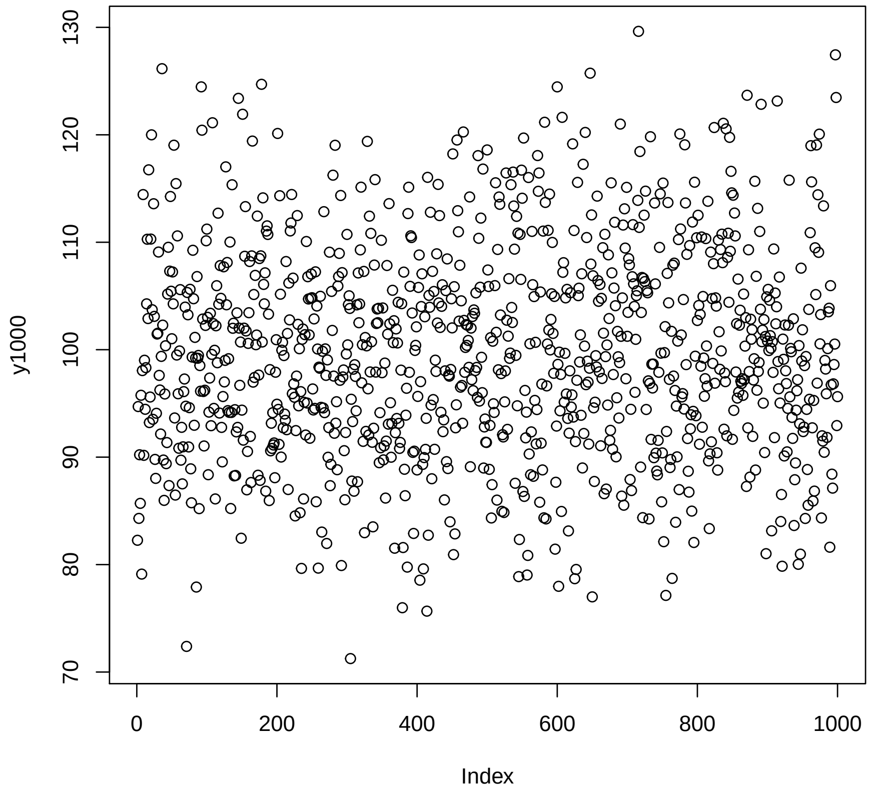
从现在开始我决定认真的学习 Bayesian 方法，以及 R 语言的 LaplacesDemon 包。那么我们就从平均值 mean 开始，慢慢的一步一步的来吧。

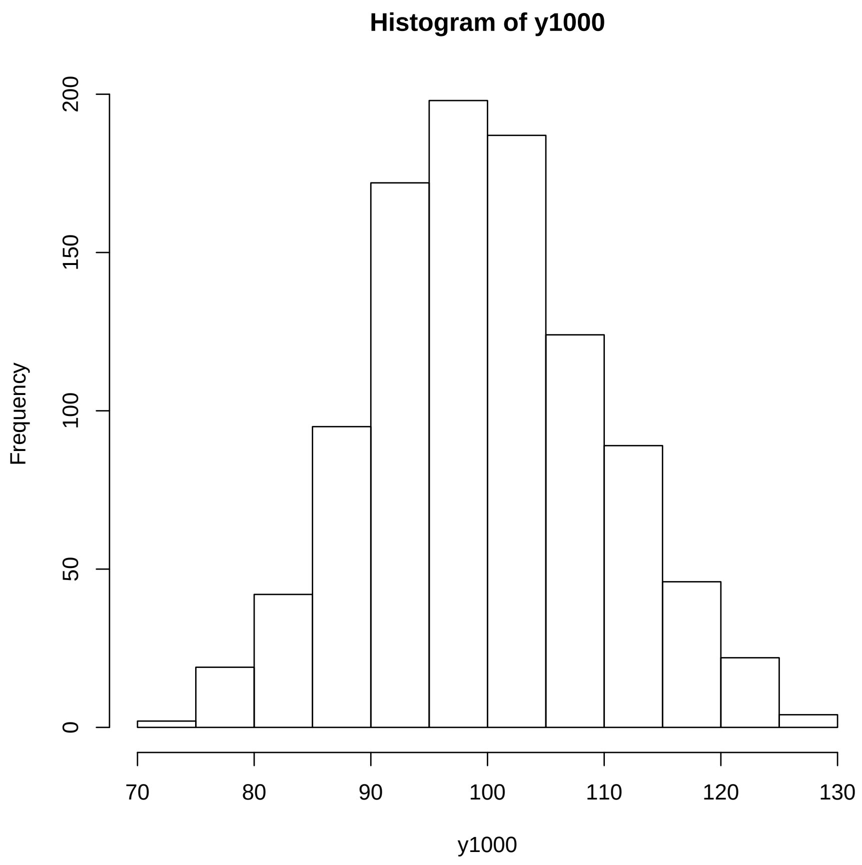
第一章：用暴力的方法求平均数和方差。

1. 创建 y1000，平均值为 100，标准误为 10 的正态数据组，共 1000 个数据。

```r
# Simple normal mean model in LaplacesDemon Generate two samples of body
# mass measurements of male peregrines
y1000 <- rnorm(n = 1000, mean = 100, sd = 10)  # Sample of 1000 birds
##
plot(y1000)
```



```r
hist(y1000)
```

