Chapter 2: Basic Statistics with R

Jixiang Wu

Associate Professor of Quantitative Genetics/Biostatistics
AHPS Department,
South Dakota State University, Brookings, SD 57007
Email: Jixiang.wu@sdstate.edu
Phone: 668-5947

Packages needed

Several R packages will be installed before you can run some functions in the following codes. These R packages include: DAAG, MASS, and leaps.

Binomial test

Pearson's chi-square test

```
#Manual test
R<-682
r<-243
(S<-R+r)

## [1] 925

(ER<-S*0.75)

## [1] 694

(Er<-S*0.25)

## [1] 231

(t<-(R-ER)^2/ER+(r-Er)^2/Er)

## [1] 0.796

(Chi_Pvalue<-1-pchisq(t,df=1))

## [1] 0.372

# Direct use of built-in function

seed=c(682,243)
ratio=c(75,25)
chisq.test(seed,p=ratio,rescale.p=TRUE)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  seed
## X-squared = 0.8, df = 1, p-value = 0.4
```

Likelihood ratio test

```
## Manual calculation
R<-682
r<-243
(S<-R+r)

## [1] 925

(ER<-S*0.75)

## [1] 694

(Er<-S*0.25)

## [1] 231

(g<-(R*log(ER/R)+r*log(Er/r))*(-2))

## [1] 0.787

(LR_Pvalue=1-pchisq(g,df=1))

## [1] 0.375

## Built-in function

binom.test(seed,p=0.75)

##
##  Exact binomial test
##
## data:  seed
## number of successes = 700, number of trials = 900, p-value = 0.4
## alternative hypothesis: true probability of success is not equal to 0.75
## 95 percent confidence interval:
##  0.708 0.765
## sample estimates:
## probability of success
##                  0.737
```

Multi-nomial test: application to independent assortment

```
geno=c(703,216,237,61)
geno.prob=c(9/16,3/16,3/16,1/16)
chisq.test(geno, p=geno.prob)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  geno
## X-squared = 4, df = 3, p-value = 0.2
```

Hardy-Weiberg Equiliabrium test

In this test, you will need to install the package genetics and load it using require function.

```
require(genetics)

## Loading required package: genetics

## Loading required package: combinat

##
## Attaching package: 'combinat'

## The following object is masked from 'package:utils':
##
##     combn

## Loading required package: gdata

## gdata: Unable to locate valid perl interpreter
## gdata:
## gdata: read.xls() will be unable to read Excel XLS and XLSX files
## gdata: unless the 'perl=' argument is used to specify the location
## gdata: of a valid perl intrpreter.
## gdata:
## gdata: (To avoid display of this message in the future, please
## gdata: ensure perl is installed and available on the executable
## gdata: search path.)

## gdata: Unable to load perl libaries needed by read.xls()
## gdata: to support 'XLX' (Excel 97-2004) files.

##

## gdata: Unable to load perl libaries needed by read.xls()
## gdata: to support 'XLSX' (Excel 2007+) files.

##

## gdata: Run the function 'installXLSXsupport()'
## gdata: to automatically download and install the perl
## gdata: libaries needed to support Excel XLS and XLSX formats.

##
## Attaching package: 'gdata'
## The following object is masked from 'package:stats':
##
##     nobs

## The following object is masked from 'package:utils':
##
##     object.size

## The following object is masked from 'package:base':
##
##     startsWith

## Loading required package: gtools

## Loading required package: MASS

## Loading required package: mvtnorm

##

## NOTE: THIS PACKAGE IS NOW OBSOLETE.

##

##   The R-Genetics project has developed an set of enhanced genetics

##   packages to replace 'genetics'. Please visit the project homepage

##   at http://rgenetics.org for informtion.

##

##
## Attaching package: 'genetics'

## The following objects are masked from 'package:base':
##
##     %in%, as.factor, order
```

```
#Blood=569*c(0.835,0.156,0.009)
Blood=c(475,89,5)
geno <- c(rep("M/M",475),
                 rep("M/N",89),
                 rep("N/N",5))

g3  <- genotype(geno)
g3
```

```
##   [1] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
##  [12] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
##  [23] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
##  [34] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
##  [45] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
##  [56] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
##  [67] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
##  [78] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
##  [89] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [100] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [111] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [122] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [133] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [144] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [155] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [166] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [177] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [188] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [199] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [210] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [221] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [232] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [243] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [254] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [265] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [276] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [287] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [298] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [309] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [320] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [331] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [342] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [353] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [364] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [375] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [386] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [397] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [408] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [419] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [430] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [441] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [452] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [463] "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M" "M/M"
## [474] "M/M" "M/M" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N"
## [485] "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N"
## [496] "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N"
## [507] "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N"
## [518] "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N"
## [529] "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N"
## [540] "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N"
## [551] "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N" "M/N"
## [562] "M/N" "M/N" "M/N" "N/N" "N/N" "N/N" "N/N" "N/N"
## Alleles: M N
```

```
HWE.chisq(g3)
```

```
## 
##  Pearson's Chi-squared test with simulated p-value (based on 10000
##  replicates)
## 
## data:  tab
## X-squared = 0.1, df = NA, p-value = 0.8
```

```
HWE.exact(g3)
```

```
## 
##  Exact Test for Hardy-Weinberg Equilibrium
## 
## data:  g3
## N11 = 500, N12 = 90, N22 = 5, N1 = 1000, N2 = 100, p-value = 0.6
```

```
HWE.test(g3)
```

```
## 
##  -----------------------------------
##  Test for Hardy-Weinberg-Equilibrium
##  -----------------------------------
## 
## Call: 
## HWE.test.genotype(x = g3)
## 
## Raw Disequlibrium for each allele pair (D)
## 
## 
##             M        N
##   M           -0.00122
##   N -0.00122          
## 
## Scaled Disequlibrium for each allele pair (D')
## 
## 
##           M      N
##   M        -0.161
##   N -0.161        
## 
## Correlation coefficient for each allele pair (r)
## 
## 
##           M      N
##   M        0.0154
##   N 0.0154        
## 
## Observed vs Expected Allele Frequencies 
## 
##          Obs     Exp  Obs-Exp
## M/M 0.83480 0.83358  0.00122
## N/M 0.07821 0.07943 -0.00122
## M/N 0.07821 0.07943 -0.00122
## N/N 0.00879 0.00757  0.00122
##
## Overall Values
##
##         Value
##   D  -0.00122
##   D' -0.16111
##   r   0.01535
##
## Confidence intervals computed via bootstrap using 1000 samples
##
##     * WARNING: The R^2 disequlibrium statistics is bounded between
##     * [0,1].  The confidence intervals for R^2 values near 0 and 1
##     * are ill-behaved. A rough correction has been applied, but
##     * the intervals still may not be correct for R^2 values near 0
##     * or 1.
##
##
##               Observed  95% CI                 NA's Contains Zero?
##   Overall D   -0.001219 (-0.009061,  0.004940) 0    YES
##   Overall D'  -0.161106 (-1.187620,  0.064303) 0    YES
##   Overall r    0.015351 (-0.064303,  0.112644) 0    YES
##   Overall R^2  0.000236 ( 0.000000,  0.008080) 0    YES
##
## Significance Test:
##
##  Exact Test for Hardy-Weinberg Equilibrium
##
## data:  g3
## N11 = 500, N12 = 90, N22 = 5, N1 = 1000, N2 = 100, p-value = 0.6
```

