Fit mutant with binomial aging model

h qin

June 28 2017 - 13 January, 2018

```
rm(list=ls())
host = "Applejack" #"Ridgeside"
if (host == "AppleJack") {
 setwd("~/github/bmc_netwk_aging_manuscript/R1/1.kaeberlein04plos")
}
if (host == "Ridgeside") {

}

library('flexsurv')
```

```
## Loading required package: survival
```

```
library('stringr')
source("../lifespan.r")
list.files()
```

```
##  [1] "0.example.histogram.density.overlay.Rmd"
##  [2] "0.fit.kaeberlein04.html"
##  [3] "0.fit.kaeberlein04.pdf"
##  [4] "0.fit.kaeberlein04.Rmd"
##  [5] "0.histogram.density.overlay.html"
##  [6] "0.histogram.density.overlay.Rmd"
##  [7] "092304.merged.rls.csv"
##  [8] "1.boostrap.kaeblein04.html"
##  [9] "1.boostrap.kaeblein04.pdf"
## [10] "1.boostrap.kaeblein04.Rmd"
## [11] "2009"
## [12] "bootstrap"
## [13] "Kerberlein04PLoS.pdf"
## [14] "lifespan.20160926.r"
## [15] "lifespan.r"
## [16] "sandbox"
```

```
tb.ori = read.table("092304.merged.rls.csv", header = T, sep="\t")
#summary(tb.ori)
```

Explore the fitting outcomes of 'flexsurv'.

```
strains = names(tb.ori)
strains
```

```
##  [1] "X031104.BY2N.RLS.tab"          "X031104.SK1.RLS.tab"
##  [3] "X042204.KLBY4716.RLS.tab"      "X052604.sir2D.rls.tab"
##  [5] "X052604.W303.rls.tab"          "X053104.BY4741.rls.tab"
##  [7] "X053104.BY4742.rls.tab"        "X053104.JSBY4741.rls.tab"
##  [9] "X061004.srp1ts.rls.tab"        "fig1a.BY4742"
```

```
## [11] "fig1a.fob1"                      "fig1a.gpa2"
## [13] "fig1a.gpr1"                      "fig1a.hxk2"
## [15] "fig1b.BY4742"                    "fig1b.fob1"
## [17] "fig1b.hxk2"                      "fig1b.fob1.hxk2"
## [19] "fig1c.BY4742"                    "fig1c.fob1"
## [21] "fig1c.gpa2"                      "fig1c.fob1.gpa2"
## [23] "fig2a.BY4742"                    "fig2a.sir2"
## [25] "fig2a.hxk2.sir2"                 "fig2a.gpa2.sir2"
## [27] "fig2b.BY4742"                    "fig2b.sir2"
## [29] "fig2b.sir2.fob1"                 "fig2c.BY4742"
## [31] "fig2c.sir2.fob1"                 "fig2c.sir2.fob1.hxk2"
## [33] "fig2d.BY4742"                    "fig2d.sir2.fob1"
## [35] "fig2d.gpa2.sir2.fob1"            "fig3.by4742.2glucose"
## [37] "fig3.sir2.fob1.2glucose"         "fig3.by4742.05glucose"
## [39] "fig3.sir2.fob1.05glucose"        "fig3.WT.01glucose"
## [41] "fig3.sir2.fob1.01glucose"        "fig3.by4742.005glucose"
## [43] "fig3.sir2.fob1.005glucose"       "fig4a.PSY316"
## [45] "fig4a.PSY316.fob1"               "fig4a.PSY316.sir2ox"
## [47] "fig4b.BY4742.2glucose"           "fig4b.BY4742.005glucose"
## [49] "fig4b.by4742.SIR2.ox.2glucose"   "fig4b.by4742.SIR2.ox.005glucose"
```

```
my.strains = c("fig4b.BY4742.2glucose",  "fig4b.by4742.SIR2.ox.2glucose",  "fig2a.sir2", "fig2b.sir2",
#my.strains = strains
tb = tb.ori[, my.strains]

report = data.frame(my.strains)
report$samplesize = NA; report$R=NA; report$t0=NA; report$n=NA; report$G=NA; #report$longfilename=NA;
```

Now, fit all RLS data sets by strains

```
for( i in 1:length(report[,1])){
  #report$samplesize[i] = length(tb[,i])
  my.data = tb[,i]
  my.data = my.data[! is.na(my.data)]
  report$samplesize[i] = length(my.data)

  GompFlex = flexsurvreg(formula = Surv(tb[,i]) ~ 1, dist = 'gompertz')
  WeibFlex = flexsurvreg(formula = Surv(tb[,i]) ~ 1, dist = 'weibull')

  report$avgLS[i] =  mean(tb[,i], na.rm=T)
  report$stdLS[i] =  sd(tb[,i], na.rm = T)
  report$CV[i] = report$stdLS[i] / report$avgLS[i]

  report$GompGFlex[i] = GompFlex$res[1,1]
  report$GompRFlex[i] = GompFlex$res[2,1]
  report$GompLogLikFlex[i] = round(GompFlex$loglik, 1)
  report$GompAICFlex[i] = round(GompFlex$AIC)

  report$WeibShapeFlex[i] = WeibFlex$res[1,1]
  report$WeibRateFlex[i] = WeibFlex$res[2,1]
  report$WeibLogLikFlex[i] = round(WeibFlex$loglik, 1)
  report$WeibAICFlex[i] = round(WeibFlex$AIC)
```

```
  #set initial values
  Rhat = report$GompRFlex[i]; # 'i' was missing. a bug costed HQ a whole afternoon.
  Ghat = report$GompGFlex[i];
  nhat = 6;
  t0= (nhat-1)/Ghat;
  fitBinom = optim ( c(Rhat, t0, nhat),  llh.binomialMortality.single.run,
                     lifespan=tb[,i],
                     #method='SANN') #SANN needs control
                     method="L-BFGS-B",
                     lower=c(1E-10, 1, 1), upper=c(1,200,20) );
  report[i, c("R", "t0", "n")] = fitBinom$par[1:3]
  report$G[i] = (report$n[i] - 1)/report$t0[i]
}
```

Show the results

```
#report[ grep("tBY", report$strains), ]
report
```

```
##                       my.strains samplesize           R        t0        n
## 1         fig4b.BY4742.2glucose         60 0.002388227  35.59111 7.703055
## 2 fig4b.by4742.SIR2.ox.2glucose         60 0.003068054  66.82089 8.067874
## 3                     fig2a.sir2         90 0.002656909  16.80162 8.109348
## 4                     fig2b.sir2         90 0.002656909  16.80162 8.109348
## 5                   fig1b.BY4742        250 0.004796452  58.10834 8.109395
## 6                     fig1b.fob1        140 0.003056547  71.37470 7.695271
## 7                     fig1b.hxk2        120 0.005736777 103.59300 7.578456
## 8                fig1b.fob1.hxk2        160 0.005902597 120.65937 6.023612
##            G    avgLS     stdLS        CV  GompGFlex   GompRFlex
## 1 0.18833511 26.06667  7.557389 0.2899254 0.13946781 0.002283890
## 2 0.10577341 34.60000 10.834972 0.3131495 0.07468834 0.004042203
## 3 0.42313480 13.96667  3.491402 0.2499810 0.27970246 0.003555048
## 4 0.42313480 13.96667  3.491402 0.2499810 0.27970246 0.003555048
## 5 0.12234725 26.62400  9.418144 0.3537464 0.08544660 0.006577615
## 6 0.09380453 37.75000 13.405746 0.3551191 0.06987148 0.003424637
## 7 0.06350290 36.73333 16.338197 0.4447785 0.04823602 0.006725196
## 8 0.04163466 48.28125 21.337707 0.4419460 0.04143860 0.004249514
##   GompLogLikFlex GompAICFlex WeibShapeFlex WeibRateFlex WeibLogLikFlex
## 1         -206.8         418      3.925269     28.75636         -206.2
## 2         -235.6         475      3.369961     38.51039         -227.8
## 3         -246.0         496      4.431406     15.29449         -240.4
## 4         -246.0         496      4.431406     15.29449         -240.4
## 5         -937.3        1879      3.024640     29.77138         -914.0
## 6         -565.5        1135      3.139805     42.25834         -559.6
## 7         -507.6        1019      2.397501     41.42400         -502.2
## 8         -713.6        1431      2.441562     54.41078         -714.3
##   WeibAICFlex
## 1         416
## 2         460
## 3         485
## 4         485
## 5        1832
## 6        1123
## 7        1008
```

```
## 8        1433
```

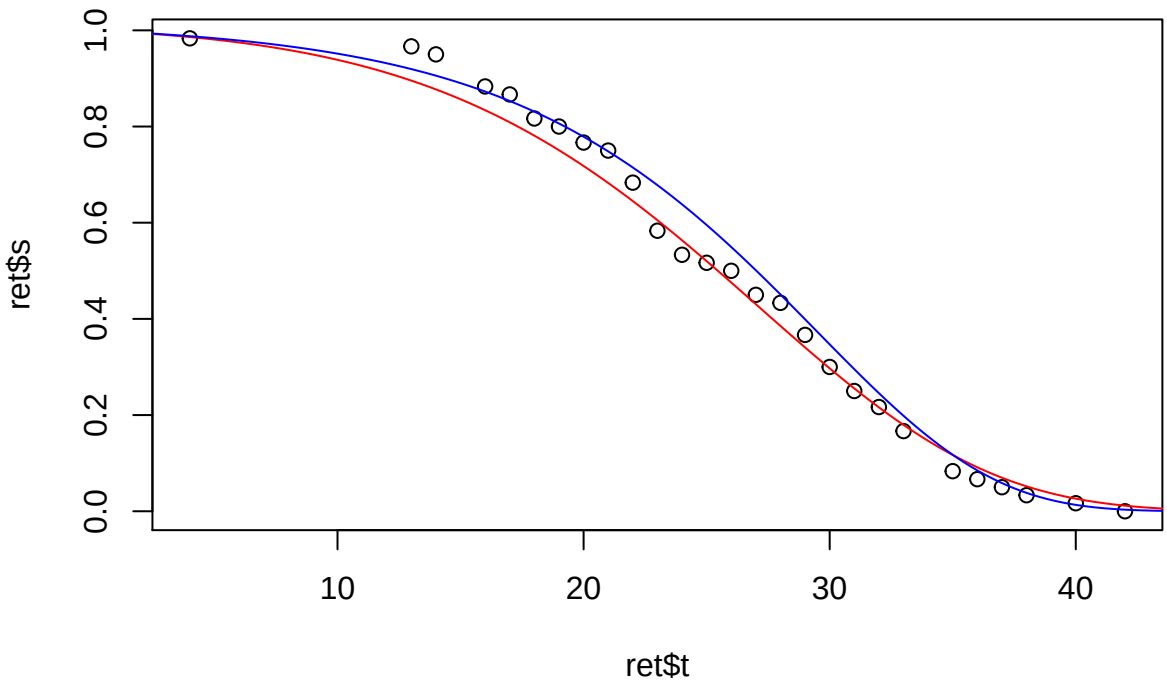
Output

```
write.csv(report, file = 'sandbox/_report_kaeberlein04_fit.csv', row.names = FALSE)
```