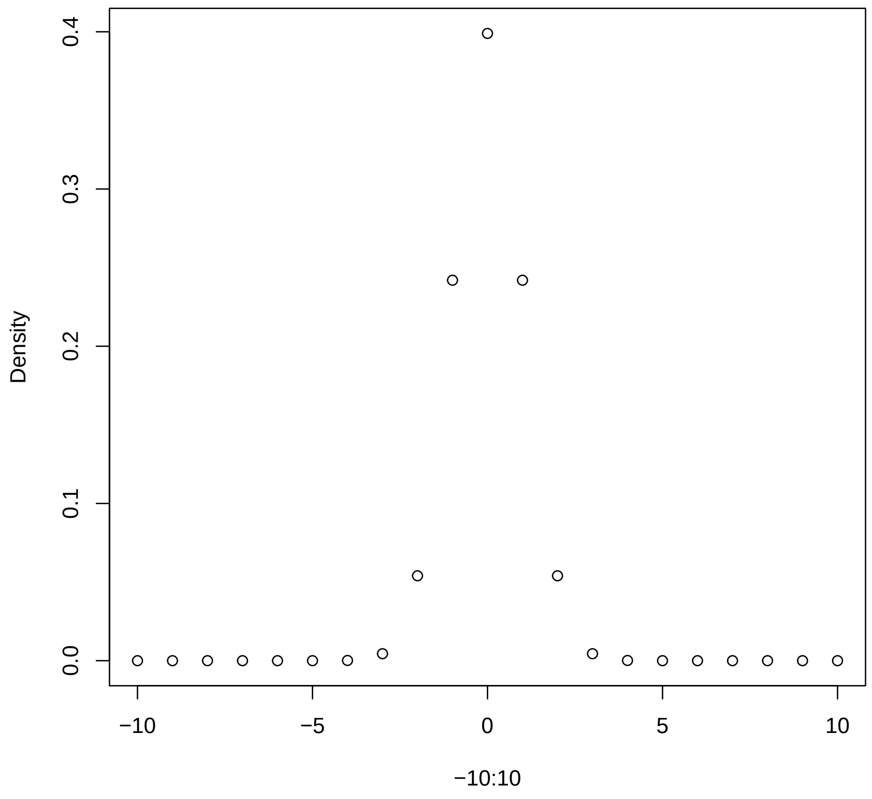


```r
##
mean(y1000)

## [1] 99.73

sd(y1000)

## [1] 9.842

##
lm0 <- lm(y1000 ~ 1)
summary(lm0)
```

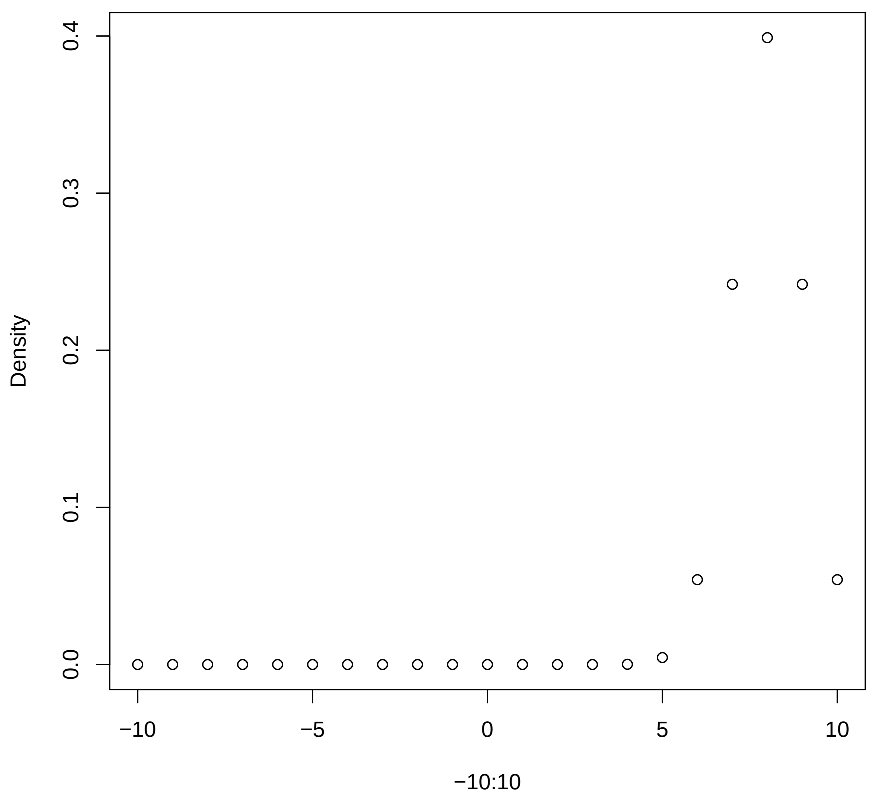
```
##
## Call:
## lm(formula = y1000 ~ 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.477  -6.959  -0.464   6.205  29.901
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   99.728      0.311     320   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.84 on 999 degrees of freedom

##
```

先了解一下密度函数是怎么一回事

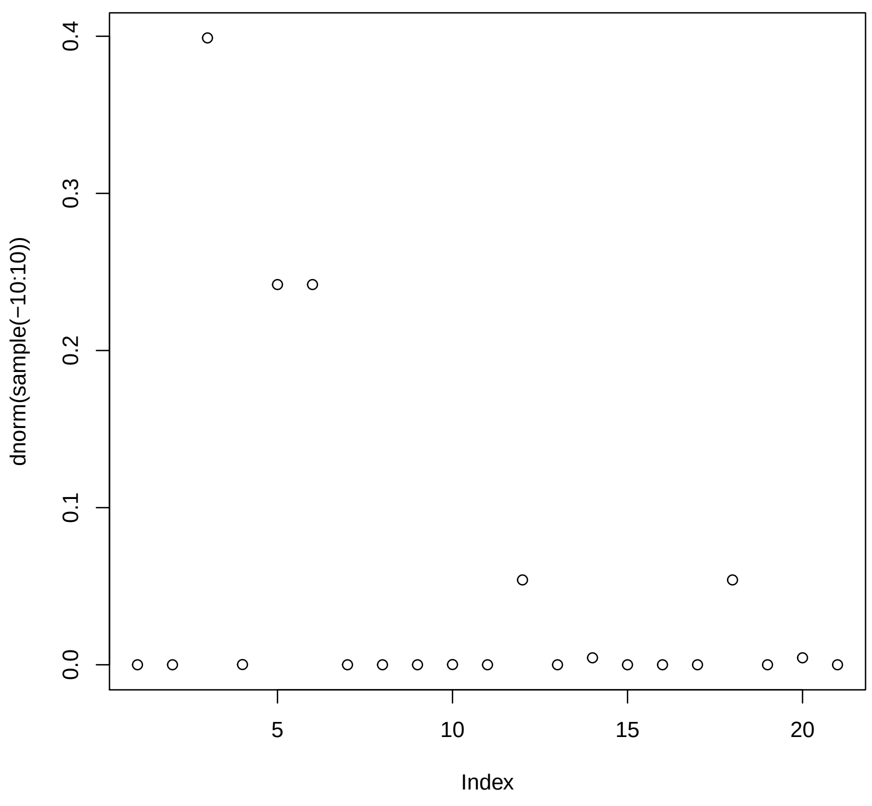
```
plot(x = -10:10, y = dnorm(-10:10, mean = 0, sd = 1), ylab = "Density")
```



```r
plot(x = -10:10, y = dnorm(-10:10, mean = 8, sd = 1), ylab = "Density")
```

