POL211 TA Session

Gento Kato

November 29, 2018

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
rm(list = ls()) # Remove all objects from workspace.
```

Distribution Functions

Binomial Distribution

```
?rbinom
```

- size = number of trials (n in lecture)
- prob = probability of success (p in lecture)

```
# Random Draws from the distribution: n = sample size
rbinom(10, size = 1000, prob = c(0.9))
```

```
##  [1] 903 892 888 895 897 910 919 898 912 920
```

```
# If you want to store generated values
a <- rbinom(10, size = 1000, prob = c(0.9))
a
```

```
##  [1] 887 893 902 892 905 899 893 902 908 905
```

```
# Pr of specific values in the distribution
# i.e., f(x) = Pr(X=x)
dbinom(900, size = 1000, prob = c(0.9))
```

```
## [1] 0.04201679
```

```
# Cummulative Probability of values
# i.e., F(x) Pr(X<=q)
pbinom(900, size = 1000, prob = c(0.9))
```

```
## [1] 0.5154177
```

```
sum(dbinom(0:900, size = 1000, prob = c(0.9)))
```

```
## [1] 0.5154177
```

```
# The value that satisfies the specific cummulative probabilities
qbinom(0.5154177, size = 1000, prob = c(0.9))
```

```
## [1] 900
```

Poisson Distribution

```
?rpois
```

- lambda = mean

```
rpois(10, 414)
```

```
##  [1] 416 405 427 417 373 404 425 433 408 398
```

```r
dpois(400, 414)
```

```
## [1] 0.01569664
```

```r
ppois(400, 414)
```

```
## [1] 0.2549781
```

```r
qpois(0.8, 414)
```

```
## [1] 431
```

Negative Binomial

```r
?rnbinom
```

- size = number of successful trials (r in lecture)
- prob = probability of success in each trial (p in lecture)

```r
rnbinom(10, size=1000, prob=0.8)
```

```
##  [1] 265 255 229 266 250 249 273 280 218 259
```

```r
dnbinom(200, size=1000, prob=0.8)
```

```
## [1] 0.0003296854
```

```r
pnbinom(250, size=1000, prob=0.8)
```

```
## [1] 0.5169171
```

```r
qnbinom(0.8, size=1000, prob=0.8)
```

```
## [1] 265
```

Continuous Uniform

```r
?runif
```

- min = lower limit of distribution (a in lecture)
- max = upper limit of distribution (b in lecture)

```r
runif(10, min=10, max=20)
```

```
##  [1] 11.89269 16.42635 15.27460 16.20346 13.13513 17.66782 12.51829
##  [8] 19.33483 13.63406 19.31468
```

```r
dunif(15, min=10, max=20)
```

```
## [1] 0.1
```

```r
punif(17, min=10, max=20)
```

```
## [1] 0.7
```

```r
qunif(0.8, min=10, max=20)
```

```
## [1] 18
```

Exponential

```
?rexp
```

- rate = lambda in lecture

```
rexp(10, rate=5)
```

```
##  [1] 0.06976083 0.21732206 0.14785124 0.27447509 0.29420726 0.21846445
##  [7] 0.11087085 0.30031731 0.07293613 0.40223845
```

```
dexp(15, rate=5)
```

```
## [1] 1.339318e-32
```

```
pexp(2, rate=2)
```

```
## [1] 0.9816844
```

```
qexp(0.8, rate=5)
```

```
## [1] 0.3218876
```

Normal

```
?rnorm
```

- mean = lower limit of distribution (a in lecture)
- sd = upper limit of distribution (b in lecture)

```
rnorm(10, mean=5, sd=10)
```

```
##  [1]  -2.8559911   7.4557199  -4.2932333   0.1909788  -9.1889776
##  [6]   3.2101397  -5.8116653 -14.4712744   8.6261825   4.3346747
```

```
dnorm(8, mean=5, sd=10)
```

```
## [1] 0.03813878
```

```
pnorm(10, mean=5, sd=10)
```

```
## [1] 0.6914625
```

```
qnorm(0.8, mean=5, sd=10)
```

```
## [1] 13.41621
```

Using Data from MASS Package (Just in case)

