Week 4

Martin Aragoneses

July 28, 2015

The structure `str()` function

Alternative to summary()... whats in this object?

```
str(airquality)
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
s <- split(airquality, airquality$Month)
str(s)
```

```
## List of 5
##  $ 5:'data.frame':   31 obs. of  6 variables:
##   ..$ Ozone  : int [1:31] 41 36 12 18 NA 28 23 19 8 NA ...
##   ..$ Solar.R: int [1:31] 190 118 149 313 NA NA 299 99 19 194 ...
##   ..$ Wind   : num [1:31] 7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##   ..$ Temp   : int [1:31] 67 72 74 62 56 66 65 59 61 69 ...
##   ..$ Month  : int [1:31] 5 5 5 5 5 5 5 5 5 5 ...
##   ..$ Day    : int [1:31] 1 2 3 4 5 6 7 8 9 10 ...
##  $ 6:'data.frame':   30 obs. of  6 variables:
##   ..$ Ozone  : int [1:30] NA NA NA NA NA NA 29 NA 71 39 ...
##   ..$ Solar.R: int [1:30] 286 287 242 186 220 264 127 273 291 323 ...
##   ..$ Wind   : num [1:30] 8.6 9.7 16.1 9.2 8.6 14.3 9.7 6.9 13.8 11.5 ...
##   ..$ Temp   : int [1:30] 78 74 67 84 85 79 82 87 90 87 ...
##   ..$ Month  : int [1:30] 6 6 6 6 6 6 6 6 6 6 ...
##   ..$ Day    : int [1:30] 1 2 3 4 5 6 7 8 9 10 ...
##  $ 7:'data.frame':   31 obs. of  6 variables:
##   ..$ Ozone  : int [1:31] 135 49 32 NA 64 40 77 97 97 85 ...
##   ..$ Solar.R: int [1:31] 269 248 236 101 175 314 276 267 272 175 ...
##   ..$ Wind   : num [1:31] 4.1 9.2 9.2 10.9 4.6 10.9 5.1 6.3 5.7 7.4 ...
##   ..$ Temp   : int [1:31] 84 85 81 84 83 83 88 92 92 89 ...
##   ..$ Month  : int [1:31] 7 7 7 7 7 7 7 7 7 7 ...
##   ..$ Day    : int [1:31] 1 2 3 4 5 6 7 8 9 10 ...
##  $ 8:'data.frame':   31 obs. of  6 variables:
##   ..$ Ozone  : int [1:31] 39 9 16 78 35 66 122 89 110 NA ...
##   ..$ Solar.R: int [1:31] 83 24 77 NA NA NA 255 229 207 222 ...
##   ..$ Wind   : num [1:31] 6.9 13.8 7.4 6.9 7.4 4.6 4 10.3 8 8.6 ...
##   ..$ Temp   : int [1:31] 81 81 82 86 85 87 89 90 90 92 ...
##   ..$ Month  : int [1:31] 8 8 8 8 8 8 8 8 8 8 ...
##   ..$ Day    : int [1:31] 1 2 3 4 5 6 7 8 9 10 ...
```

```
##  $ 9:'data.frame':   30 obs. of  6 variables:
##   ..$ Ozone  : int [1:30] 96 78 73 91 47 32 20 23 21 24 ...
##   ..$ Solar.R: int [1:30] 167 197 183 189 95 92 252 220 230 259 ...
##   ..$ Wind   : num [1:30] 6.9 5.1 2.8 4.6 7.4 15.5 10.9 10.3 10.9 9.7 ...
##   ..$ Temp   : int [1:30] 91 92 93 93 87 84 80 78 75 73 ...
##   ..$ Month  : int [1:30] 9 9 9 9 9 9 9 9 9 9 ...
##   ..$ Day    : int [1:30] 1 2 3 4 5 6 7 8 9 10 ...
```

Simple simulation

Generating random numbers

```
# r for random number generation
# d for density
# p for cumulative distribution (CDF)
# q for quantile distribution
str(rnorm)
```

```
## function (n, mean = 0, sd = 1)
```

```
str(dnorm)
```

```
## function (x, mean = 0, sd = 1, log = FALSE)
```

```
str(pnorm)
```

```
## function (q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
```

```
str(qnorm)
```

```
## function (p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
```

```
set.seed(1)
rnorm(5)
```

```
## [1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078
```

```
rnorm(5)
```

```
## [1] -0.8204684  0.4874291  0.7383247  0.5757814 -0.3053884
```

```
set.seed(1) # reset seed
rnorm(5)
```

```
## [1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078
```

```r
## Cummulative distribution
# Pr(x<=2) if rate (mean, lambda) is 2
ppois(2, 2)
```

```
## [1] 0.6766764
```

```r
# Pr(x<=4) if rate (mean, lambda) is 2
ppois(4, 2)
```

```
## [1] 0.947347
```

```r
# Pr(x<=6) if rate (mean, lambda) is 2
ppois(6,2)
```

```
## [1] 0.9954662
```