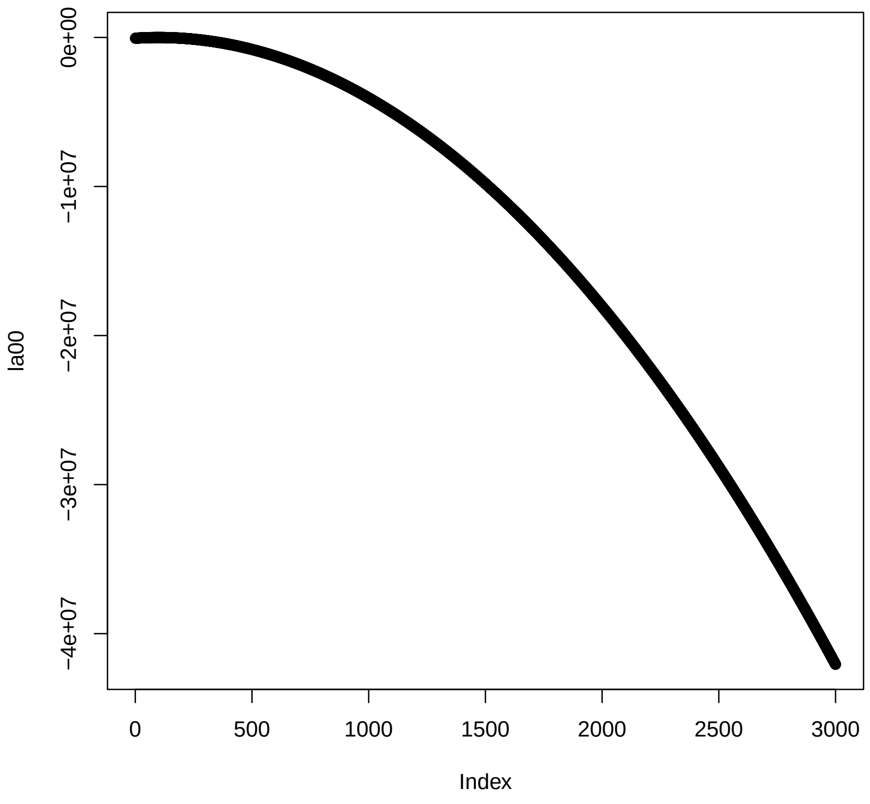


```
sum(dnorm(-10:10, mean = 0, sd = 1))

## [1] 1

sum(dnorm(-10:10, mean = 8, sd = 1))

## [1] 0.9954

#
plot(dnorm(sample(-10:10)))
```

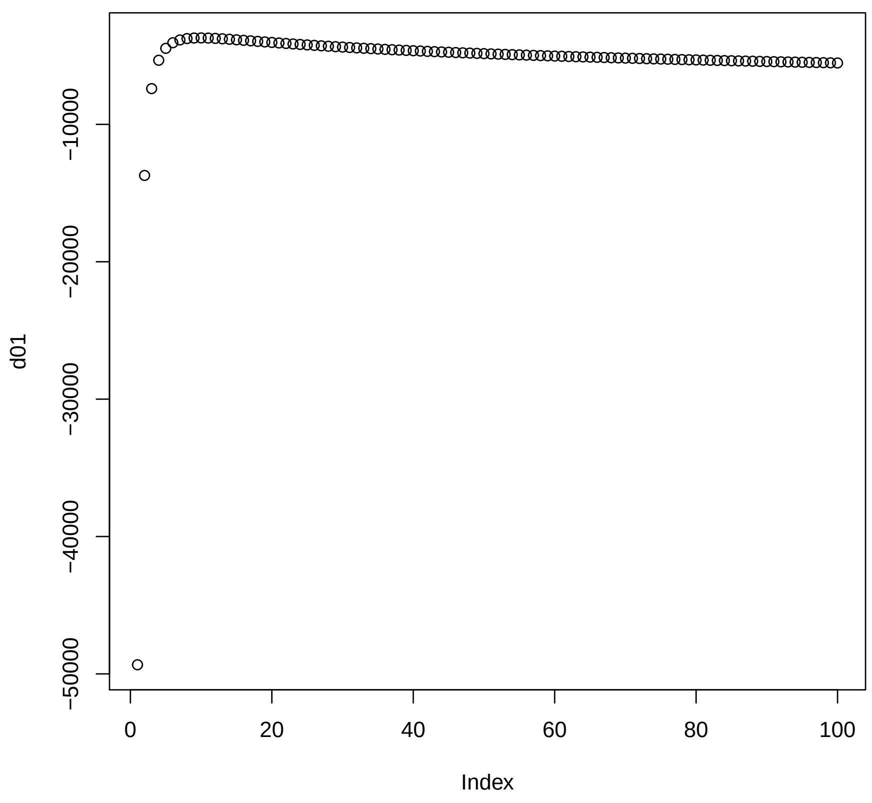


如果假设的正态分布组的方差是 1，平均数是 1:3000，哪个平均数最有可能，使实际数据的密度函数之和最大呢？如果假设的正态分布组的平均数是 100，方差是 1:100，哪个方差最有可能，使实际数据的密度函数之和最大呢？重复 50 次之后的结果是不是可靠呢？

```r
population.sd <- 1
for (i in 1:50) {
    mu <- 1:3000
   la00 <- sapply(mu, function(xx) sum(dnorm(y1000, xx, population.sd, log = TRUE)))
    mu <- mu[which.max(la00)]
    population.sd <- 1:100
   d01 <- sapply(population.sd, function(xx) sum(dnorm(y1000, mu, xx, log = TRUE)))
    population.sd <- population.sd[which.max(d01)]
}
c(mean = mu, sd = population.sd)

## mean   sd
##  100   10

#
plot(la00)
```



```
plot(d01)
```

