PSet 1 - IRT Lab

Filipe Recch

January 24, 2018

1 Exploration of collections of bernoulli variables

Q. Compute all the correlations of the columns of this matrix (x1). What do you notice?

```
set.seed(12311)
x1<-matrix(rbinom(1000,1,.5),100,10)

cor(x1)

##               [,1]         [,2]         [,3]          [,4]         [,5]
##  [1,]  1.000000000  0.008410789 -0.009896948 -0.3032770731  0.094629567
##  [2,]  0.008410789  1.000000000  0.089886562 -0.0669602615 -0.057570773
##  [3,] -0.009896948  0.089886562  1.000000000  0.0295469723 -0.125809070
##  [4,] -0.303277073 -0.066960261  0.029546972  1.0000000000  0.089214650
##  [5,]  0.094629567 -0.057570773 -0.125809070  0.0892146505  1.000000000
##  [6,]  0.237526763  0.130744090 -0.001602564 -0.0525279507  0.035484609
##  [7,]  0.081814407 -0.074443750  0.047043222 -0.0988646639  0.060409150
##  [8,]  0.098085811 -0.016333199  0.059259270  0.0455242322 -0.103165975
##  [9,] -0.149960697 -0.107907043  0.053719716 -0.0004168548 -0.001638407
## [10,]  0.025551766  0.179665184  0.060860872  0.0365014114 -0.058030861
##               [,6]         [,7]        [,8]          [,9]       [,10]
##  [1,]  0.237526763  0.081814407  0.09808581 -0.1499606967  0.02555177
##  [2,]  0.130744090 -0.074443750 -0.01633320 -0.1079070433  0.17966518
##  [3,] -0.001602564  0.047043222  0.05925927  0.0537197158  0.06086087
##  [4,] -0.052527951 -0.098864664  0.04552423 -0.0004168548  0.03650141
##  [5,]  0.035484609  0.060409150 -0.10316597 -0.0016384067 -0.05803086
##  [6,]  1.000000000  0.087597723  0.09929932 -0.1904608106 -0.05925927
##  [7,]  0.087597723  1.000000000  0.02350749  0.0090628774  0.05755282
##  [8,]  0.099299317  0.023507488  1.00000000  0.0370118105  0.08043217
##  [9,] -0.190460811  0.009062877  0.03701181  1.0000000000  0.08500515
## [10,] -0.059259270  0.057552816  0.08043217  0.0850051472  1.00000000
```

Q. Compute the row sums. What is the variation in row sums?

```
var(rowSums(x1))

## [1] 2.706667
```

Q. If you considered the 1s/0s correct and incorrect responses to test items (where the rows are people and the columns are items), does this seem like it could have come from a realistic scenario?

Feel free to ignore this bit. I'm going to generate a new set of data.

Q. Now go back through the above questions and see what you make of this new matrix x2. Specifically, how does it compare to the first matrix x1 in terms of whether it seems like a realistic set of item responses? What characteristics (feel free to explore other features of the data) influence your opinion on this point?

```
var(rowSums(x2))

## [1] 5.111111
```

2 Question 2

(A) How would you compare the association between y1/y2 & x? (B) How would you interpret the regression coefficients from (say) m1? (C) Do m1 and m2 show equivalent model fit? Can you notice anything peculiar about either y1 or y2 (in terms of their association with x)?

```
load("ps1-logreg.Rdata")

m1 <- glm(y1~x,df,family="binomial")
m2 <- glm(y2~x,df,family="binomial")
```