t-test

```
hd<-read.table("snphead.txt",header=TRUE)
attach(hd)
names(hd)

## [1] "SNP"  "Head"

t.test(Head~SNP, alternative="two.sided",var.equal=TRUE, conf.level=.95)

##
##  Two Sample t-test
##
## data:  Head by SNP
## t = -3, df = 90, p-value = 0.005
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.278 -0.767
## sample estimates:
## mean in group AA mean in group BB
##             58.1             60.6
```

Linear regression analysis

```
cot=read.table("CotYldReg.txt",header=TRUE)
attach(cot)
names(cot)
```

```
## [1] "LP" "BW" "BN" "LS" "SB" "LY"
```

```
head(cot)
```

```
##     LP   BW  BN   LS   SB   LY
## 1 41.4 5.08 634 74.3 28.3 1333
## 2 38.6 5.49 577 68.9 30.7 1221
## 3 38.0 6.21 533 66.9 35.3 1256
## 4 38.2 5.27 599 69.3 29.0 1204
## 5 40.7 5.48 748 67.4 33.0 1664
## 6 40.3 6.23 561 73.0 34.3 1407
```

```
y=LY
reg=lm(y~LP+BW)
summary(reg)
```

```
## 
## Call:
## lm(formula = y ~ LP + BW)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
##   -672   -143     18    158    586 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1266.85     316.07   -4.01  8.1e-05 ***
## LP             56.32       7.16    7.87  1.1e-13 ***
## BW             48.52      24.07    2.02    0.045 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 223 on 253 degrees of freedom
## Multiple R-squared:  0.201,  Adjusted R-squared:  0.195 
## F-statistic: 31.9 on 2 and 253 DF,  p-value: 4.56e-13
```