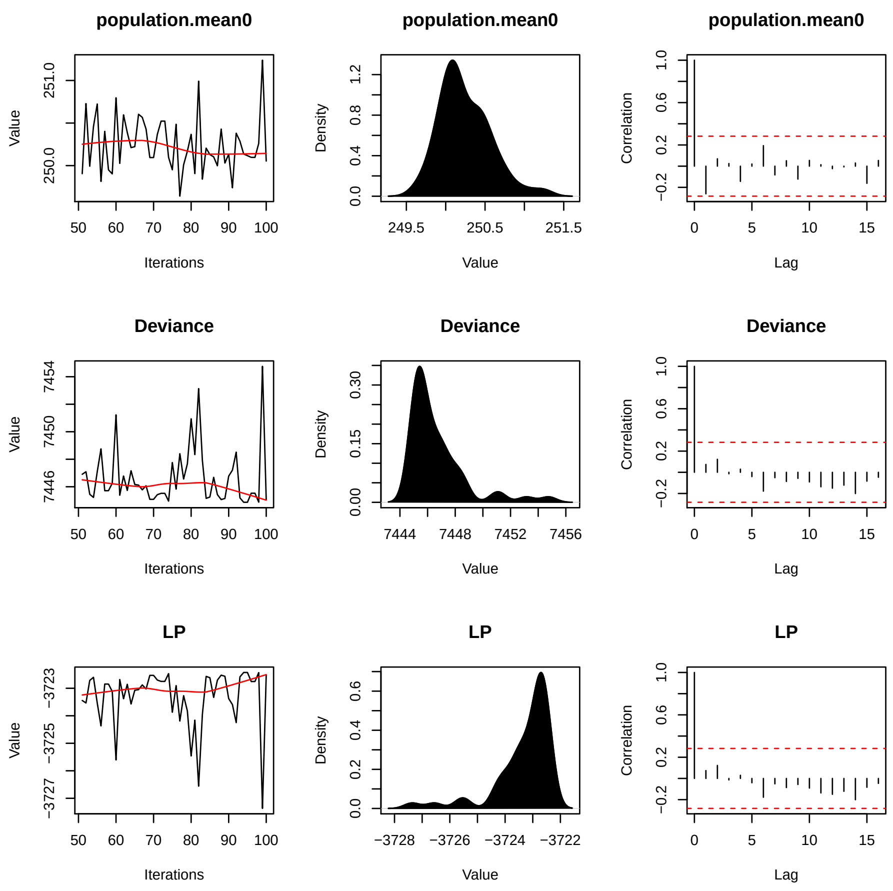


来看看 R 语言的 LaplacesDemon 包是怎么将暴力进行到底的呢？Model function 就相当于一个 for 循环。那么就让 Model 循环 10000 次吧。

```
# Load library
library(LaplacesDemon)

## Loading required package:  parallel

y1000 <- rnorm(n = 1000, mean = 250, sd = 10)  # Sample of 1000 birds
## Model specification
Model <- function(parm, Data) {
    # Parameters
    population.mean <- parm[1]
    population.sd <- parm[2]
    # Prior density
    population.mean.prior <- dunif(population.mean, 0, 5000)
    population.sd.prior <- dunif(population.sd, 0, 100)
    # Log-Likelihood
    mu <- population.mean
    LL <- sum(dnorm(Data$mass, mu, population.sd, log = TRUE))
    # Log-Posterior
    LP <- LL + population.mean.prior + population.sd.prior
   Modelout <- list(LP = LP, Dev = -2 * LL, Monitor = c(LP), yhat = rnorm(Data$N,
        mu, population.sd), parm = parm)
    return(Modelout)
}
# Prepare the data
parm.names <- c("population.mean0", "population.sd0")
Data <- list(mass = y1000, N = length(y1000), mon.names = c("LP"), parm.names = parm.names)
# Initial values
Initial.Values <- c(1000, -250)
# MCMC settings
ni <- 5000  # Number of draws from posterior (for each chain)
st <- 1000  # Steps when status message should be given
nt <- 50  # Thinning rate #  Abate autocorrelation
# Run LaplacesDemon
out <- LaplacesDemon(Model, Data = Data, Initial.Values, Iterations = ni, Status = st,
    Thinning = nt)

##
## Laplace's Demon was called on Wed Oct 30 18:04:56 2013
##
## Performing initial checks...

## Warning:  NaNs produced
## Warning:  NAs produced

## Generating initial values due to a non-finite posterior.

## Warning:  NaNs produced
## Warning:  NAs produced
## Warning: NaNs produced
## Warning: NAs produced
## Warning: NaNs produced
## Warning: NAs produced
## Warning: NaNs produced
## Warning: NAs produced
```

```
# Have a look at some summary statistics
out
```

```
.Values,
## Covariance Matrix: (NOT SHOWN HERE; diagonal shown instead)
## population.mean0   population.sd0
##            2.835            2.835
##
## Covariance (Diagonal) History: (NOT SHOWN HERE)
## Deviance Information Criterion (DIC):
##          All Stationary
## Dbar    7956     7449.2
## pD   1433069      278.3
## DIC  1441024     7727.5
##
## Delayed Rejection (DR): 0
## Initial Values:
## [1] 42.79 51.29
##
## Iterations: 5000
## Log(Marginal Likelihood): NA
## Minutes of run-time: 0.02
## Model: (NOT SHOWN HERE)
## Monitor: (NOT SHOWN HERE)
## Parameters (Number of): 2
## Periodicity: 5001
## Posterior1: (NOT SHOWN HERE)
## Posterior2: (NOT SHOWN HERE)
## Recommended Burn-In of Thinned Samples: 10
## Recommended Burn-In of Un-thinned Samples: 500
## Recommended Thinning: 400
## Status is displayed every 1000 iterations
## Summary1: (SHOWN BELOW)
## Summary2: (SHOWN BELOW)
## Thinned Samples: 100
## Thinning: 50
##
##
## Summary of All Samples
##                     Mean      SD   MCSE    ESS       LB  Median     UB
## population.mean0  239.16  39.27  14.56  7.461   86.407   250.2   250.8
## population.sd0     18.86  28.52  11.47 12.477    9.633    10.1   118.7
## Deviance        7955.58 1692.97 666.52 11.773 7444.874  7446.4 13105.7
## LP             -3977.78  846.49 333.26 11.773 -6552.836 -3723.2 -3722.4
##
##
## Summary of Stationary Samples
##                        Mean      SD      MCSE ESS        LB   Median
## population.mean0   250.23  0.4124   0.04023  90   249.668   250.22
## population.sd0      10.11  0.4844   0.05074  90     9.624    10.06
## Deviance          7449.19 23.5941   2.45715  90  7444.873  7446.20
## LP               -3724.59 11.7971 333.26228  90 -3726.344 -3723.09
##                         UB
## population.mean0   250.78
## population.sd0      10.58
## Deviance          7452.71
## LP               -3722.43

# Plotting output
plot(out, BurnIn = 50, Data, Parms = (".mean0"))
```