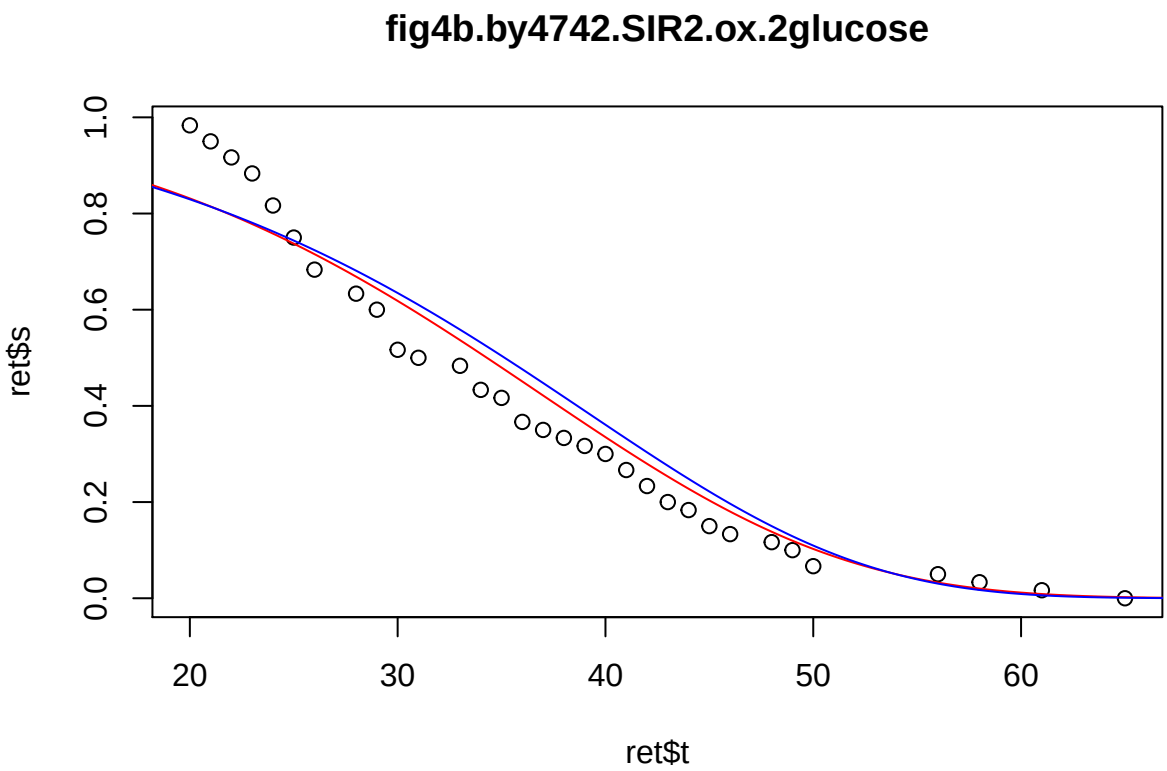
Overlay the binomial aging model with observation.

see http://hongqinlab.blogspot.com/2013/12/binomial-mortailty-model.html m = R (1 + t/t0)^(n-1) s = exp((R t0/n)*(1 - (1+t/t0)^n))

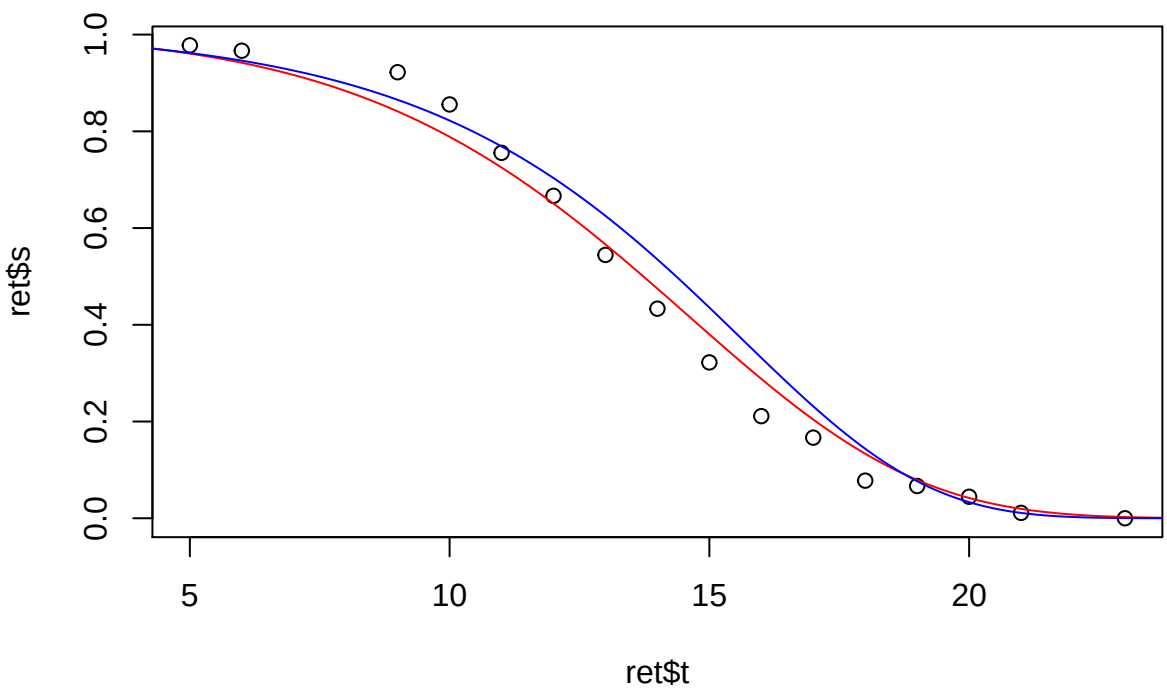
```
for( i in 1:length(report[,1])){
  #report$samplesize[i] = length(tb[,i])
  my.data = tb[,i]
  my.data = my.data[! is.na(my.data)]

  ret = calculate.s(my.data)
  plot( ret$s ~ ret$t, main=my.strains[i]);
  print (report[i,  ]);

    #overlay binomial aging viability
  print (report[i, c("R", "t0", "n", "G")] );
  t = seq(0,max(ret$t*1.1), by=0.1);
  # s = exp( (R t0/n)*(1 - (1+t/t0)^n ) )
  s = exp( (report$R[i]* report$t0[i]/report$n[i])*(1 - (1+t/report$t0[i])^report$n[i] ) ) ;
  lines(s ~ t, col='red')

  #overlay gompertz  viability
  s.g =  G.s( c(report$GompRFlex[i], report$GompGFlex[i], 0), t )
  lines(s.g ~ t, col='blue')

}
```



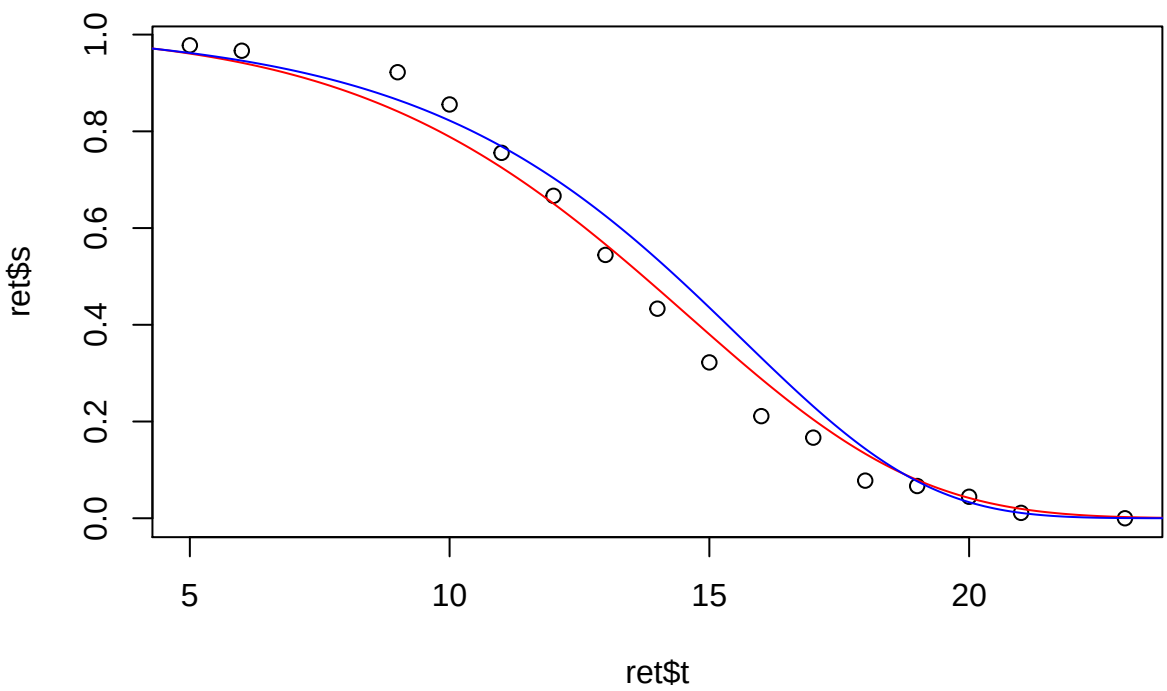

```
##              my.strains samplesize           R       t0        n         G
## 1 fig4b.BY4742.2glucose         60 0.002388227 35.59111 7.703055 0.1883351
##      avgLS    stdLS        CV GompGFlex  GompRFlex GompLogLikFlex
## 1 26.06667 7.557389 0.2899254 0.1394678 0.00228389         -206.8
##   GompAICFlex WeibShapeFlex WeibRateFlex WeibLogLikFlex WeibAICFlex
## 1         418      3.925269     28.75636         -206.2         416
##             R       t0        n         G
## 1 0.002388227 35.59111 7.703055 0.1883351
```



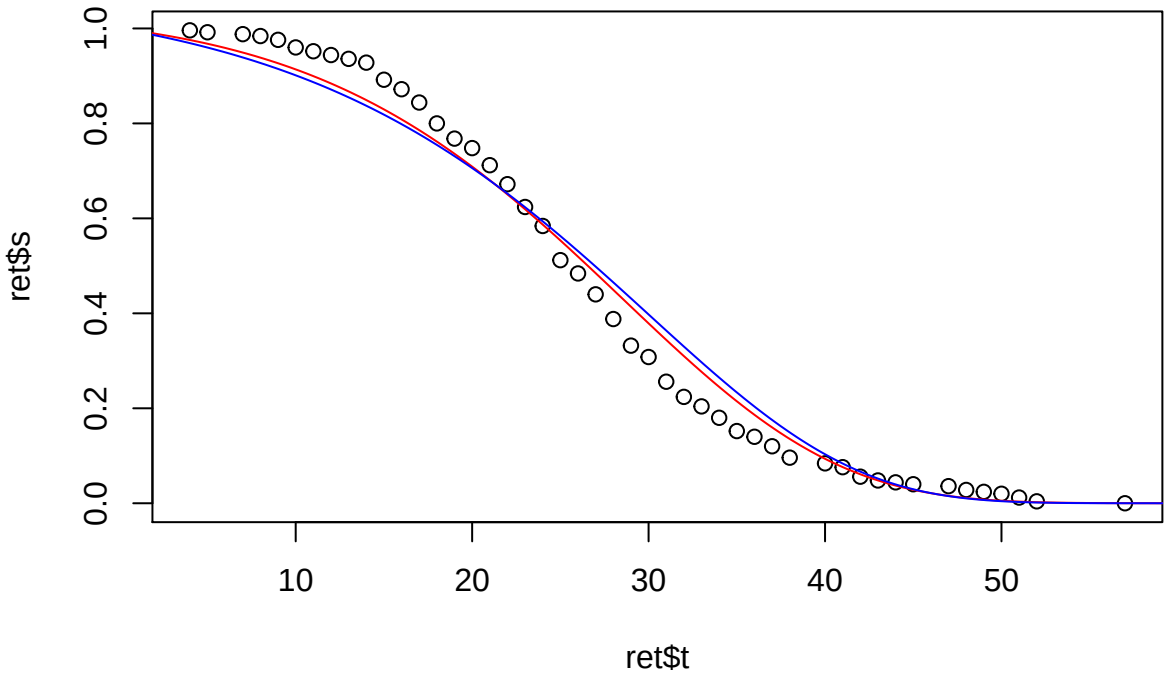

```
##                       my.strains samplesize           R       t0        n
## 2 fig4b.by4742.SIR2.ox.2glucose         60 0.003068054 66.82089 8.067874
##           G avgLS    stdLS        CV  GompGFlex   GompRFlex GompLogLikFlex
## 2 0.1057734  34.6 10.83497 0.3131495 0.07468834 0.004042203         -235.6
##   GompAICFlex WeibShapeFlex WeibRateFlex WeibLogLikFlex WeibAICFlex
## 2         475      3.369961     38.51039         -227.8         460
##             R       t0        n         G
## 2 0.003068054 66.82089 8.067874 0.1057734
```



