Demo 10-12

2022-10-12

R has common syntax for working with R.V.s

Each distribution in R has an R name:

- Binomial: `binom`
- Hypergeometric: `hyper`
- Geometric: `geom`
- Negative Binomial: `nbinom`
- Poisson: `pois`
- (Continuous) Unifoirm: `unif`
- Normal: `norm`

(Missing: Discrete uniform)

Append function name to 1 of 4 letters to calculate specific values

1. Compute values of PMF / PDF: `d`

EX: `dnorm(x, mean = 0, sd = 1)` computes the value of the PDF of N(0,1) at x

2. Compute values of CDF: `p`

EX: `pnorm(x, mean = 0, sd = 1)` computing the value of CDF of N(0,1) at x.

3. Compute values of Quantile function: `q`

EX: `qnorm(p, mean = 0, sd = 1)` computes quantile function $F^{-1}(p)$

4. Generate a sample from the distribution of a variable: `r`

EX: `rnorm(1, mean = 0, sd = 1)` a single instance of a random value from N(0,1) distribution

Histograms

A histogram is a visual display of distribution of sample.

Suppose we sample n from the distribution of some variable

```
x<- rnorm(100, mean = 0, sd = 1)
x
```

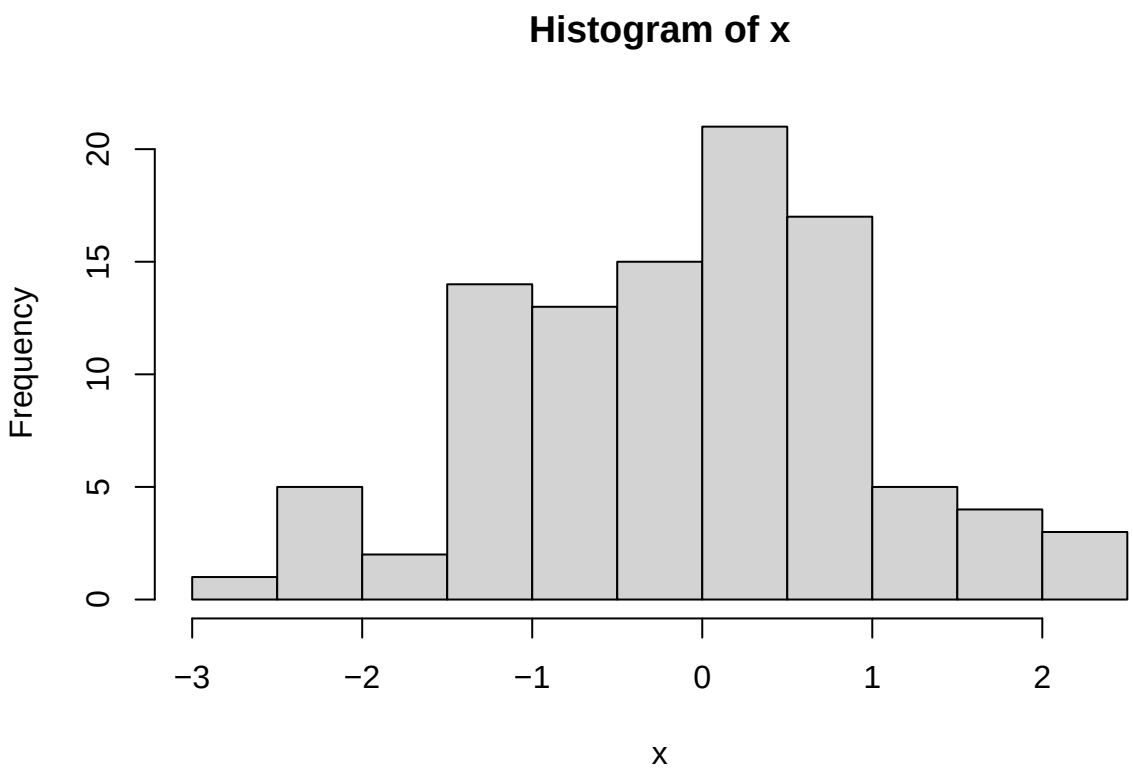
```
##   [1] -1.85963675  1.11053949 -0.84331521  2.10462811 -0.87544212  0.37705581
##   [7] -0.43616483 -1.03012071 -0.51335623 -0.59865456  0.77014704 -1.23913144
##  [13]  0.04647179 -1.19995545  0.27086962  0.22301492 -0.14798787  1.38103753
##  [19] -1.31278803 -0.53394622 -0.65669577  0.36036888  0.68340781 -0.34292248
##  [25] -2.29227276  0.42969335 -0.15733925 -1.42705299 -0.65832166 -1.11923521
##  [31]  0.29790363  0.35561762  0.08718992  1.83926166 -0.41284253 -1.06496413
##  [37] -0.25803771  0.39954321 -0.07890099  0.84385082 -0.86465122  0.21460478
##  [43] -1.04268148  1.28381681  0.81487554  0.84998359 -1.04645423 -0.30970750
```

```
##  [49]  0.43953998  0.01509710  0.66331433 -2.01084894  0.86341651  0.34883324
##  [55]  1.57966539 -0.62396714 -1.45072074  1.57368945  0.23165261  0.64256686
##  [61]  0.18163573 -2.17729125  0.69618635  1.36616341 -0.87212970 -0.34582166
##  [67] -0.17209807  0.08810603 -1.53629114 -0.17007523  0.20209468  0.71345012
##  [73] -0.05439385 -0.18557627  0.06727655  0.42499112  2.14993674  0.69675535
##  [79] -1.05646250 -1.01064136 -1.16219086 -1.10746720  0.75663754 -0.18735072
##  [85] -2.57299839  0.94507213 -2.38353675 -0.80920609  0.74727607 -0.64786519
##  [91]  2.39714572  0.93680209 -2.32127363  1.47232768  0.15295745  0.98955699
##  [97] -0.40190141  0.61445452 -0.90321099  1.80859943
```