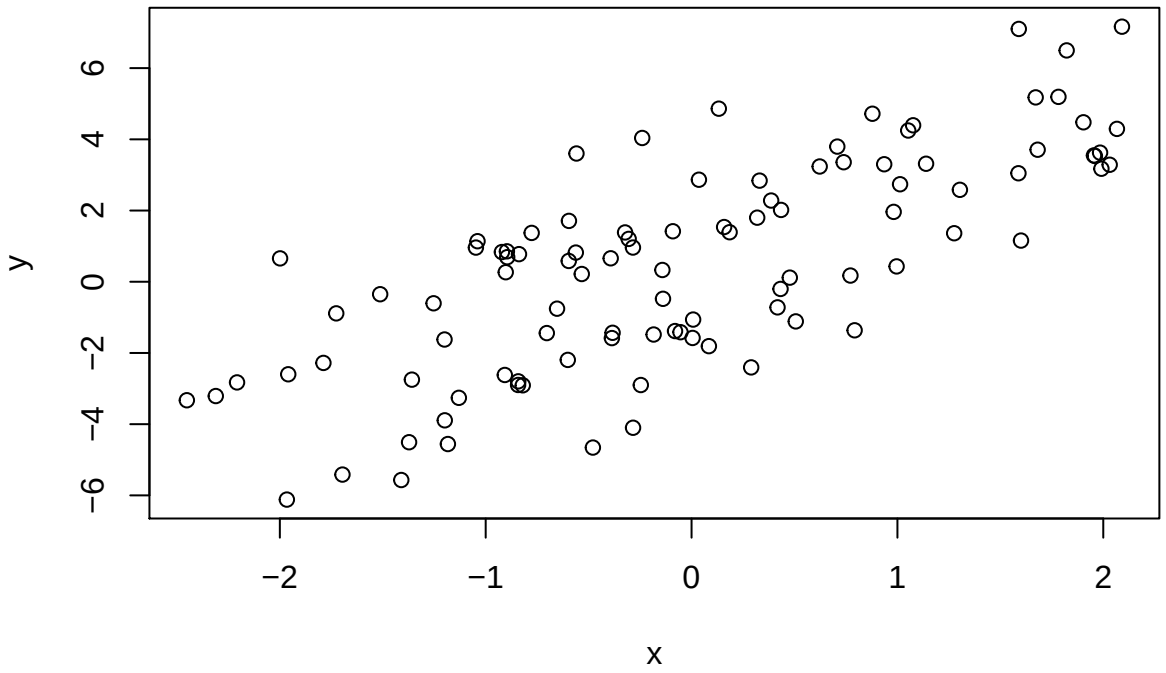
Simulation from linear model

```r
# y = b_0 + b_1*x + e
# e ~ N(0,2^2), x ~ N(0, 1^2), (b_0, b_1) = (0.5, 2)
# 100 observations

set.seed(2)
x <- rnorm(100)
e <- rnorm(100, 0, 2)
y <- 0.5 + 2*x + e
summary(y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -6.1180 -1.5790  0.6741  0.4970  2.9110  7.1640
```

```r
plot(x, y)
```