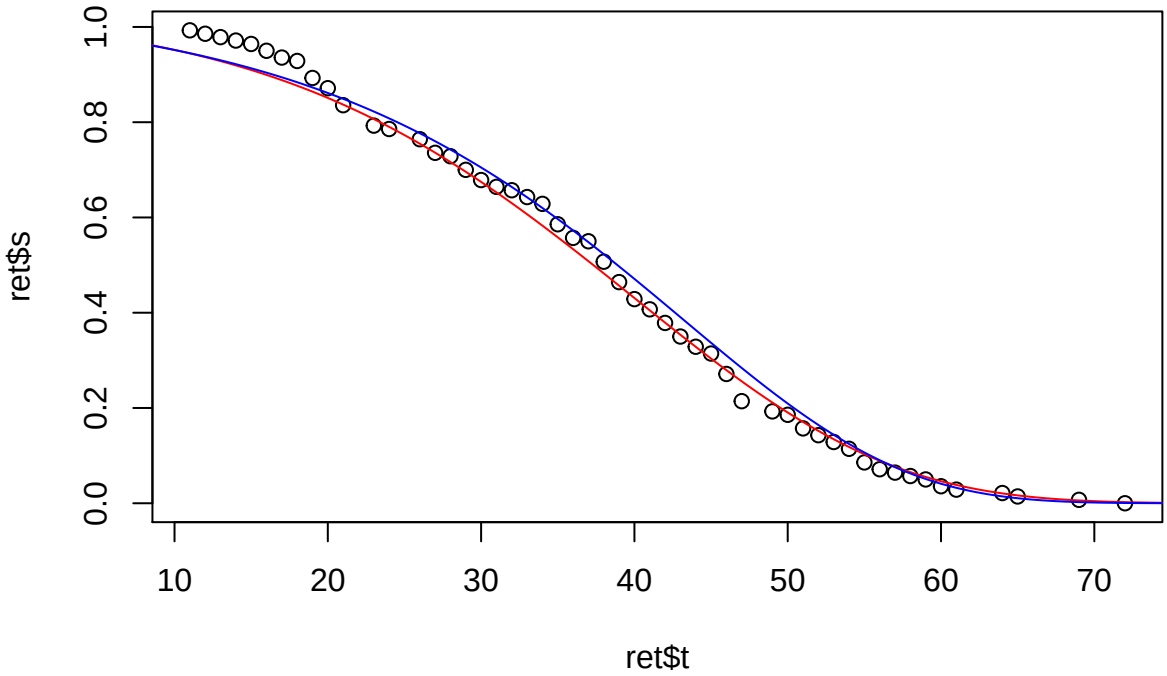

```
##   my.strains samplesize           R       t0        n         G    avgLS
## 3 fig2a.sir2         90 0.002656909 16.80162 8.109348 0.4231348 13.96667
##      stdLS       CV GompGFlex   GompRFlex GompLogLikFlex GompAICFlex
## 3 3.491402 0.249981 0.2797025 0.003555048           -246         496
##   WeibShapeFlex WeibRateFlex WeibLogLikFlex WeibAICFlex
## 3      4.431406     15.29449         -240.4         485
##             R       t0        n         G
## 3 0.002656909 16.80162 8.109348 0.4231348
```



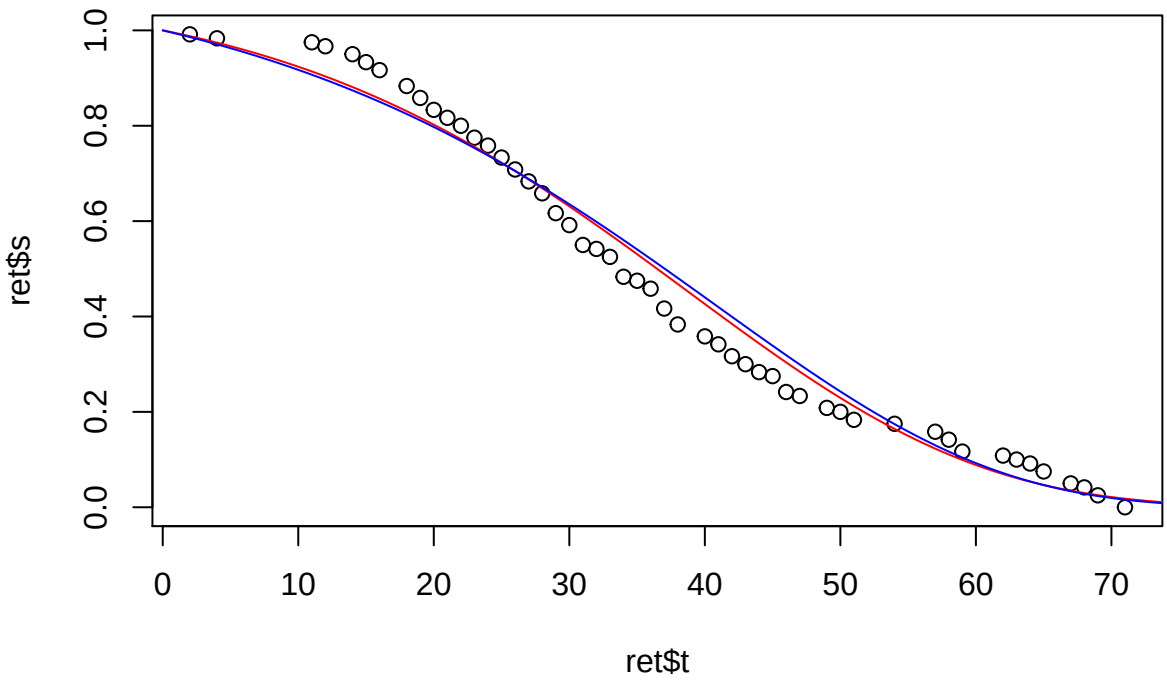

```
##   my.strains samplesize           R       t0        n         G    avgLS
## 4 fig2b.sir2         90 0.002656909 16.80162 8.109348 0.4231348 13.96667
##      stdLS       CV GompGFlex   GompRFlex GompLogLikFlex GompAICFlex
## 4 3.491402 0.249981 0.2797025 0.003555048           -246         496
##   WeibShapeFlex WeibRateFlex WeibLogLikFlex WeibAICFlex
## 4      4.431406     15.29449         -240.4         485
##             R       t0        n         G
## 4 0.002656909 16.80162 8.109348 0.4231348
```



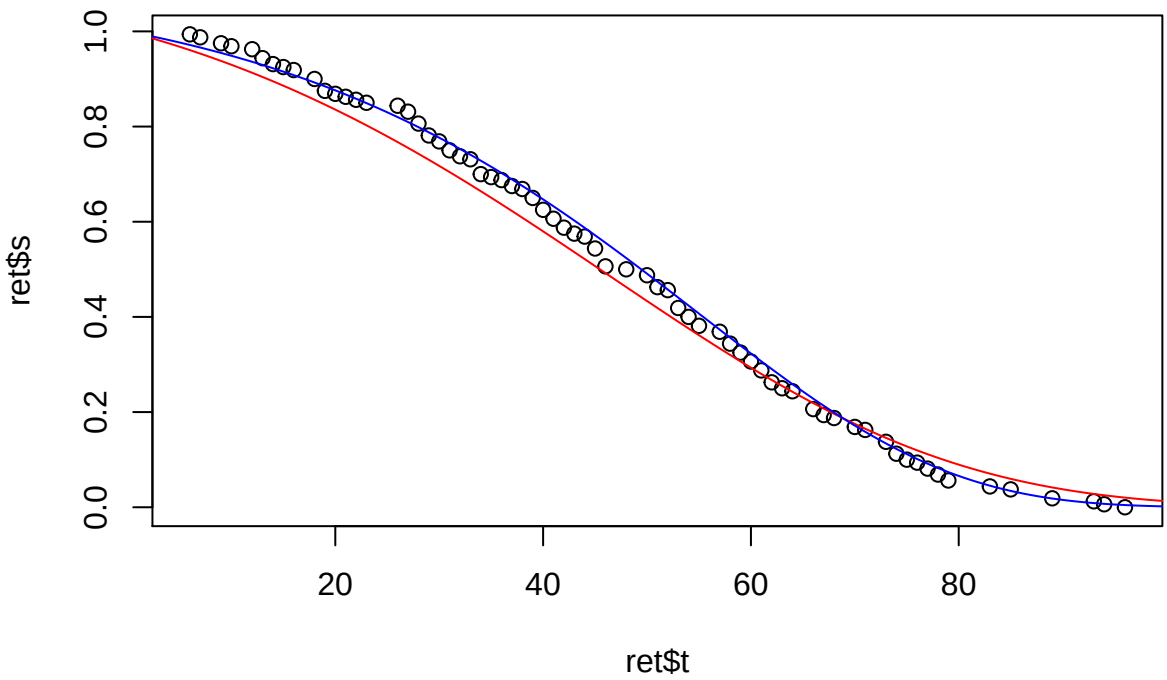

```
##     my.strains samplesize           R       t0        n         G  avgLS
## 5 fig1b.BY4742        250 0.004796452 58.10834 8.109395 0.1223473 26.624
##      stdLS        CV GompGFlex   GompRFlex GompLogLikFlex GompAICFlex
## 5 9.418144 0.3537464 0.0854466 0.006577615         -937.3        1879
##   WeibShapeFlex WeibRateFlex WeibLogLikFlex WeibAICFlex
## 5       3.02464     29.77138           -914        1832
##             R       t0        n         G
## 5 0.004796452 58.10834 8.109395 0.1223473
```




```
##   my.strains samplesize           R      t0        n          G avgLS
## 6 fig1b.fob1        140 0.003056547 71.3747 7.695271 0.09380453 37.75
##      stdLS        CV  GompGFlex   GompRFlex GompLogLikFlex GompAICFlex
## 6 13.40575 0.3551191 0.06987148 0.003424637         -565.5        1135
##   WeibShapeFlex WeibRateFlex WeibLogLikFlex WeibAICFlex
## 6      3.139805     42.25834         -559.6        1123
##             R      t0        n          G
## 6 0.003056547 71.3747 7.695271 0.09380453
```




```
##   my.strains samplesize           R      t0        n         G    avgLS
## 7 fig1b.hxk2        120 0.005736777 103.593 7.578456 0.0635029 36.73333
##     stdLS        CV  GompGFlex   GompRFlex GompLogLikFlex GompAICFlex
## 7 16.3382 0.4447785 0.04823602 0.006725196         -507.6        1019
##   WeibShapeFlex WeibRateFlex WeibLogLikFlex WeibAICFlex
## 7      2.397501       41.424         -502.2        1008
##             R      t0        n         G
## 7 0.005736777 103.593 7.578456 0.0635029
```



