sturges2$density[b]))
```

```
## [1] 7.00000000 0.06841663
```

```
# exact value
dnorm(12,10,5)
```

```
## [1] 0.07365403
```

```
# both values are close to the exact value
```

Scott's Normal Reference Rule:

To select an optimal (or good) smoothing parameter for density estimation, one needs to establish a criterion for comparing smoothing parameters. One approach aims to minimize the squared error in the estimate. Following Scott's approach, we briefly summarize some of the main ideas on L_2 criteria.

MSE of a density estimator $\hat{f}(x)$ at x is

$$\text{MSE}(\hat{f}(x)) = E(\hat{f}(x) - f(x))^2 = V(\hat{f}(x)) + \text{bias}^2(\hat{f}(x))$$

It is simpler to consider the statistic, mean integrated squared error (MISE), given by

$$\text{MISE} = \int E(\hat{f}(x) - f(x))^2 dx = \int \text{MSE}(\hat{f}(x))$$

Optimal choice of bin width is

$$h_n^* = (\frac{6n}{\int f'(x)^2 dx})^{1/3}$$

with Asymptotic MISE;

$$\text{AMISE}^* = [\frac{9}{16} \int f'(x)^2 dx]^{1/3} n^{-2/3}$$

In density estimation f is unknown, so the optimal h cannot be computed exactly, but the asymptotically optimal h depends on the unknown density only through its first derivative.

Scott's Normal Reference Rule, which is calibrated to a normal distribution with variance σ^2, specifies a bin width

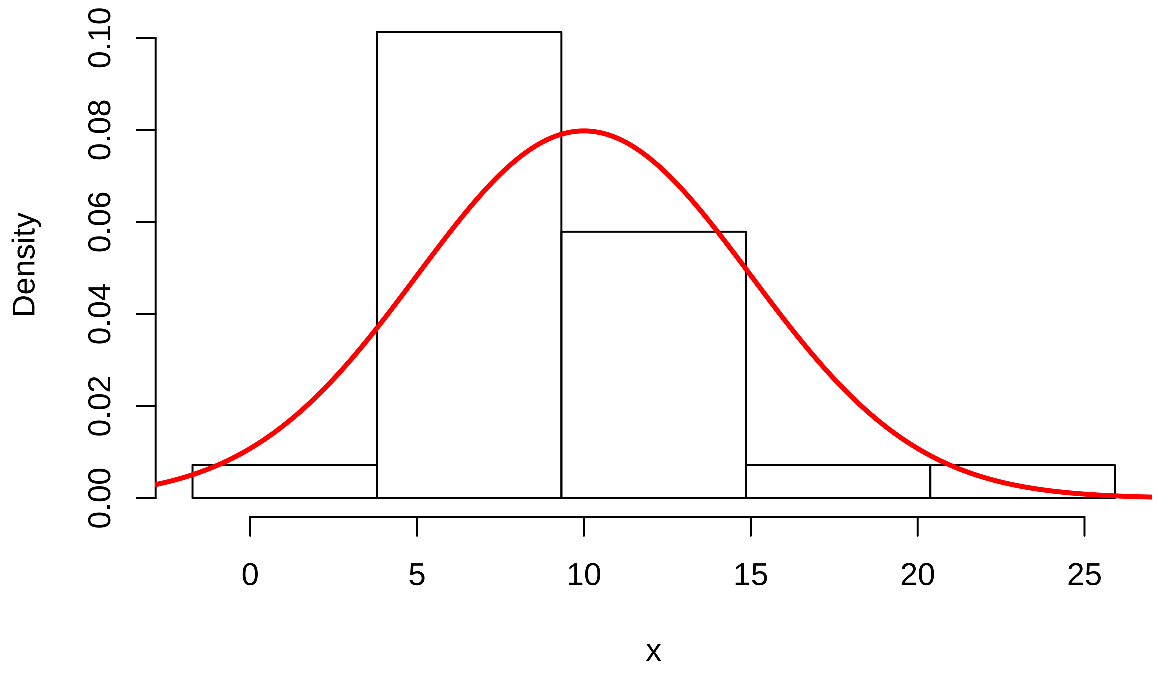
$$\hat{h} = 3.49 * \hat{\sigma} * n^{1/3}$$

where σ^2 is an estimate of the population standard deviation.

Question 2:

Generate random sample of size n = 25 from Normal(10,25) and using Scott's Normal Reference Rule plot a histogram and comment on the histogram.