第二章：关于平均值、T 检验、线性回归、单变量方差分析、双变量方差分析、协方差分析

数据：mass 生物量, pop 种群, region 范围, hab 栖息地, svl 体长

```
mass <- c(6, 8, 5, 7, 9, 11)
pop <- factor(c(1, 1, 2, 2, 3, 3))
region <- factor(c(1, 1, 1, 1, 2, 2))
hab <- factor(c(1, 2, 3, 1, 2, 3))
svl <- c(40, 45, 39, 50, 52, 57)
```

平均值：$\text{mass}_i = + _i, i \; Normal(0, ^2)$

```
# mean
lm(mass ~ 1)  #  massi = + i ,     i~Normal(0, ^2)

##
## Call:
## lm(formula = mass ~ 1)
##
## Coefficients:
## (Intercept)
##        7.67

model.matrix(mass ~ 1)

##   (Intercept)
## 1           1
## 2           1
## 3           1
## 4           1
## 5           1
## 6           1
## attr(,"assign")
## [1] 0
```

t 检验：$\text{mass}_i = + * region_i + _i ; _i \ Normal(0, ^2) mass_i \ Normal(+ * region, ^2)$

c(6, 8, 5, 7, 9, 11) = * (1, 1, 1, 1, 1, 1) + * factor(c(1,1,1,1,2,2)) + c(1, 2, 3, 4, 5, 6)

```
# t-test
lm(mass ~ region)  #

##
## Call:
## lm(formula = mass ~ region)
##
## Coefficients:
## (Intercept)      region2
##         6.5          3.5
```

```r
model.matrix(~region)

##   (Intercept) region2
## 1           1       0
## 2           1       0
## 3           1       0
## 4           1       0
## 5           1       1
## 6           1       1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$region
## [1] "contr.treatment"

summary(lm(mass ~ region))

##
## Call:
## lm(formula = mass ~ region)
##
## Residuals:
##    1    2    3    4    5    6
## -0.5  1.5 -1.5  0.5 -1.0  1.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.500      0.661    9.83   0.0006 ***
## region2        3.500      1.146    3.06   0.0378 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.32 on 4 degrees of freedom
## Multiple R-squared:   0.7,Adjusted R-squared:  0.625
## F-statistic: 9.33 on 1 and 4 DF,  p-value: 0.0378

# mass_i = + *region_i+ _i  _i~Normal(0, ^2) mass_i~Normal( + *region, ^2)
# c(6, 8, 5, 7, 9, 11) =  * (1, 1, 1, 1, 1, 1) +  * factor(c(1,1,1,1,2,2))
# + c( 1, 2, 3, 4, 5, 6)
lm(mass ~ region - 1)

##
## Call:
## lm(formula = mass ~ region - 1)
##
## Coefficients:
## region1  region2
##     6.5     10.0

model.matrix(~region - 1)

##   region1 region2
## 1       1       0
## 2       1       0
## 3       1       0
## 4       1       0
## 5       0       1
## 6       0       1
## attr(,"assign")
## [1] 1 1
## attr(,"contrasts")
## attr(,"contrasts")$region
## [1] "contr.treatment"

summary(lm(mass ~ region - 1))

##
## Call:
## lm(formula = mass ~ region - 1)
##
## Residuals:
##    1    2    3    4    5    6
## -0.5  1.5 -1.5  0.5 -1.0  1.0
##
```

```
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## region1    6.500      0.661    9.83  0.00060 ***
## region2   10.000      0.935   10.69  0.00043 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.32 on 4 degrees of freedom
## Multiple R-squared:  0.981,Adjusted R-squared:  0.972
## F-statistic:  105 on 2 and 4 DF,  p-value: 0.000347

# 6.5 + 3.5 = 10.0
```

简单线性回归：$\text{mass}_i = + * svl_i + {}_i; {}_i \; Normal(0, {}^2) mass_i \; Normal(+ * svl_i, {}^2)$

```
lm(mass ~ svl)

##
## Call:
## lm(formula = mass ~ svl)
##
## Coefficients:
## (Intercept)          svl
##       -5.56         0.28

model.matrix(~svl)

##   (Intercept) svl
## 1           1  40
## 2           1  45
## 3           1  39
## 4           1  50
## 5           1  52
## 6           1  57
## attr(,"assign")
## [1] 0 1
```

```
#
lm(mass ~ svl - 1)

##
## Call:
## lm(formula = mass ~ svl - 1)
##
## Coefficients:
##   svl
## 0.165

model.matrix(~svl - 1)

##   svl
## 1  40
## 2  45
## 3  39
## 4  50
## 5  52
## 6  57
## attr(,"assign")
## [1] 1
```