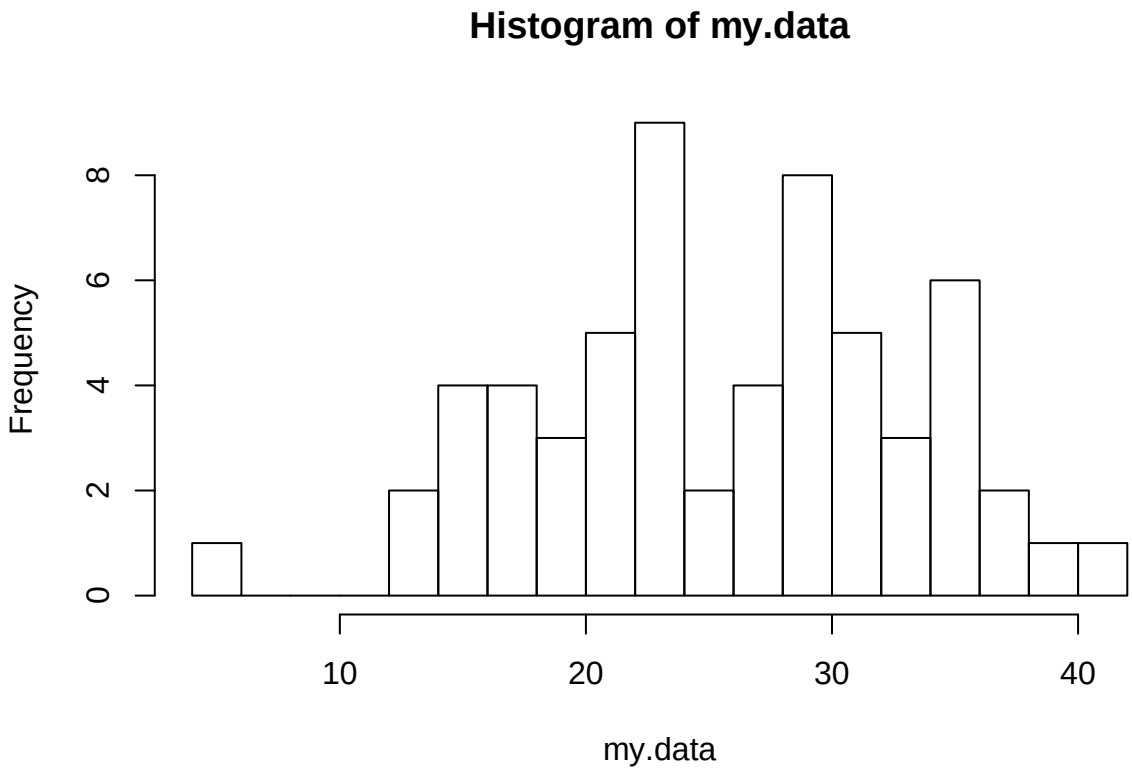

```
##        my.strains samplesize           R       t0        n          G
## 8 fig1b.fob1.hxk2        160 0.005902597 120.6594 6.023612 0.04163466
##      avgLS    stdLS       CV GompGFlex   GompRFlex GompLogLikFlex
## 8 48.28125 21.33771 0.441946 0.0414386 0.004249514         -713.6
##   GompAICFlex WeibShapeFlex WeibRateFlex WeibLogLikFlex WeibAICFlex
## 8        1431      2.441562     54.41078         -714.3        1433
##             R       t0        n          G
## 8 0.005902597 120.6594 6.023612 0.04163466
```

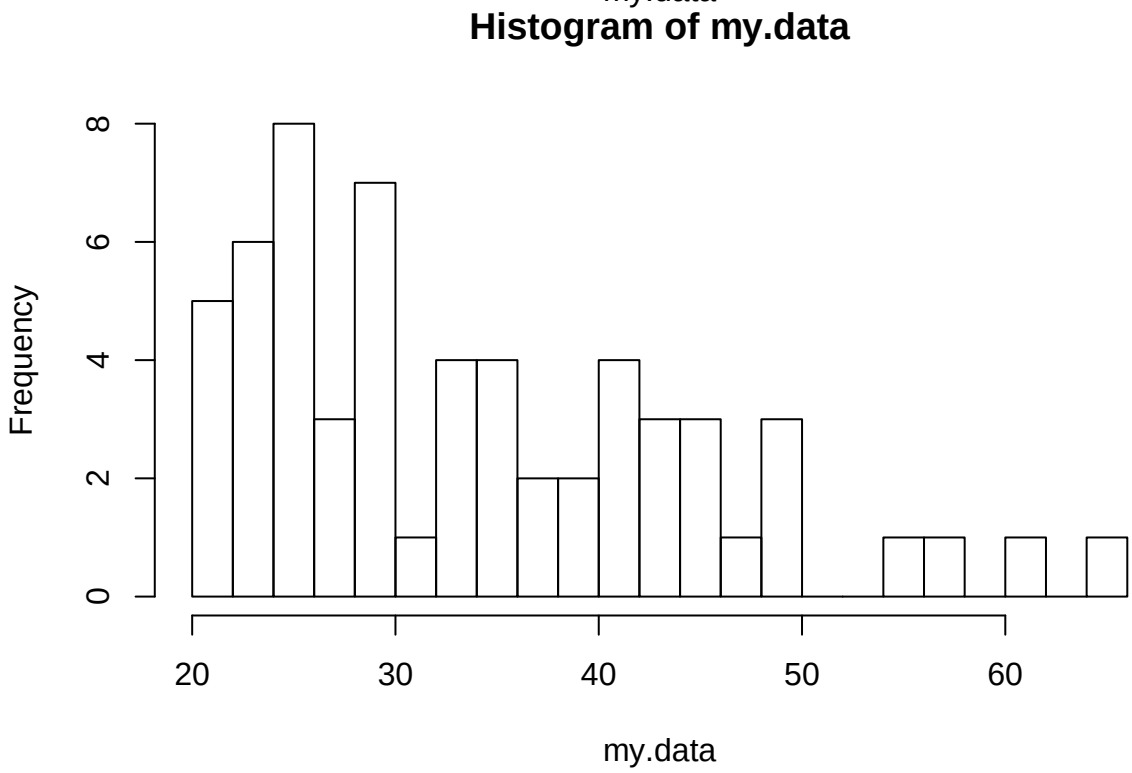
over probility mass function with binomial model

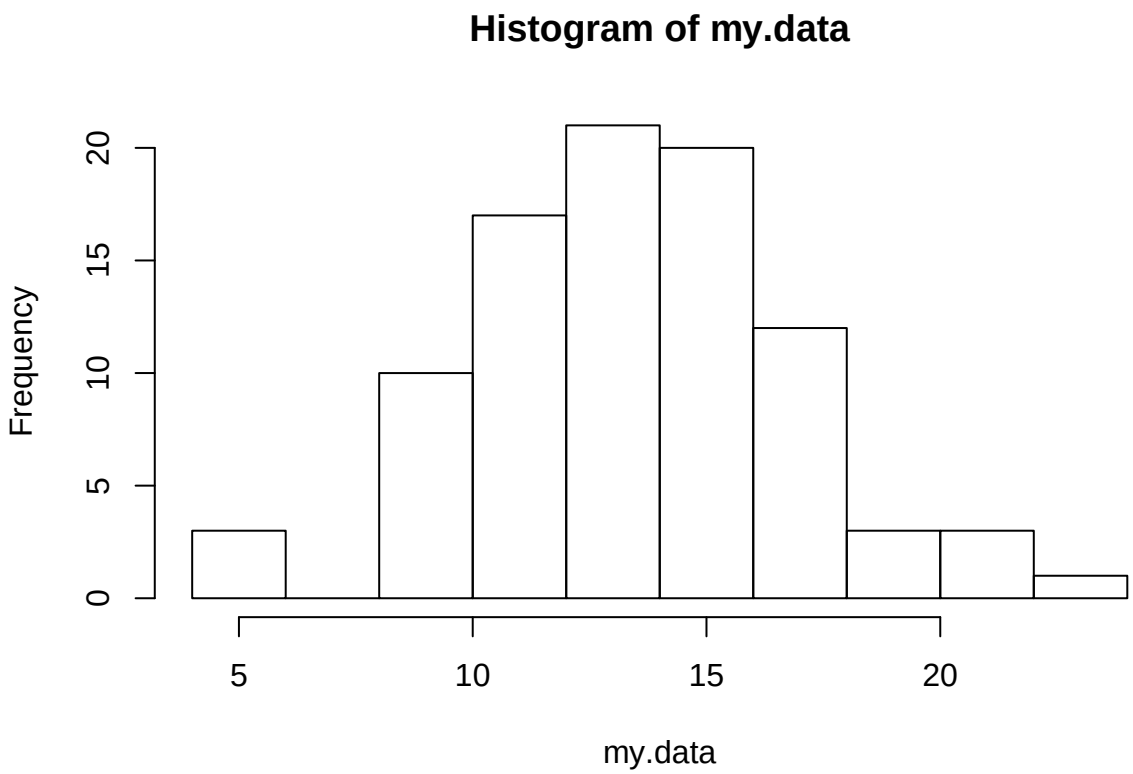
see http://hongqinlab.blogspot.com/2013/12/binomial-mortailty-model.html m = R (1 + t/t0)^(n-1) s = exp((R t0/n)*(1 - (1+t/t0)^n))*
pdf = sm

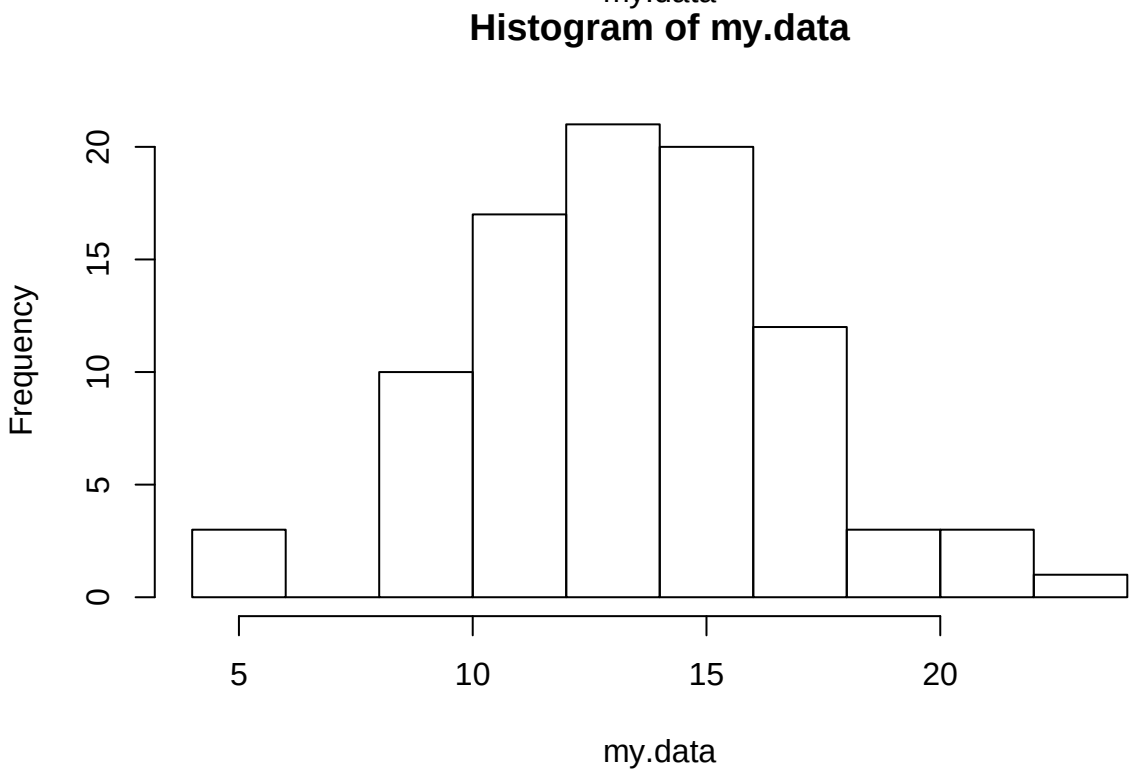
```
for( i in 1:length(report[,1])){
  #report$samplesize[i] = length(tb[,i])
  my.data = tb[,i]
  my.data = my.data[! is.na(my.data)]

  h= hist(my.data, br=max(my.data)/2)
  plot( h$density ~ h$mids, main=my.strains[i], xlab="RLS",ylab="density")
  t= seq(0, max(h$mids), by=0.1)
  s = exp( (report$R[i]* report$t0[i]/report$n[i])*(1 - (1+t/report$t0[i])^report$n[i] ) ) ;
  m = report$R[i]*(1 + t/ report$t0[i])^(report$n[i] -1 )
  pdf = s*m
  lines( pdf ~ t, col='red')
```