Divide range of sampled values into equal length intervals. Count number of sampled points in each interval.

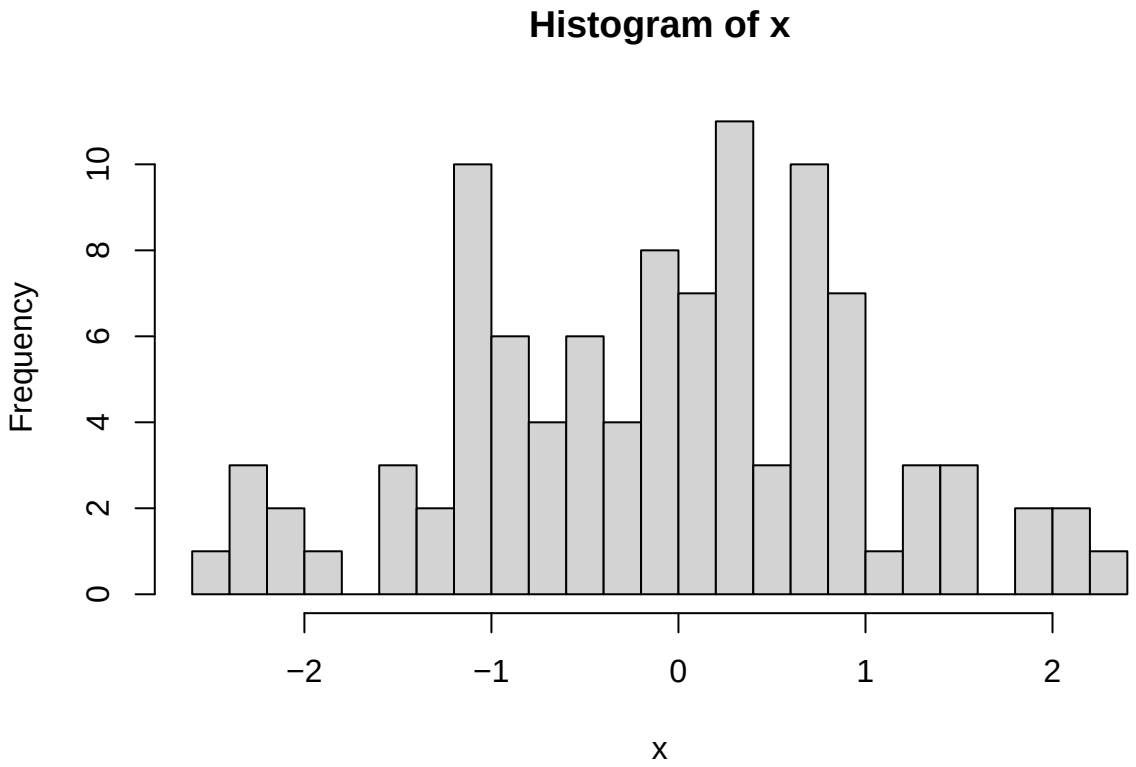
Plot rectangles whose base lies above interval and whose height corresponds to count of number of points in interval.

```
hist(x)
```