单变量方差分解：

$.j_i$ j pop i $mass_i = + .j_i * pop_i + {}_i ; {}_i \ Normal(0, {}^2) mass_i \ Normal(+ .j_i * pop_i, {}^2)$

Each parameterization is better suited to a different aim: the effects model is better for testing for differences and the means model is better for presentation.

```
# effects model
lm(mass ~ pop)

##
## Call:
## lm(formula = mass ~ pop)
```

```
##
## Coefficients:
## (Intercept)         pop2         pop3
##           7           -1            3

model.matrix(~pop)

##   (Intercept) pop2 pop3
## 1           1    0    0
## 2           1    0    0
## 3           1    1    0
## 4           1    1    0
## 5           1    0    1
## 6           1    0    1
## attr(,"assign")
## [1] 0 1 1
## attr(,"contrasts")
## attr(,"contrasts")$pop
## [1] "contr.treatment"

# means model
lm(mass ~ pop - 1)

##
## Call:
## lm(formula = mass ~ pop - 1)
##
## Coefficients:
## pop1  pop2  pop3
##    7     6    10

model.matrix(~pop - 1)

##   pop1 pop2 pop3
## 1    1    0    0
## 2    1    0    0
## 3    0    1    0
## 4    0    1    0
## 5    0    0    1
## 6    0    0    1
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$pop
## [1] "contr.treatment"
```

双变量方差分解：two-way analysis of variance

model 1：$\text{mass}_i = +{}_j.i * region_i + {}_k.i * hab_i + {}_i; {}_i\ Normal(0, {}^2)$

model 2：$\text{mass}_i = +{}_j.i * region_i + {}_k.i * hab_i + {}_j.k.i * region_i * hab_i + {}_i; {}_i\ Normal(0, {}^2)$

model 3：$\text{mass}_i = {}_j.k.i * region_i * hab_i + {}_i; {}_i\ Normal(0, {}^2)$

```
lm(mass ~ region + hab)

##
## Call:
## lm(formula = mass ~ region + hab)
##
## Coefficients:
## (Intercept)      region2         hab2          hab3
##        6.50         3.50         0.25         -0.25

model.matrix(~region + hab)  # model 1

##   (Intercept) region2 hab2 hab3
## 1           1       0    0    0
## 2           1       0    1    0
## 3           1       0    0    1
## 4           1       0    0    0
## 5           1       1    1    0
## 6           1       1    0    1
## attr(,"assign")
## [1] 0 1 2 2
## attr(,"contrasts")
## attr(,"contrasts")$region
## [1] "contr.treatment"
##
## attr(,"contrasts")$hab
## [1] "contr.treatment"
```

```
#
lm(mass ~ region * hab)
```

```
##
## Call:
## lm(formula = mass ~ region * hab)
##
## Coefficients:
##  (Intercept)      region2         hab2         hab3  region2:hab2
##          6.5          6.0          1.5         -1.5          -5.0
## region2:hab3
##           NA
```

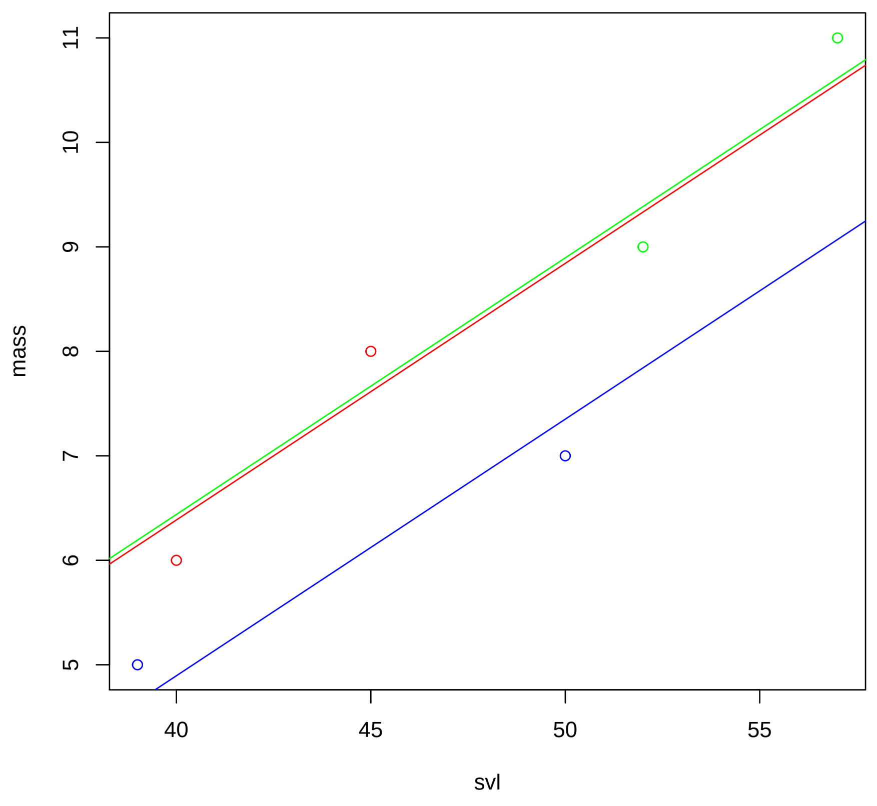
```
model.matrix(~region * hab)  # model 2
```

```
##   (Intercept) region2 hab2 hab3 region2:hab2 region2:hab3
## 1           1       0    0    0            0            0
## 2           1       0    1    0            0            0
## 3           1       0    0    1            0            0
## 4           1       0    0    0            0            0
## 5           1       1    1    0            1            0
## 6           1       1    0    1            0            1
## attr(,"assign")
## [1] 0 1 2 2 3 3
## attr(,"contrasts")
## attr(,"contrasts")$region
## [1] "contr.treatment"
##
## attr(,"contrasts")$hab
## [1] "contr.treatment"
```

```
#
lm(mass ~ region * hab - 1 - region - hab)

##
## Call:
## lm(formula = mass ~ region * hab - 1 - region - hab)
##
## Coefficients:
## region1:hab1  region2:hab1  region1:hab2  region2:hab2  region1:hab3
##          6.5            NA           8.0           9.0           5.0
## region2:hab3
##         11.0

model.matrix(~region * hab - 1 - region - hab)  # model 3

##   region1:hab1 region2:hab1 region1:hab2 region2:hab2 region1:hab3
## 1            1            0            0            0            0
## 2            0            0            1            0            0
## 3            0            0            0            0            1
## 4            1            0            0            0            0
## 5            0            0            0            1            0
## 6            0            0            0            0            0
##   region2:hab3
## 1            0
## 2            0
## 3            0
## 4            0
## 5            0
## 6            1
## attr(,"assign")
## [1] 1 1 1 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$region
## [1] "contr.treatment"
##
## attr(,"contrasts")$hab
## [1] "contr.treatment"
```