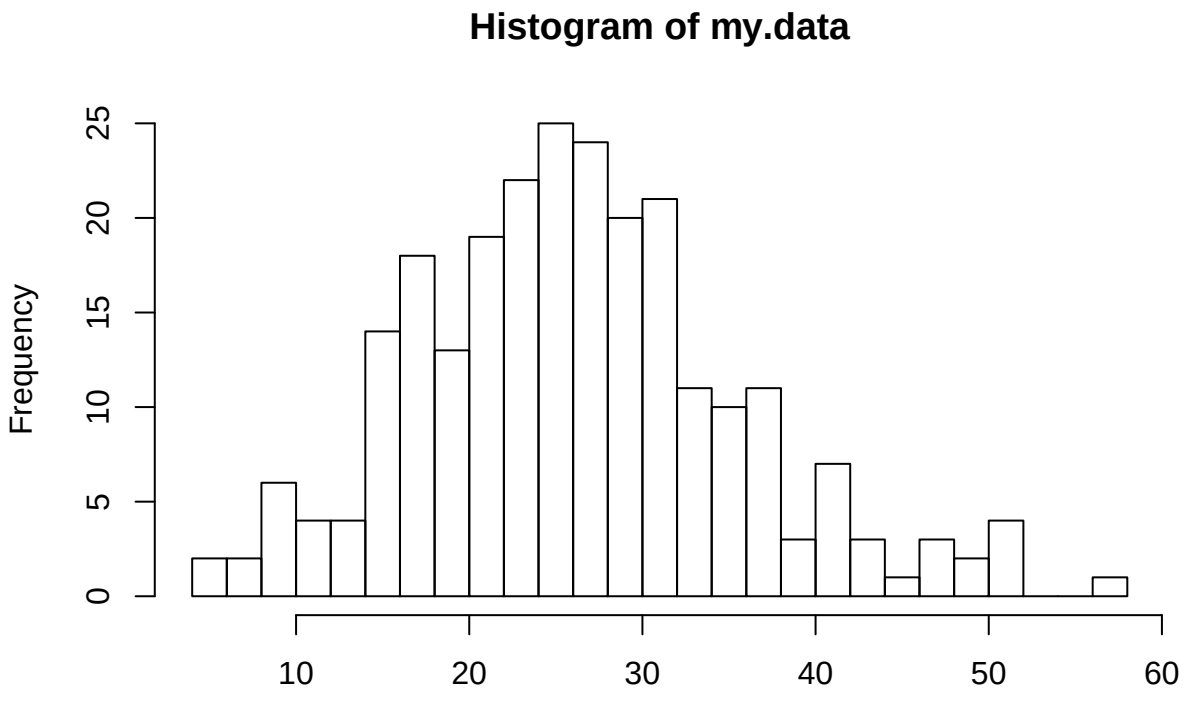
```
  s.g =  G.s( c(report$GompRFlex[i], report$GompGFlex[i]), t );
  m.g =  report$GompRFlex[i]*exp(report$GompGFlex[i]*t)
  pdf.g = s.g * m.g
  lines( pdf.g ~ t, col="blue")
}
```






Histogram of my.data
Frequency
0
5
10
15
20
5
10
15
20
my.data
density
0.00
0.04
0.08
0.12
5


Histogram of my.data
Frequency
0
5
10
15
20
5
10
15
20
my.data
density
0.00
0.04
0.08
0.12
5


Histogram of my.data
Frequency
0
5
10
15
20
25
10
20
30
40
50
60
my.data

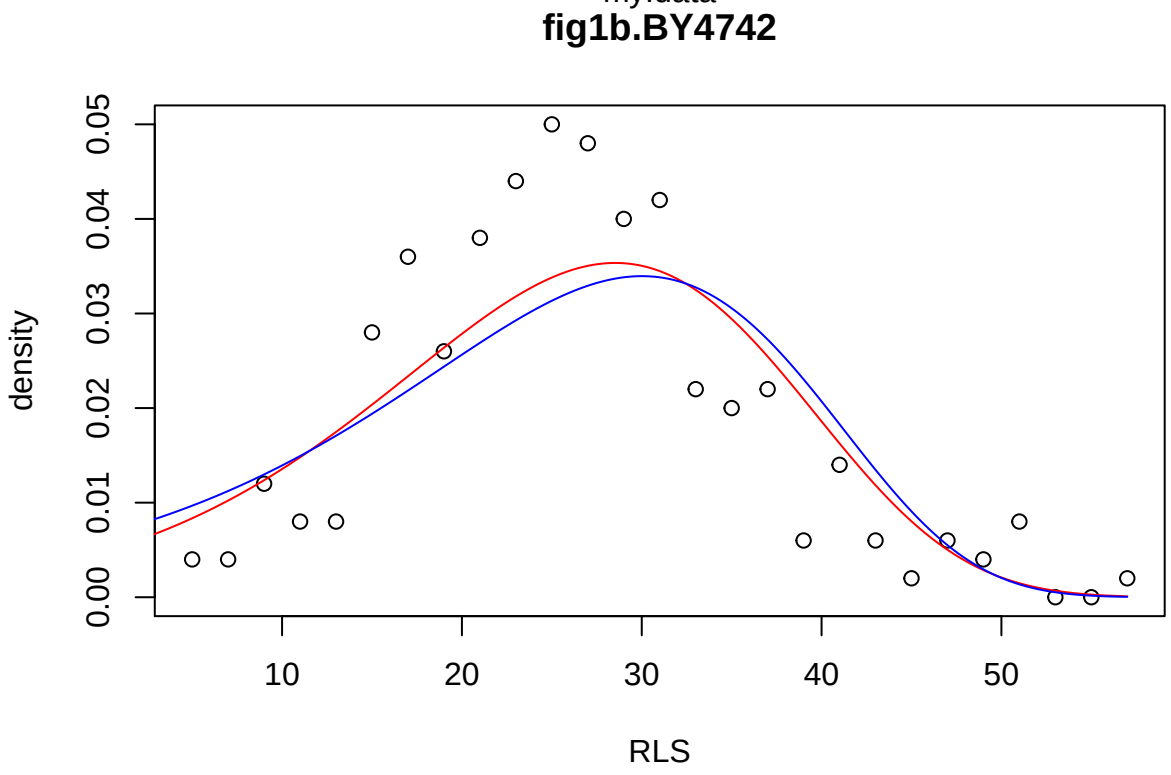
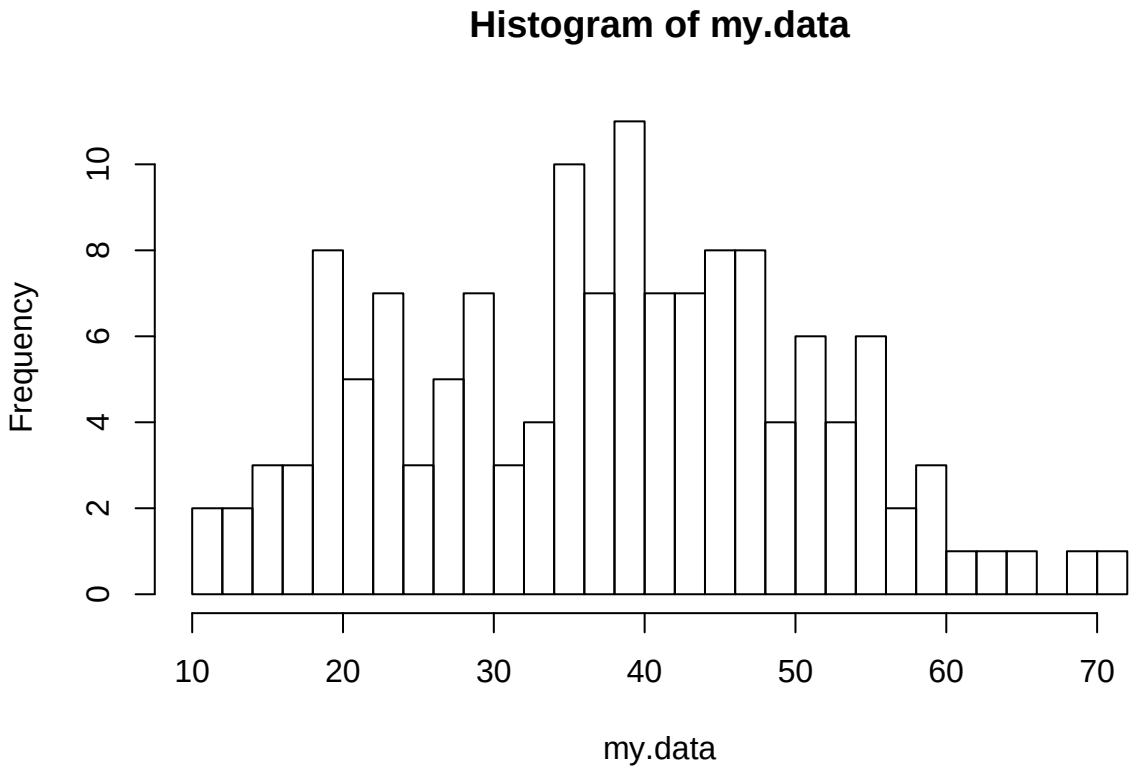
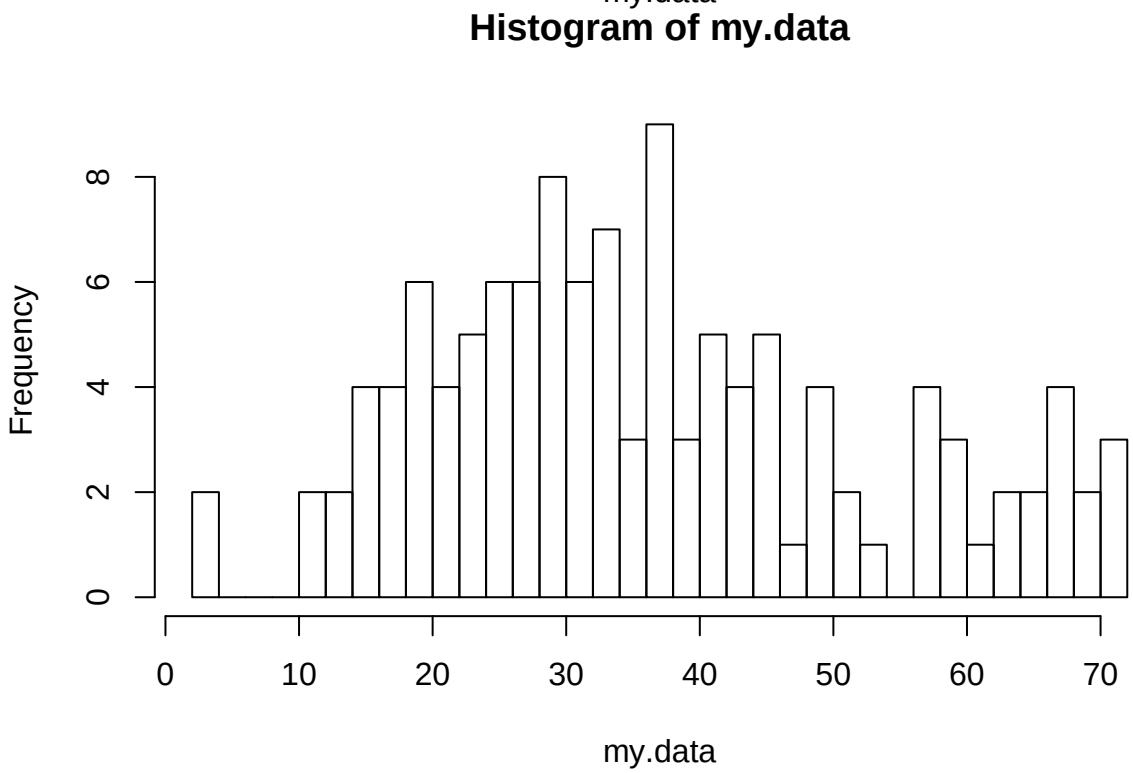
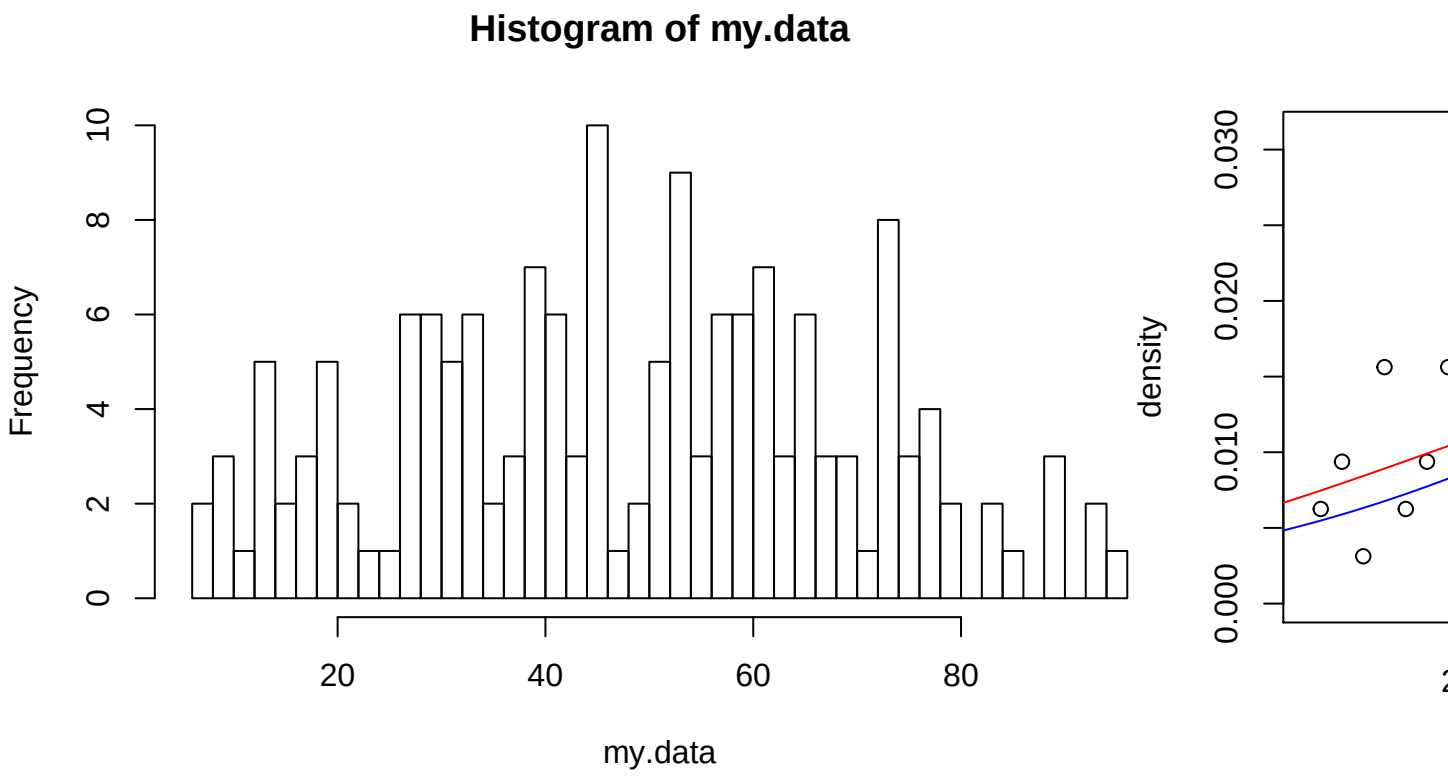