```r
set.seed(1234)
x <- rnorm(25, 10, 5)

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.scott <- hist(x, breaks = scotts(x)$breaks, freq = FALSE, main = "")
z <- seq(min(x) - scotts(x)$h, max(y) +  scotts(x)$h, 0.01)
lines(z, dnorm(z,mean = 10, sd = 5), col = "Red", lwd = 2.5)
```



Now, let $x_0 = 12$, locate the bin containing the point x_0 and then compute the relative frequency. Compare with exact value (dnorm(12,10,5))

```r
x0 <- 12
b <- which.min(h.scott$breaks <= x0) - 1
print(c(b, h.scott$density[b]))
```

```
## [1] 3.00000000 0.05789638
```

```r
# exact value
dnorm(12,10,5)
```

```
## [1] 0.07365403
```

```r
# both values are close to the exact value
```

Freedman-Diaconis Rule:

Scott's normal reference rule above is a member of a class of rules that select the optimal bin width according to a formula $\hat{h} = T_n^{-1/3}$, where T is a statistic. These $n^{-1/3}$ rules are related to the fact that the optimal rate of decay of bin width with respect to L_p norms is $n^{-1/3}$. The Freedman-Diaconis Rule is another member of this class. For the FD rule, the statistic T is twice the sample interquartile range. That is,

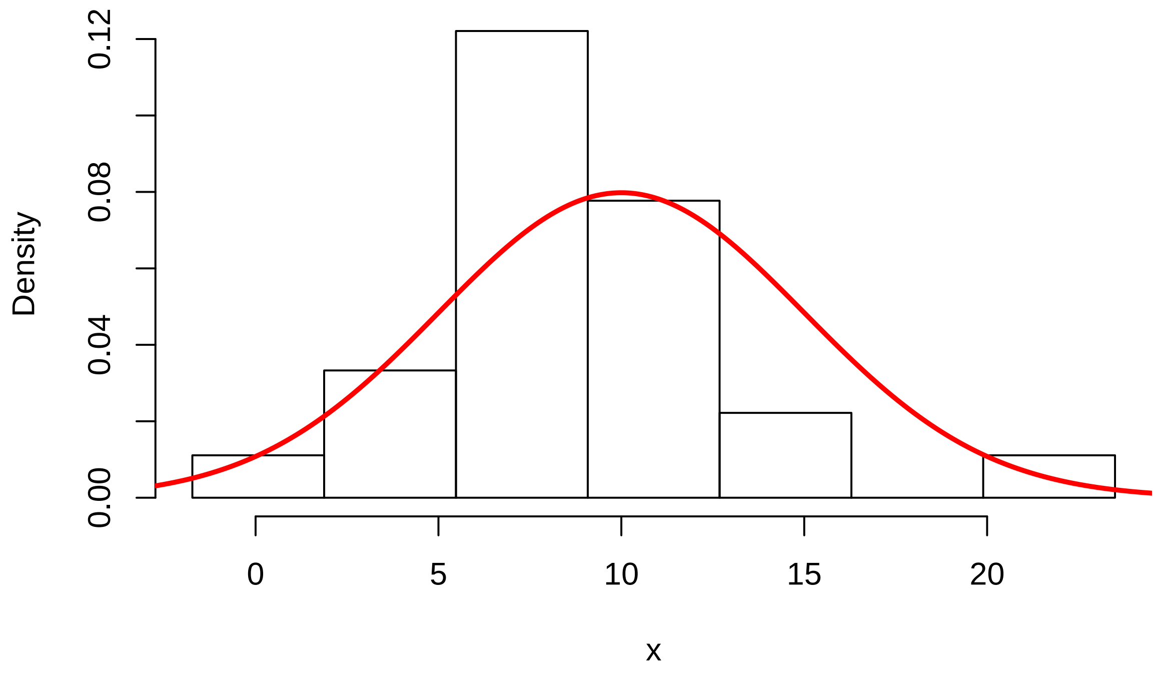
$$\hat{h} = 2 * (\text{IQR}) * n^{-1/3}, \quad \text{IQR} = \text{interquartile range}$$

The IQR is less sensitive than sample standard deviation to outliers in the data.

Question 3:

Generate random sample of size n = 25 from Normal(10,25) and using Freedman-Diaconis Rule plot a histogram and comment on the histogram.