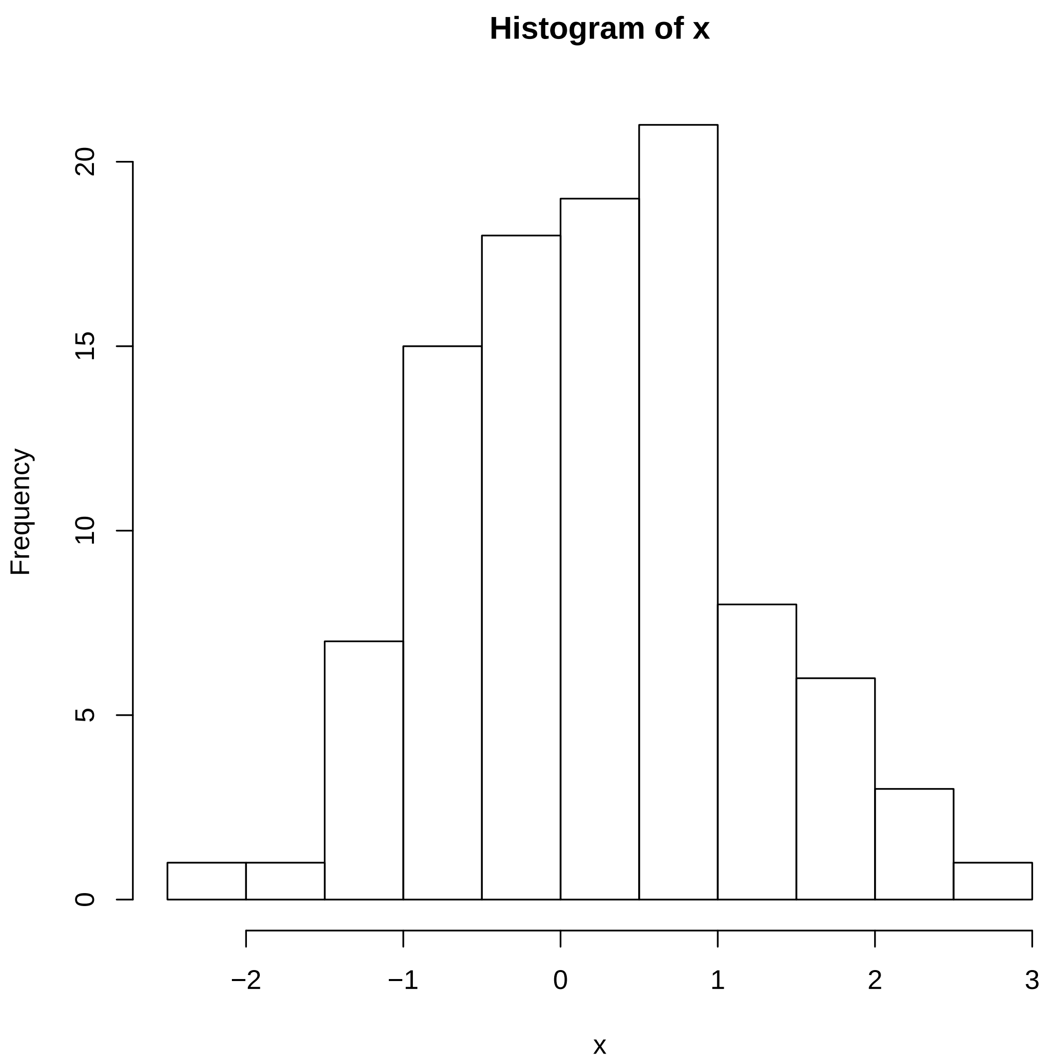
3 Likelihood exploration

Suppose we just observed x, a bunch of random numbers. We first want to see what the distribution looks like. We can do this:

```
x<-rnorm(100)
hist(x)
```

Table 1:

	Dependent variable:	
	y1	y2
	(1)	(2)
x	0.996^{***}	1.440^{***}
	(0.084)	(0.109)
Constant	0.061	1.441^{***}
	(0.069)	(0.097)
Observations	1,000	1,000
Log Likelihood	−602.437	−448.444
Akaike Inf. Crit.	1,208.875	900.889
Note:	*p<0.1; **p<0.05; ***p<0.01	



24

Looks vaguely normalish, no? [Of course, you can see that I'm drawing variates from the normal distribution, so this isn't surprising. Pretend you didn't see that part!] So what if we wanted to estimate the mean and var of the normal distribution that may have generated this set of draws.