fig1b.BY4742
density
0.00
0.01
0.02
0.03
0.04
0.05
10
20
30
40
50
RLS

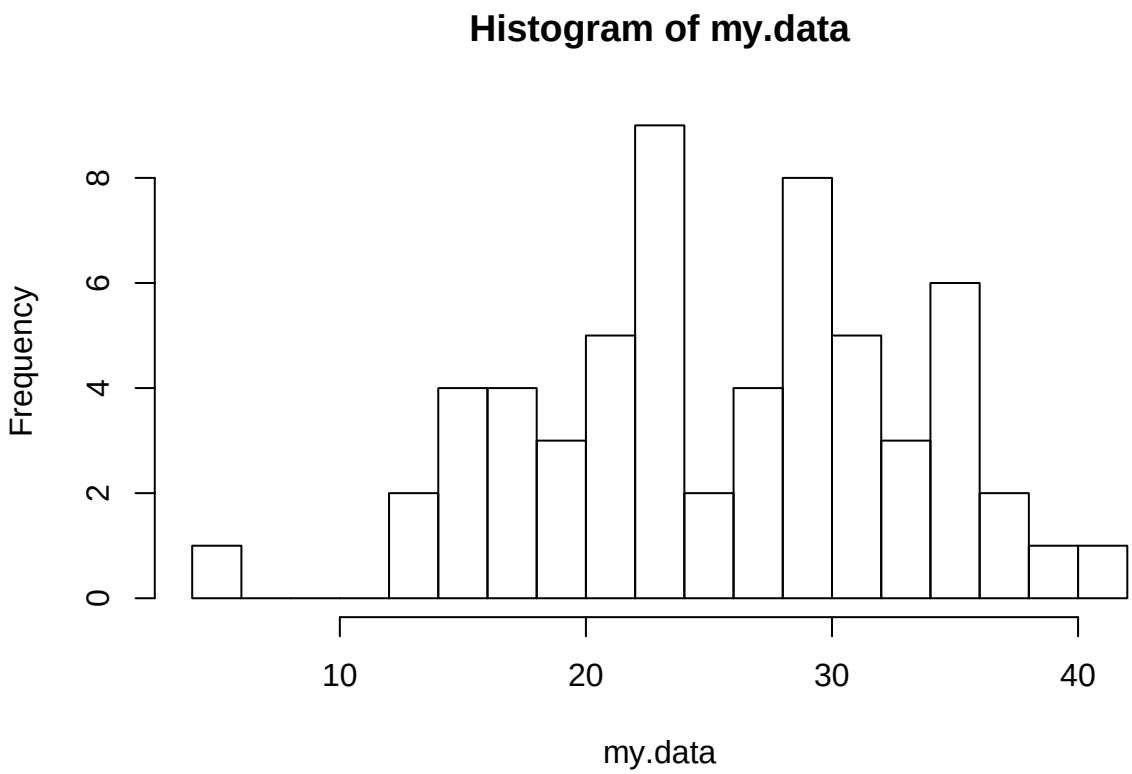

Histogram of my.data
Frequency
0
2
4
6
8
10
10
20
30
40
50
60
70
my.data
density
0.00
0.01
0.02
0.03
0.04
10

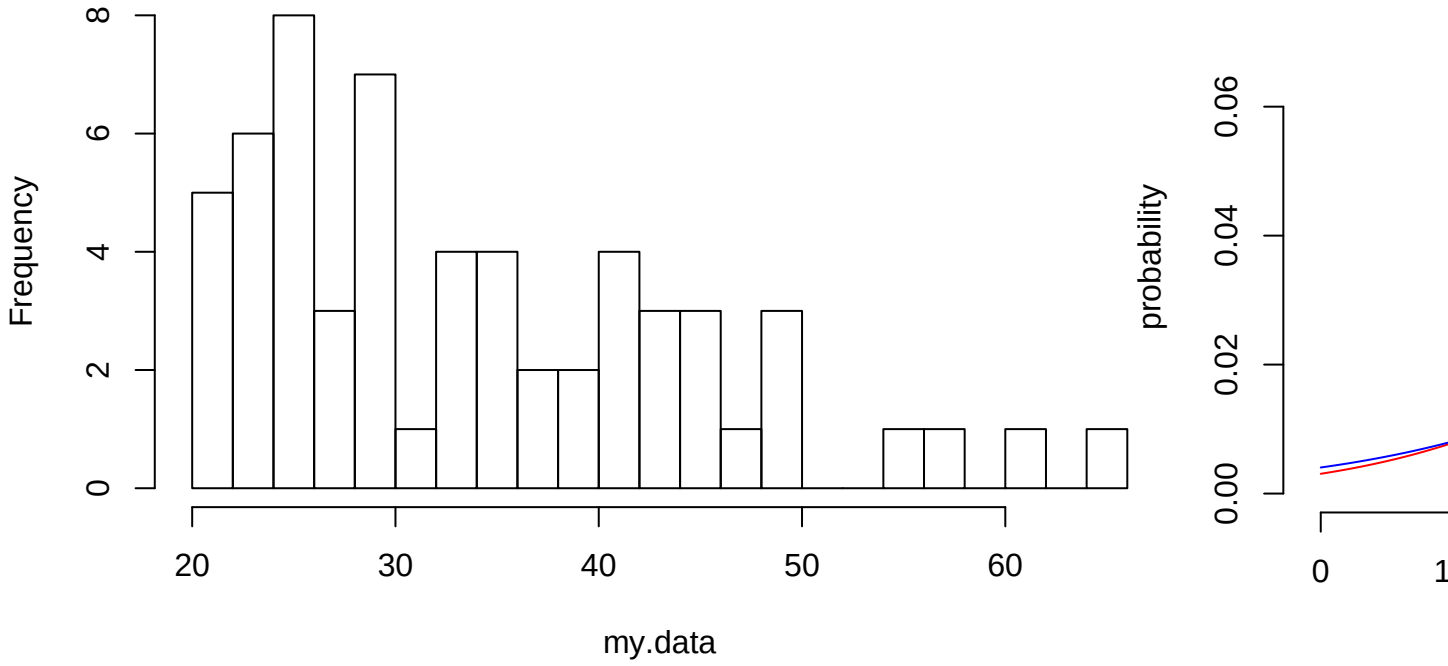

Histogram of my.data
Frequency
0
2
4
6
8
0
10
20
30
40
50
60
70
my.data
density
0.00
0.01
0.02
0.03
10