协方差分析：analysis of covariance

```
#
fm <- lm(mass ~ pop + svl)  # Refit model
fm

##
## Call:
## lm(formula = mass ~ pop + svl)
##
## Coefficients:
## (Intercept)         pop2         pop3          svl
##     -3.4386      -1.4912       0.0526       0.2456

model.matrix(~pop + svl)

##   (Intercept) pop2 pop3 svl
## 1           1    0    0  40
## 2           1    0    0  45
## 3           1    1    0  39
## 4           1    1    0  50
## 5           1    0    1  52
## 6           1    0    1  57
## attr(,"assign")
## [1] 0 1 1 2
## attr(,"contrasts")
## attr(,"contrasts")$pop
## [1] "contr.treatment"

plot(svl, mass, col = c(rep("red", 2), rep("blue", 2), rep("green", 2)))
abline(fm$coef[1], fm$coef[4], col = "red")
abline(fm$coef[1] + fm$coef[2], fm$coef[4], col = "blue")
abline(fm$coef[1] + fm$coef[3], fm$coef[4], col = "green")
```

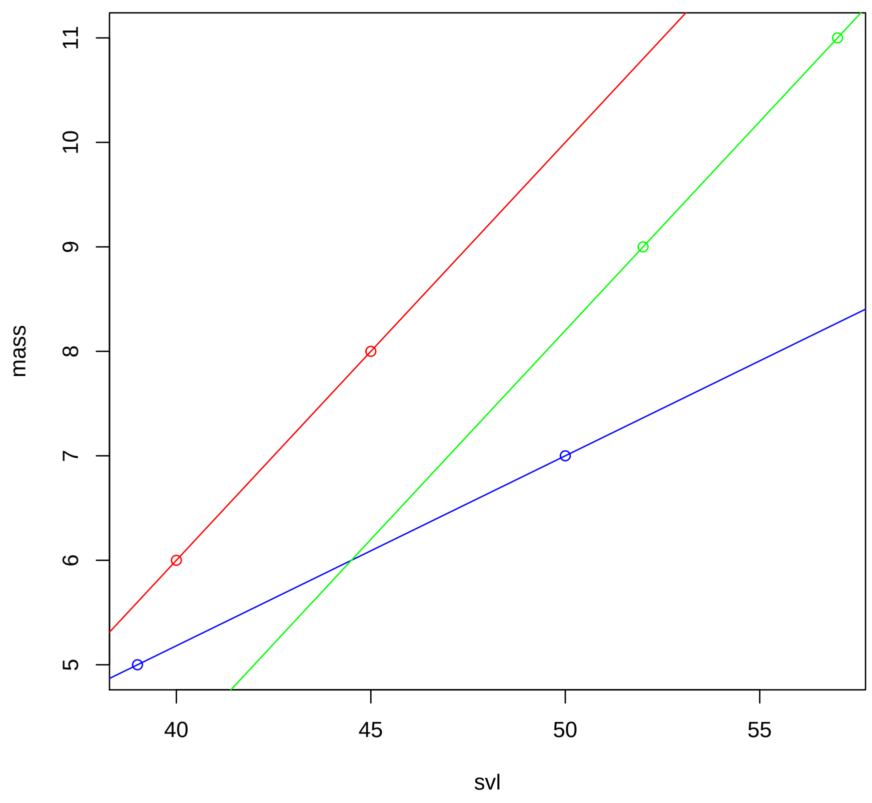


```
#
fm <- lm(mass ~ pop * svl)  # Refit model
fm

##
## Call:
## lm(formula = mass ~ pop * svl)
##
## Coefficients:
## (Intercept)         pop2         pop3          svl     pop2:svl
##   -1.00e+01     7.91e+00    -1.80e+00     4.00e-01    -2.18e-01
##    pop3:svl
##   -1.60e-15
```

```
model.matrix(~pop * svl)
```

```
##   (Intercept) pop2 pop3 svl pop2:svl pop3:svl
## 1           1    0    0  40        0        0
## 2           1    0    0  45        0        0
## 3           1    1    0  39       39        0
## 4           1    1    0  50       50        0
## 5           1    0    1  52        0       52
## 6           1    0    1  57        0       57
## attr(,"assign")
## [1] 0 1 1 2 3 3
## attr(,"contrasts")
## attr(,"contrasts")$pop
## [1] "contr.treatment"
```