```
# Install package (only once per PC)
```

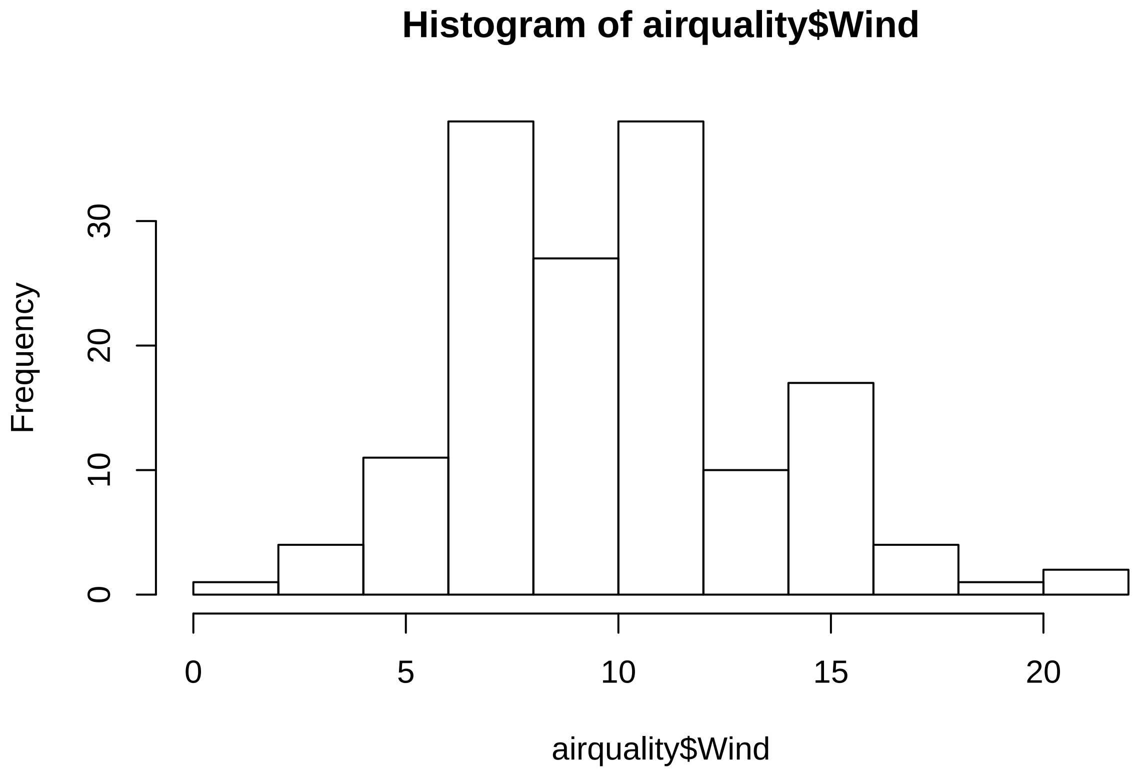
```
install.packages("MASS")
```

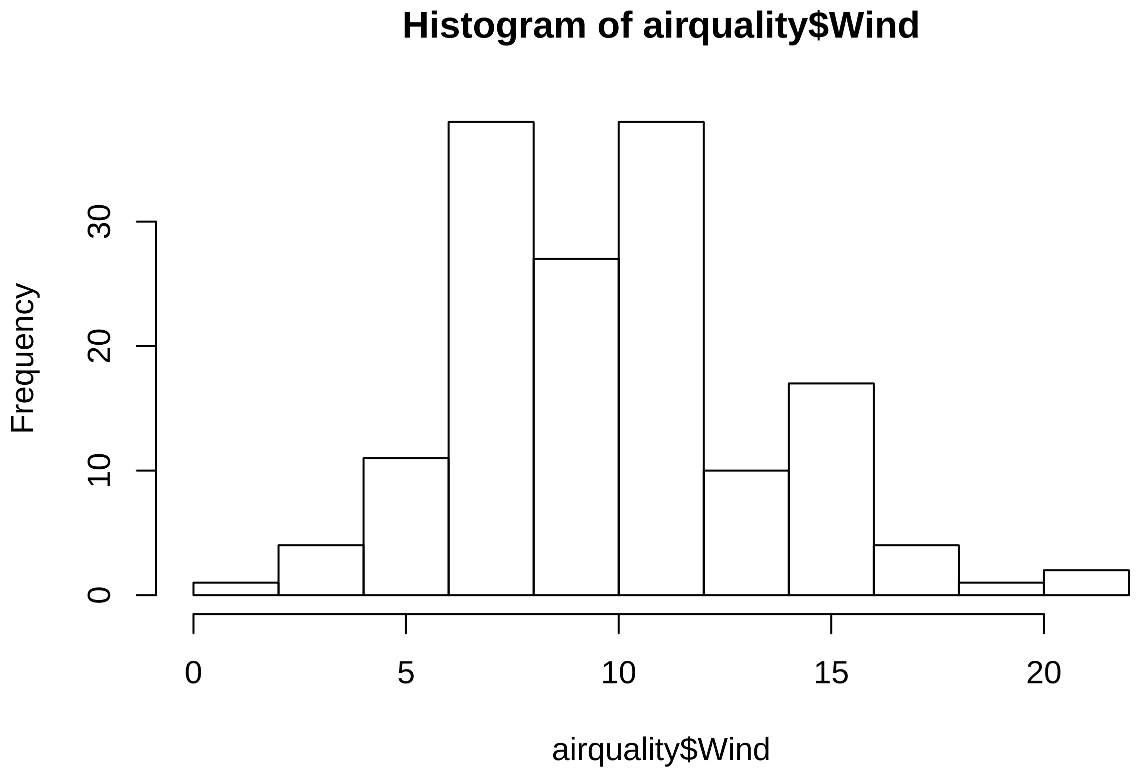
```
library(MASS)