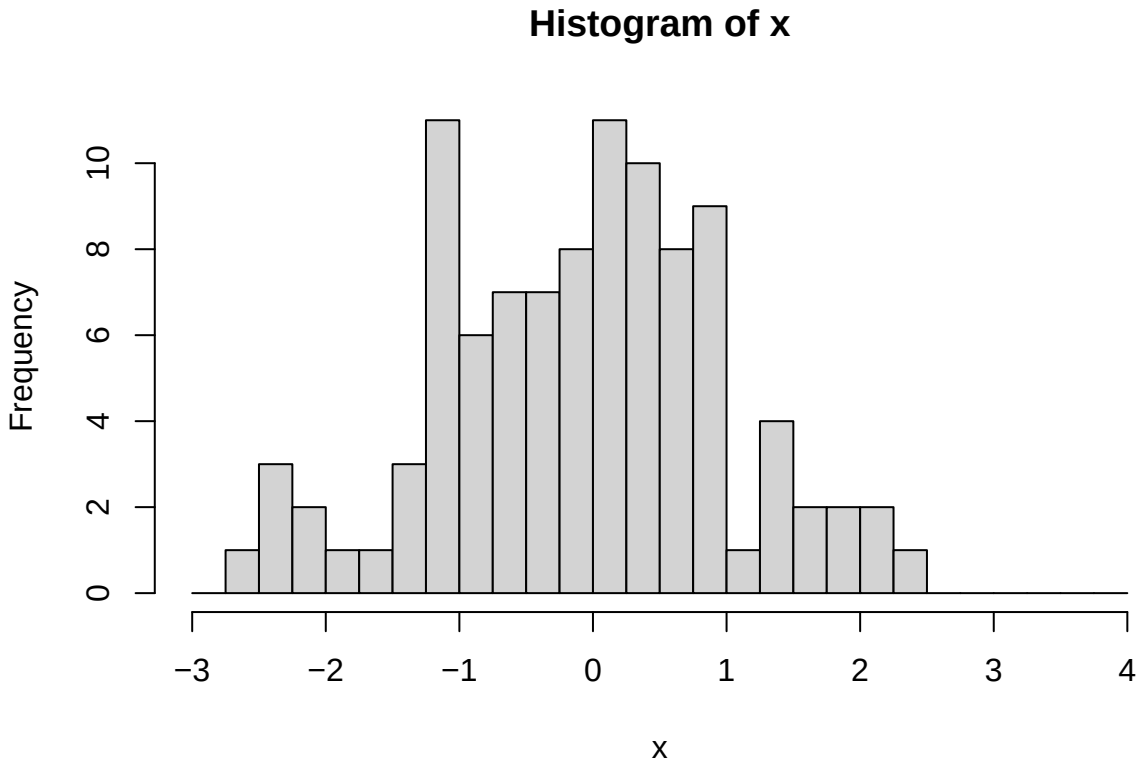


Can also explicitly determine length or number of intervals

```
#to specify m intervals, add breaks = m
hist(x, breaks = 20)
```




```
# for specific lengths, create sequence of endpoints
s <- seq(from = -3, to = 4, by =.25)
hist(x, breaks = s)
```



Often, we want to standardize heights of bars so total area of histogram is 1.

In this way, area of each bar is proportion of observations in each bars interval.

```
hist(x, freq = F)
```