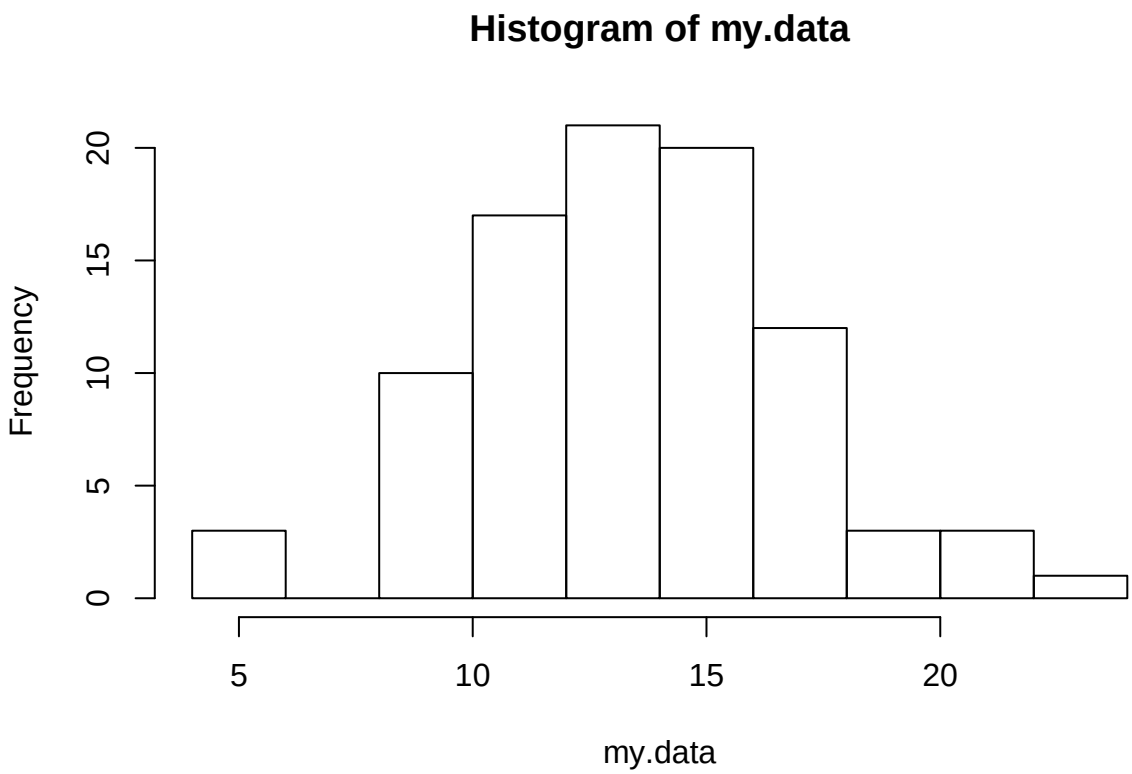


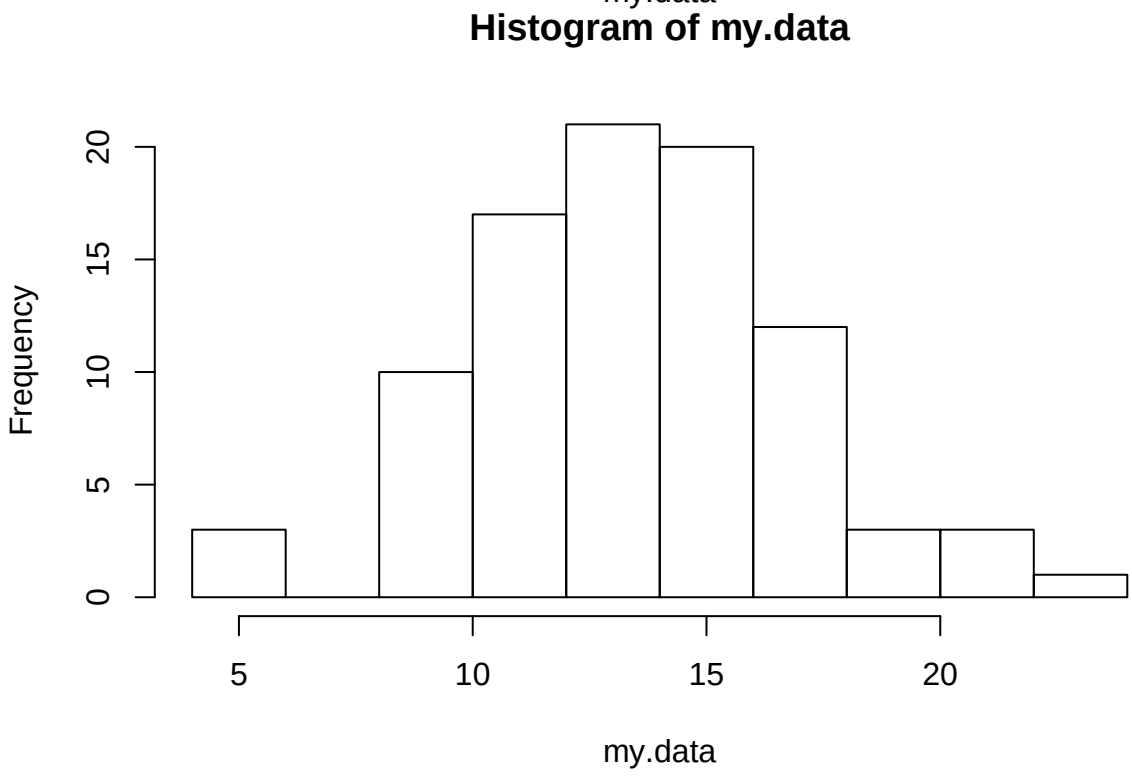
It seems binomial model aging is a reasonable fit whenevern Gompertz model is a reasonable fit.

```
for( i in 1:length(report[,1])){
  my.data = tb[,i]
  my.data = my.data[! is.na(my.data)]

  h= hist(my.data, br=max(my.data)/2);
  hist(my.data, probability = TRUE, col='gray', border='white', xlab='RLS', ylab='probability',
       main=my.strains[i], ylim=c(0, max(h$density)*1.1), xlim=c(0, max(h$mids)*1.1))
  #plot( h$density ~ h$mids, main=my.strains[i], xlab="RLS",ylab="density")
  #par(new=TRUE);
  t= seq(0, max(h$mids)*1.2, by=0.1)
  s = exp( (report$R[i]* report$t0[i]/report$n[i])*(1 - (1+t/report$t0[i])^report$n[i] ) ) ;
  m = report$R[i]*(1 + t/ report$t0[i])^(report$n[i] -1 )
  pdf = s*m
  lines( pdf ~ t, col='red')

  s.g =  G.s( c(report$GompRFlex[i], report$GompGFlex[i]), t );
  m.g =  report$GompRFlex[i]*exp(report$GompGFlex[i]*t)
  pdf.g = s.g * m.g
  lines( pdf.g ~ t, col="blue")
}
```


Histogram of my.data
Frequency
0
2
4
6
8
10
20
30
40
my.data
probability
0.00
0.02
0.04
0.06
0.08
0


Histogram of my.data
Frequency
0
2
4
6
8
20
30
40
50
60
my.data
probability
0.00
0.02
0.04
0.06
0
1


Histogram of my.data
Frequency
0
5
10
15
20
5
10
15
20
my.data
probability
0.00
0.04
0.08
0.12
0


Histogram of my.data
Frequency
0
5
10
15
20
5
10
15
20
my.data
probability
0.00
0.04
0.08
0.12
0