# Data list
data()

# Choose Data
data(airquality)
```

```
# See help file
?airquality
# list of variable names
names(airquality)
```

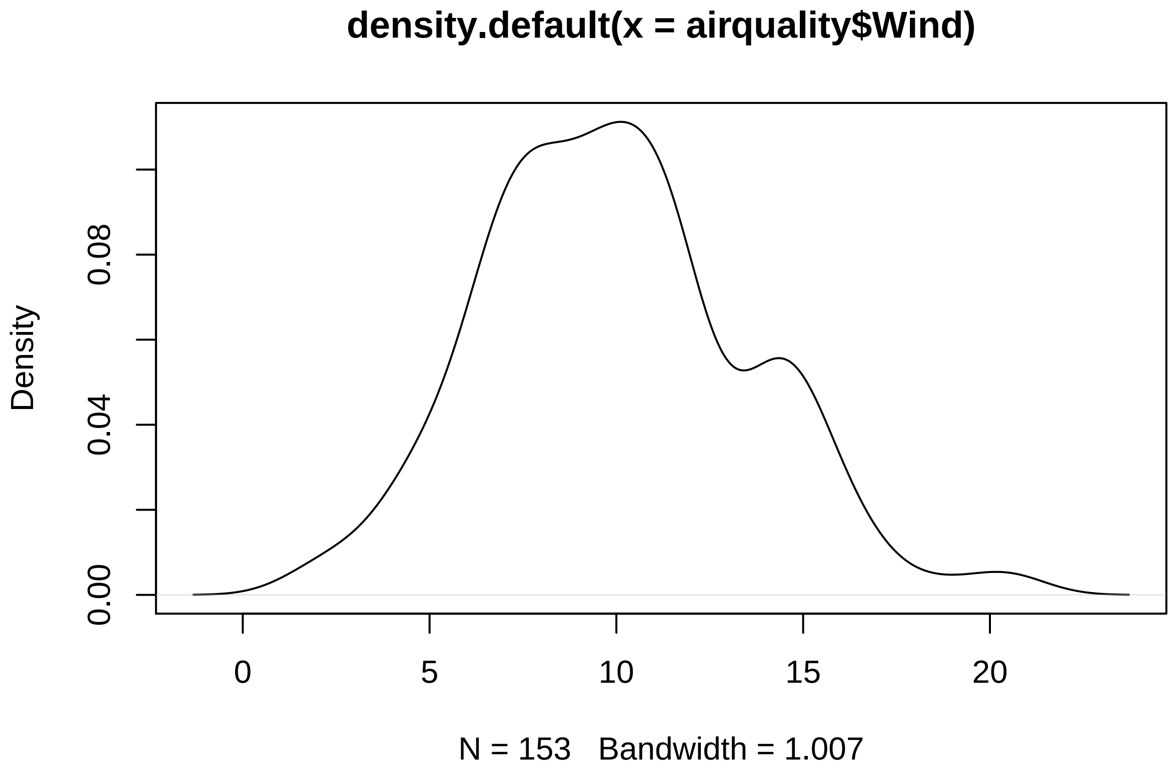
```
## [1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
```

```
# Plot
plot(hist(airquality$Wind)) # Histogram
```





```r
plot(density(airquality$Wind)) # Density
```


density.default(x = airquality$Wind)
Density
0.00
0.04
0.08
0
5
10
15
20
N = 153   Bandwidth = 1.007