Linear regression with variable selection

Backward elimination

```
g=lm(LY~., data=cot)
summary(g)
```

```
##
## Call:
## lm(formula = LY ~ ., data = cot)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -119.15  -11.68    3.44   15.96   89.88
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.34e+03   4.38e+01  -53.35   <2e-16 ***
## LP           1.51e+01   5.52e+00    2.73   0.0067 **
## BW           7.09e+01   3.55e+01    2.00   0.0469 *
## BN           2.00e+00   1.95e-02  102.82   <2e-16 ***
## LS           9.79e+00   2.78e+00    3.52   0.0005 ***
## SB           2.43e+01   6.78e+00    3.59   0.0004 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.6 on 250 degrees of freedom
## Multiple R-squared:  0.986,  Adjusted R-squared:  0.986
## F-statistic: 3.52e+03 on 5 and 250 DF,  p-value: <2e-16
```

```
g=update(g, .~. -BW)
summary(g)
```

```
##
## Call:
## lm(formula = LY ~ LP + BN + LS + SB, data = cot)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -120.93  -11.87    3.73   16.57   87.17
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.33e+03   4.39e+01  -53.07   <2e-16 ***
## LP           4.46e+00   1.49e+00    2.99    0.003 **
## BN           2.00e+00   1.96e-02  102.20   <2e-16 ***
## LS           1.53e+01   4.72e-01   32.35   <2e-16 ***
## SB           3.78e+01   6.68e-01   56.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 29.8 on 251 degrees of freedom
## Multiple R-squared:  0.986,  Adjusted R-squared:  0.986 
## F-statistic: 4.35e+03 on 4 and 251 DF,  p-value: <2e-16
```

```
g=update(g, .~. -LP)
summary(g)
```

```
## 
## Call:
## lm(formula = LY ~ BN + LS + SB, data = cot)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -117.73  -12.03    4.76   15.58   86.18 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -2.27e+03   3.92e+01   -57.8   <2e-16 ***
## BN           2.03e+00   1.66e-02   122.8   <2e-16 ***
## LS           1.63e+01   3.18e-01    51.4   <2e-16 ***
## SB           3.83e+01   6.57e-01    58.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 30.3 on 252 degrees of freedom
## Multiple R-squared:  0.985,  Adjusted R-squared:  0.985 
## F-statistic: 5.62e+03 on 3 and 252 DF,  p-value: <2e-16
```

Stepwise selection

```
g=lm(LY~., data=cot)
summary(g)
```

```
## 
## Call:
## lm(formula = LY ~ ., data = cot)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -119.15  -11.68    3.44   15.96   89.88 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -2.34e+03   4.38e+01  -53.35   <2e-16 ***
## LP           1.51e+01   5.52e+00    2.73   0.0067 ** 
## BW           7.09e+01   3.55e+01    2.00   0.0469 *  
## BN           2.00e+00   1.95e-02  102.82   <2e-16 ***
## LS           9.79e+00   2.78e+00    3.52   0.0005 ***
## SB           2.43e+01   6.78e+00    3.59   0.0004 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 29.6 on 250 degrees of freedom
## Multiple R-squared:  0.986,  Adjusted R-squared:  0.986
## F-statistic: 3.52e+03 on 5 and 250 DF,  p-value: <2e-16
```

```r
step(g)
```

```
## Start:  AIC=1741
## LY ~ LP + BW + BN + LS + SB
##
##        Df Sum of Sq     RSS  AIC
## <none>                219477 1741
## - BW    1      3501  222979 1743
## - LP    1      6554  226032 1747
## - LS    1     10908  230386 1751
## - SB    1     11294  230771 1752
## - BN    1   9281085 9500562 2704

##
## Call:
## lm(formula = LY ~ LP + BW + BN + LS + SB, data = cot)
##
## Coefficients:
## (Intercept)           LP           BW           BN           LS
##    -2336.24        15.08        70.91         2.00         9.79
##          SB
##       24.31
```

Another stepwise selection

```r
require(MASS)
require(DAAG)
```

```
## Loading required package: DAAG

## Loading required package: lattice

##
## Attaching package: 'DAAG'

## The following object is masked from 'package:MASS':
##
##     hills
```

```r
fit <- lm(y~BN+LP+BW+LS+SB,data=cot)
step <- stepAIC(fit, direction="both")
```

```
## Start:  AIC=1741
## y ~ BN + LP + BW + LS + SB
##
##        Df Sum of Sq     RSS  AIC
## <none>                219477 1741
## - BW    1      3501  222979 1743
## - LP    1      6554  226032 1747
## - LS    1     10908  230386 1751
## - SB    1     11294  230771 1752
## - BN    1   9281085 9500562 2704
```

```
step$anova # display results
```

```
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## y ~ BN + LP + BW + LS + SB
## 
## Final Model:
## y ~ BN + LP + BW + LS + SB
## 
## 
##   Step Df Deviance Resid. Df Resid. Dev  AIC
## 1                        250     219477 1741
```

Best subset selection

```
require(leaps)
```

```
## Loading required package: leaps
```

```
b=regsubsets(LY~., data=cot, nbest=2)
summary(b)
```

```
## Subset selection object
## Call: regsubsets.formula(LY ~ ., data = cot, nbest = 2)
## 5 Variables  (and intercept)
##    Forced in Forced out
## LP     FALSE      FALSE
## BW     FALSE      FALSE
## BN     FALSE      FALSE
## LS     FALSE      FALSE
## SB     FALSE      FALSE
## 2 subsets of each size up to 5
## Selection Algorithm: exhaustive
##          LP  BW  BN  LS  SB 
## 1  ( 1 ) " " " " "*" " " " "
## 1  ( 2 ) "*" " " " " " " " "
## 2  ( 1 ) " " "*" "*" " " " "
## 2  ( 2 ) " " " " "*" " " "*"
## 3  ( 1 ) "*" "*" "*" " " " "
## 3  ( 2 ) " " " " "*" "*" "*"
## 4  ( 1 ) "*" " " "*" "*" "*"
## 4  ( 2 ) " " "*" "*" "*" "*"
## 5  ( 1 ) "*" "*" "*" "*" "*"
```