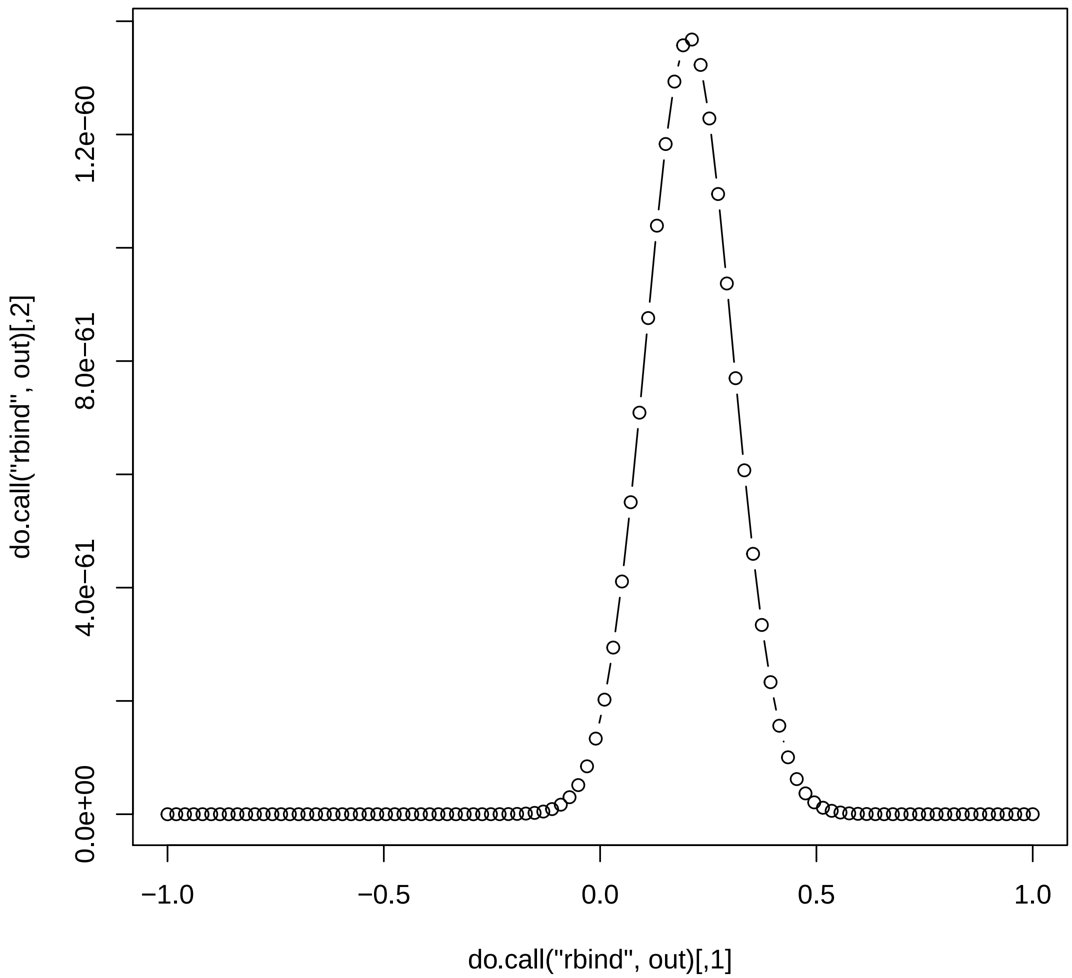
How do we do this? The key statistical technique is to consider the likelihood. Let's start by writing a function that computes the likelihood for "x" in a normal with unknown mean and var (collectively, "pars").

```
likelihood<-function(pars,x) { #see the first eqn here, http://mathworld.wolfram.c
    tmp<-exp(-(x-pars[1])^2/(2*pars[2]))
    tmp/sqrt(2*pars[2]*pi)
}
```

To completely evaluate this function, we would need to know x and pars. We only know x (this is the problem of estimation in general: what are the values in pars?). With known x, we can think of the likelihood as the "probability" of observing the draws x from a normal distribution with parameters pars. That is, we are thinking of the likelihood as a function of pars (x is known).

Let's think about what we get if the mean is unknown and the SD=1 Q. what do you notice?

```
out<-list()
for (m in seq(-1,1,length.out=100)) {
    like<-rep(NA,length(x))
    for (i in 1:length(x)) {
        like[i]<-likelihood(c(m,1),x[i])
    }
    c(c(m,prod(like)))->out[[as.character(m) ]]
}
plot(do.call("rbind",out),type="b") #this is a likelihood surface where we're seei
```



From a computational perspective, working with the products of small numbers is very unstable. So we instead work with the sum of the logs. Why is this ok? First of all, $log(xy) = log(x) + log(y)$ Second, log(f(x)) is a monotic transformation of f(x). So if we maximize log(f(x)) funtion, then we've also maximized f(x) Below is a function that will do this.

Q. How do our estimates vary in accuracy as a function of the sample size (change 100 to something much bigger and much smaller in the top line)

```
ll<-function(pars,x) {
```

```
    likelihood<-function(pars,x) {
        tmp<-exp(-(x-pars[1])^2/(2*pars[2]))
        tmp/sqrt(2*pars[2]*pi)
    }
    like<-rep(NA,length(x))
    for (i in 1:length(x)) {
        like[i]<-likelihood(pars,x[i])
    }
    -1*sum(log(like))
}
optim(par=c(-2,2),ll,x=x)$par #these are the mean and variance estimates produced
```

```
## [1] 0.2058405 0.9189130
```

4 Item Quality

Consider the item statistics (p-values & item-total correlations) discussed in the Crocker & Algina text. What do you think? As a point of contrast, consider them vis-a-vis the item statistics generated by this data:

```
emp_rash <- read.table("emp-rasch.txt")
rasch <- read.table("rasch.txt")
```

5 Buffon's Needle Problem