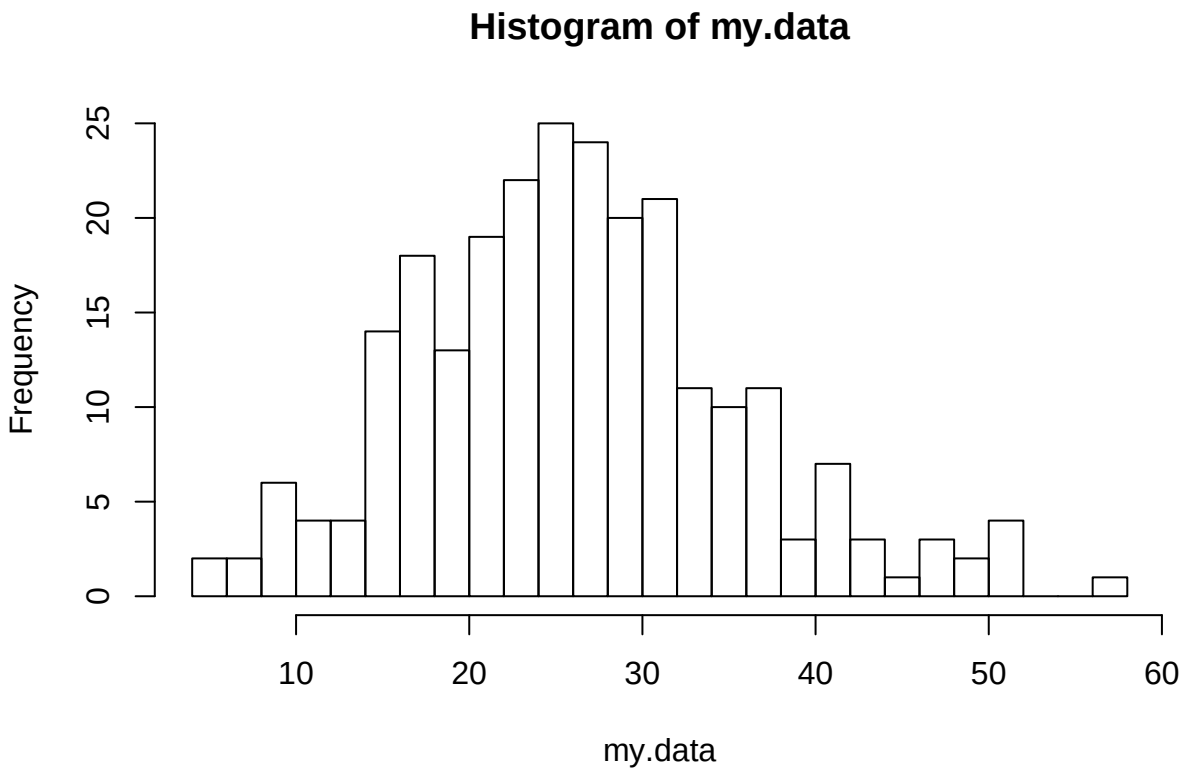
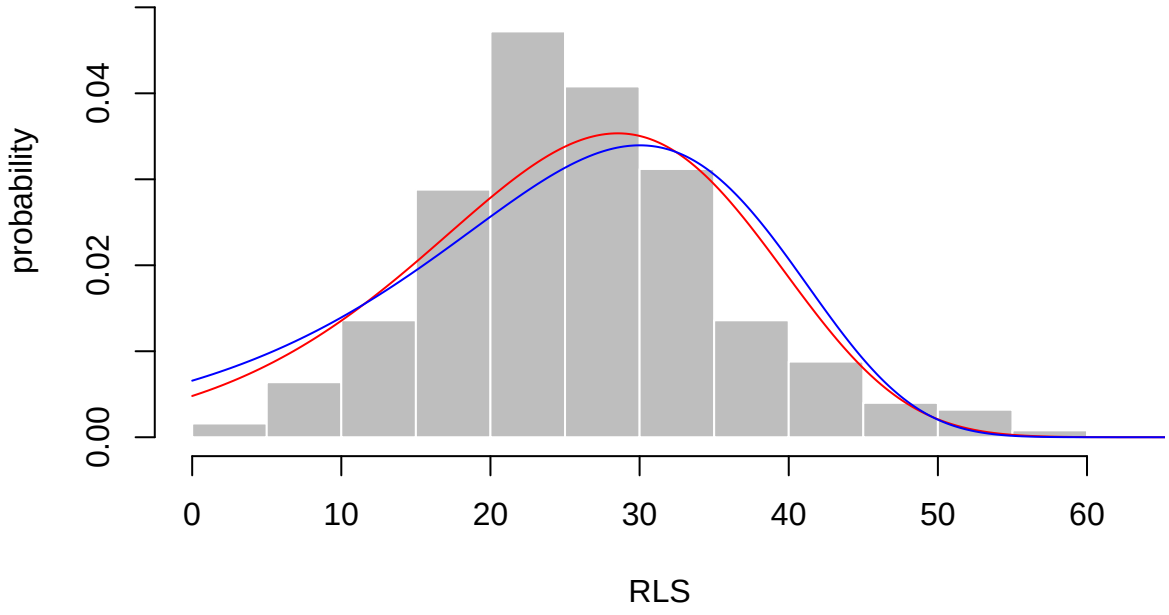
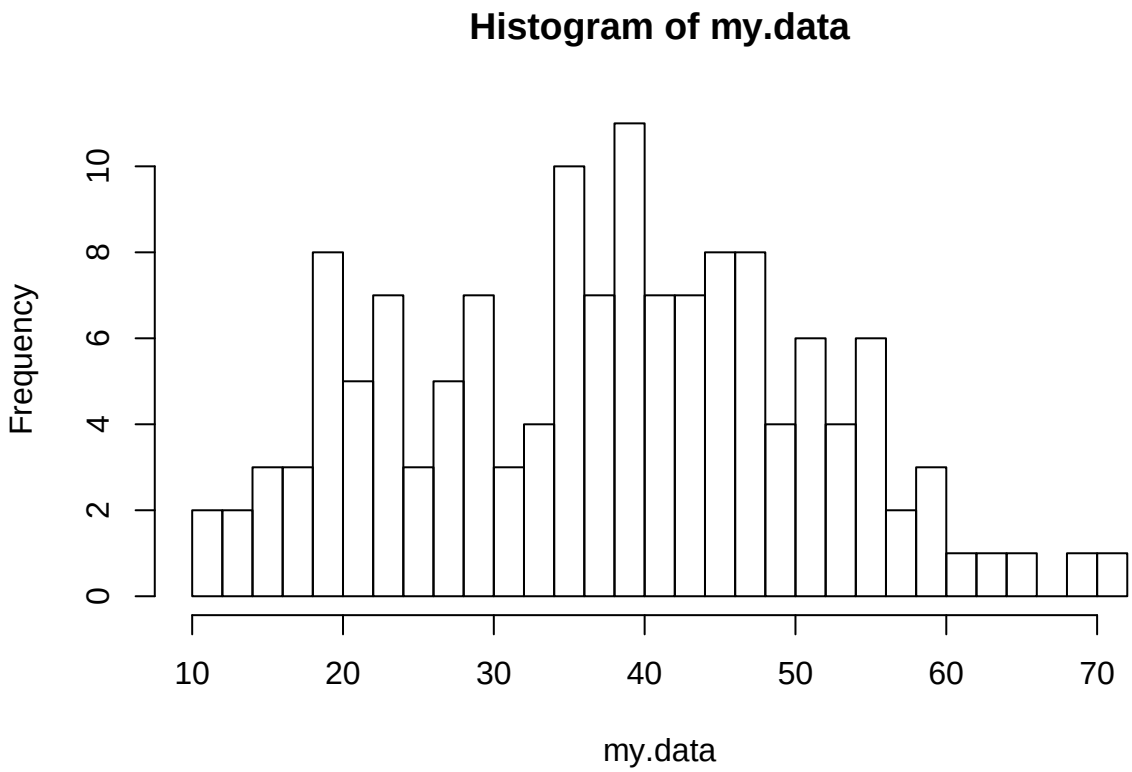
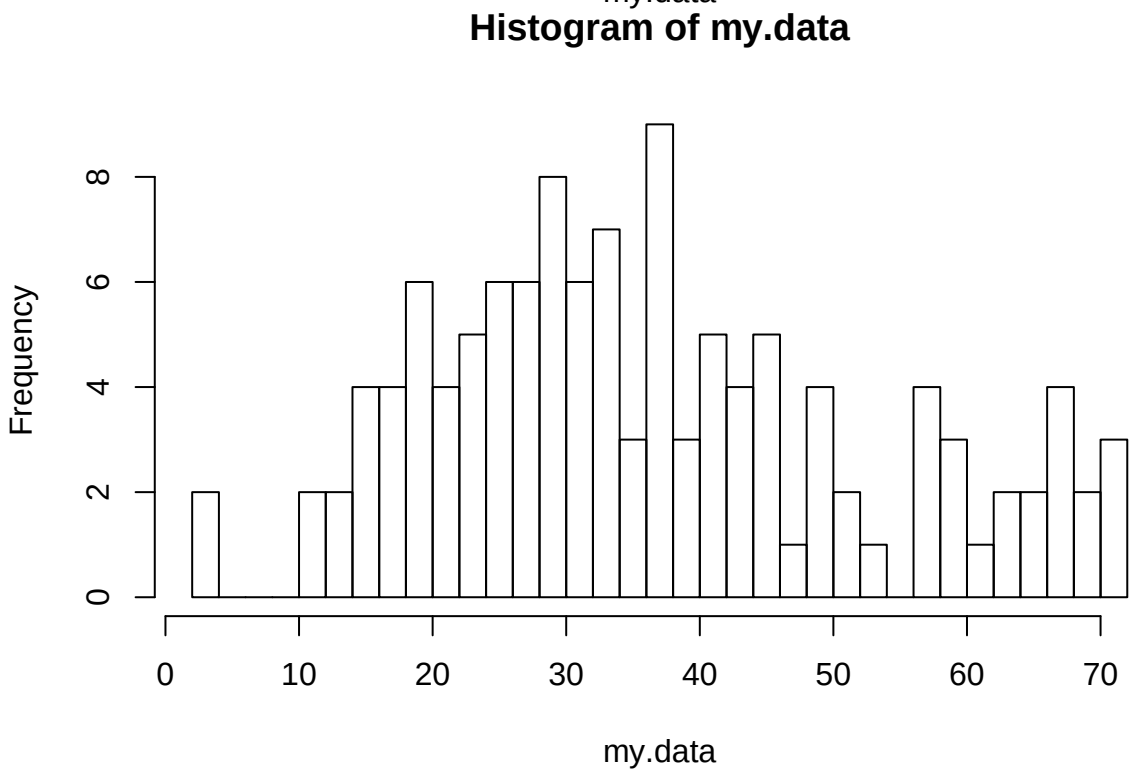
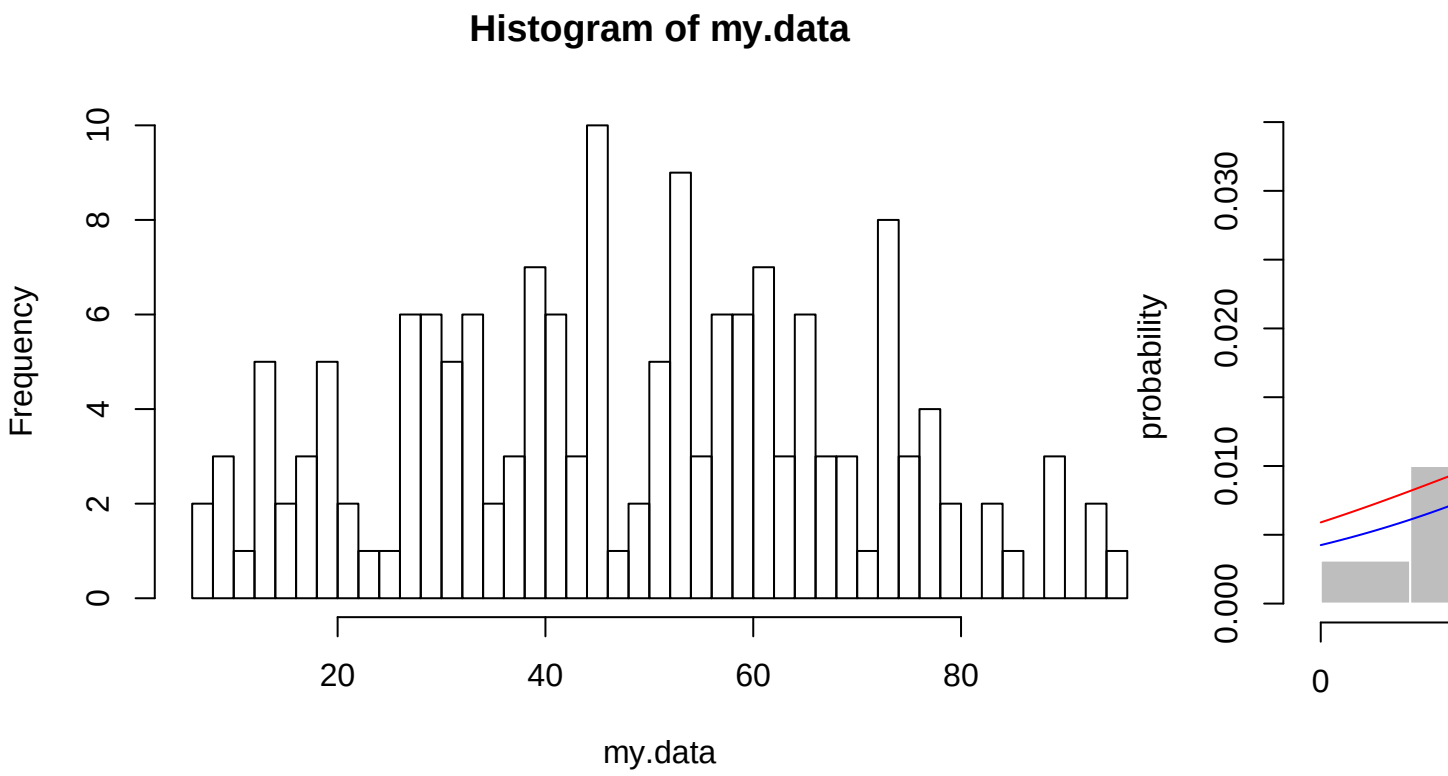


fig1b.BY4742




Histogram of my.data
Frequency
0
2
4
6
8
10
10
20
30
40
50
60
70
my.data
probability
0.00
0.01
0.02
0.03
0.04
0


Histogram of my.data
Frequency
0
2
4
6
8
0
10
20
30
40
50
60
70
my.data
probability
0.00
0.01
0.02
0.03
0.04
0


Histogram of my.data
Frequency
0
2
4
6
8
10
20
40
60
80
my.data
probability
0.000
0.010
0.020
0.030
0