```r
set.seed(1234)
x <- rnorm(25, 10, 5)

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.FD <- hist(x, breaks = FD(x)$breaks, freq = FALSE, main = "")
z <- seq(min(x) - FD(x)$h, max(y) +  FD(x)$h, 0.01)
lines(z, dnorm(z,mean = 10, sd = 5), col = "Red", lwd = 2.5)
```



Now, let $x_0 = 12$, locate the bin containing the point x_0 and then compute the relative frequency. Compare with exact value (dnorm(12,10,5))

```
x0 <- 12
b <- which.min(h.FD$breaks <= x0) - 1
print(c(b, h.FD$density[b]))
```

```
## [1] 4.00000000 0.07770125
```

```
# exact value
dnorm(12,10,5)
```

```
## [1] 0.07365403
```

```
# both values are close to the exact value
```

Frequency Polygon Density Estimate:

All histogram density estimates are piecewise continuous but not continuous over the entire range of the data. A frequency polygon provides a continuous density estimate from the same frequency distribution used to produce the histogram. The frequency polygon is constructed by computing the density estimate at the midpoint of each class interval, and using linear interpolation for the estimates between consecutive midpoints.

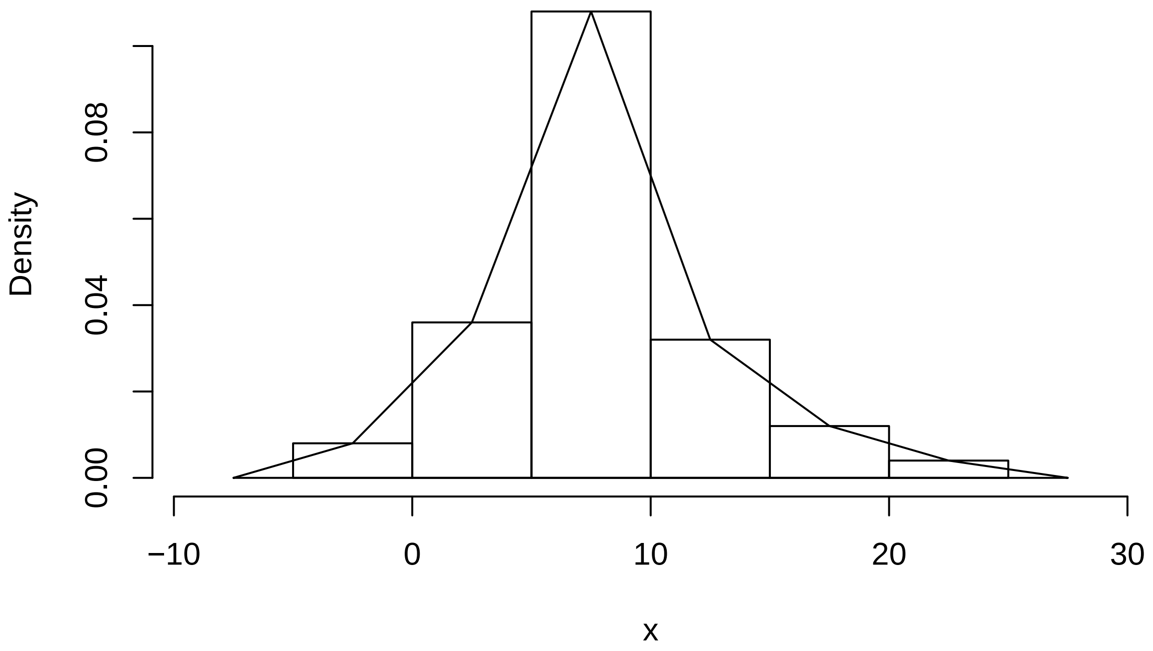
For normal densities, the optimal frequency polygon bin width is

$$h_n^{fp} = 2.15 * \sigma * n^{-1/5}$$

Question 4:

Generate random sample of size n = 25 from Normal(10,25) and using Freedman-Diaconis Rule plot a histogram and comment on the histogram.

```r
set.seed(1234)
x <- rnorm(50, 10, 5)

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, br = br))

}

h.freq <- hist(x, breaks = freq_poly(x)$br, freq = FALSE,
               main = "",xlim = freq_poly(x)$brplus)

#density est at vertices of polygon
vx <- h.freq$mids
vy <- h.freq$density

delta <- diff(vx)[1] # h after pretty is applied
k <- length(vx)
vx <- vx + delta # the bins on the ends
vx <- c(vx[1] - 2 * delta, vx[1] - delta, vx)
vy <- c(0, vy, 0)
# add the polygon to the histogram
polygon(vx, vy)
```


Density
0.00
0.04
0.08
−10
0
10
20
30
x