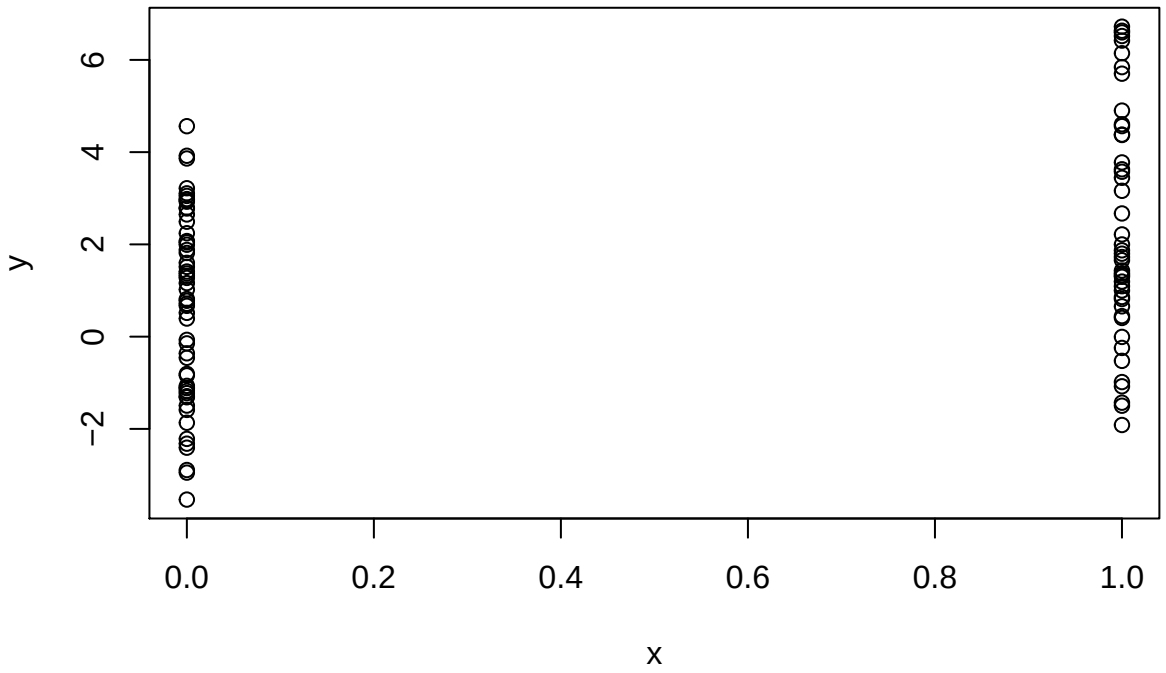


If x is some treatment, we can generate it from binary data

```
set.seed(2)
x <- rbinom(100, 1, 0.5)
e <- rnorm(100, 0, 2)
y <- 0.5 + 2*x + e
summary(y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.5320 -0.3856  1.3520  1.4480  2.9670  6.7200
```

```
plot(x, y)
```



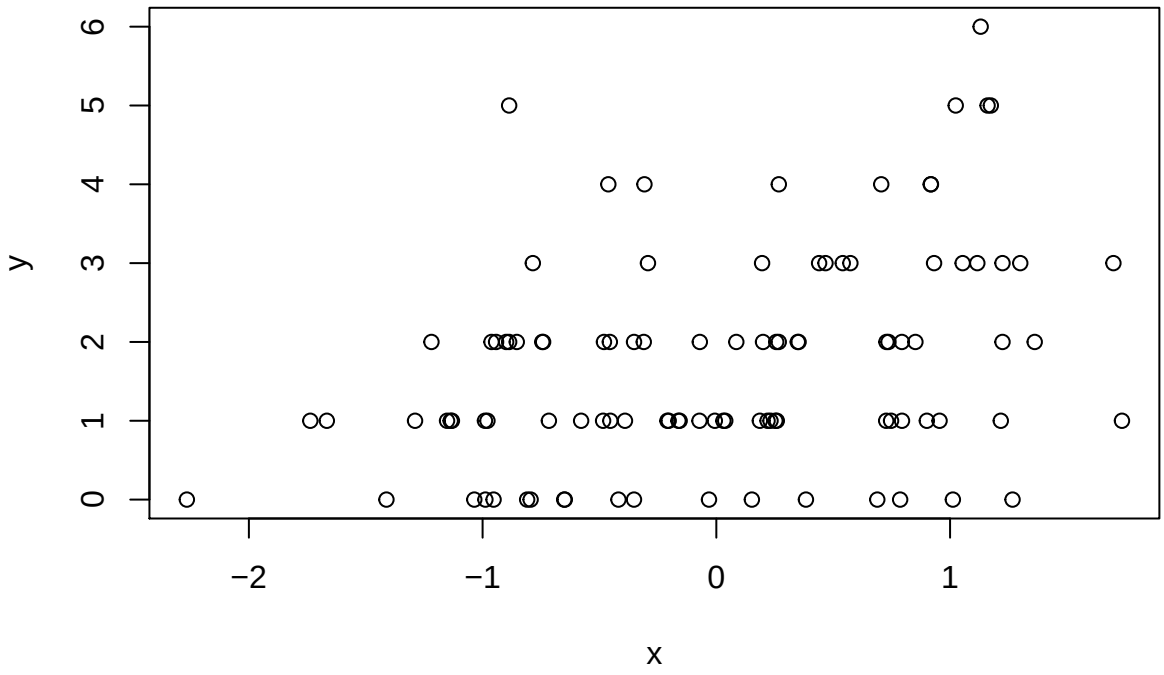
With count y (instead of continuous), the error is not going to be normal, we can use Poisson

```
## Y ~ Poisson(mu)
## log(mu) = b_0 + b_1*x
## b_0 = 0.5, b_1 = 0.3
set.seed(3)
x <- rnorm(100)
mu <- exp(0.5 + 0.3*x)

y <- rpois(100, exp(log(mu)))
summary(y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    1.00    1.00    1.72    2.00    6.00
```

```
plot(x, y)
```


y
0
1
2
3
4
5
6
−2
−1
0
1
x