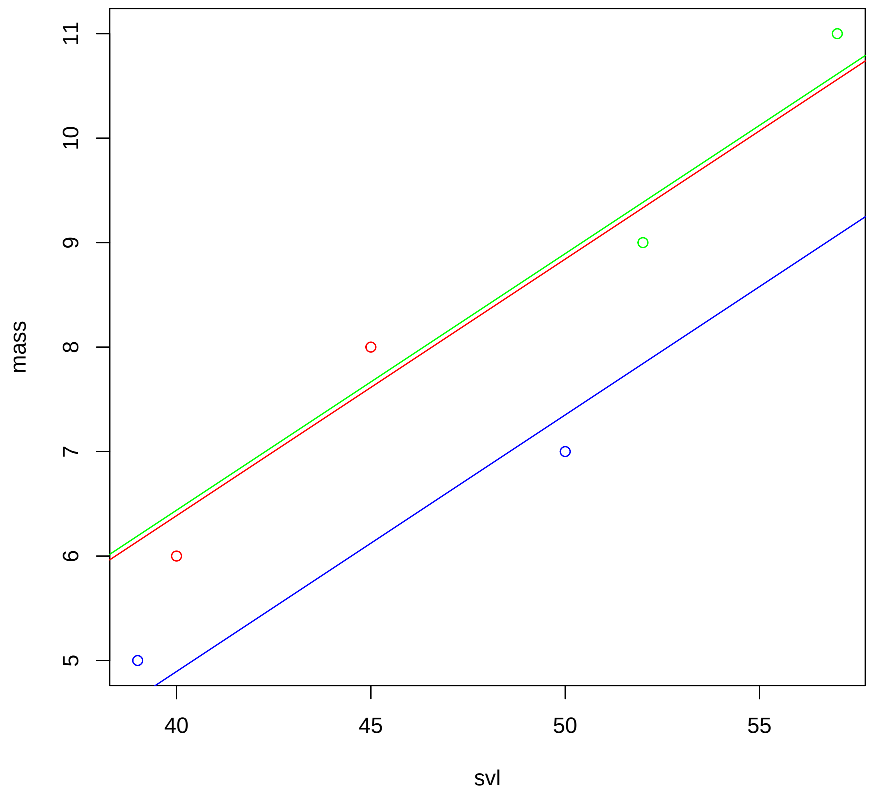
```
plot(svl, mass, col = c(rep("red", 2), rep("blue", 2), rep("green", 2)))
abline(fm$coef[1], fm$coef[4], col = "red")
abline(fm$coef[1] + fm$coef[2], fm$coef[4] + fm$coef[5], col = "blue")
abline(fm$coef[1] + fm$coef[3], fm$coef[4] + fm$coef[6], col = "green")
```



```
#
fm <- lm(mass ~ pop + svl - 1)
fm

##
## Call:
## lm(formula = mass ~ pop + svl - 1)
##
## Coefficients:
##   pop1    pop2    pop3     svl
## -3.439  -4.930  -3.386   0.246

model.matrix(~pop + svl - 1)
```

```
##   pop1 pop2 pop3 svl
## 1    1    0    0  40
## 2    1    0    0  45
## 3    0    1    0  39
## 4    0    1    0  50
## 5    0    0    1  52
## 6    0    0    1  57
## attr(,"assign")
## [1] 1 1 1 2
## attr(,"contrasts")
## attr(,"contrasts")$pop
## [1] "contr.treatment"
```

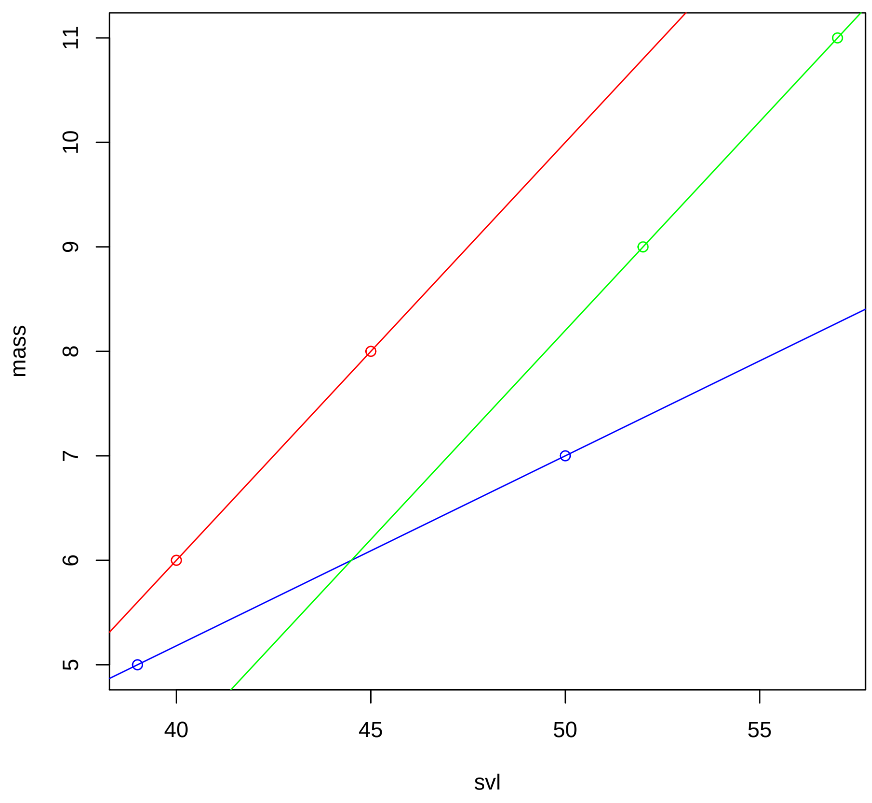
```
plot(svl, mass, col = c(rep("red", 2), rep("blue", 2), rep("green", 2)))
abline(fm$coef[1], fm$coef[4], col = "red")
abline(fm$coef[2], fm$coef[4], col = "blue")
abline(fm$coef[3], fm$coef[4], col = "green")
```



```
#
fm <- lm(mass ~ pop * svl - 1 - svl)
fm

##
## Call:
## lm(formula = mass ~ pop * svl - 1 - svl)
##
## Coefficients:
##     pop1      pop2      pop3  pop1:svl  pop2:svl  pop3:svl
##  -10.000    -2.091   -11.800     0.400     0.182     0.400

model.matrix(~pop * svl - 1 - svl)
```

```
##   pop1 pop2 pop3 pop1:svl pop2:svl pop3:svl
## 1    1    0    0       40        0        0
## 2    1    0    0       45        0        0
## 3    0    1    0        0       39        0
## 4    0    1    0        0       50        0
## 5    0    0    1        0        0       52
## 6    0    0    1        0        0       57
## attr(,"assign")
## [1] 1 1 1 2 2 2
## attr(,"contrasts")
## attr(,"contrasts")$pop
## [1] "contr.treatment"
```

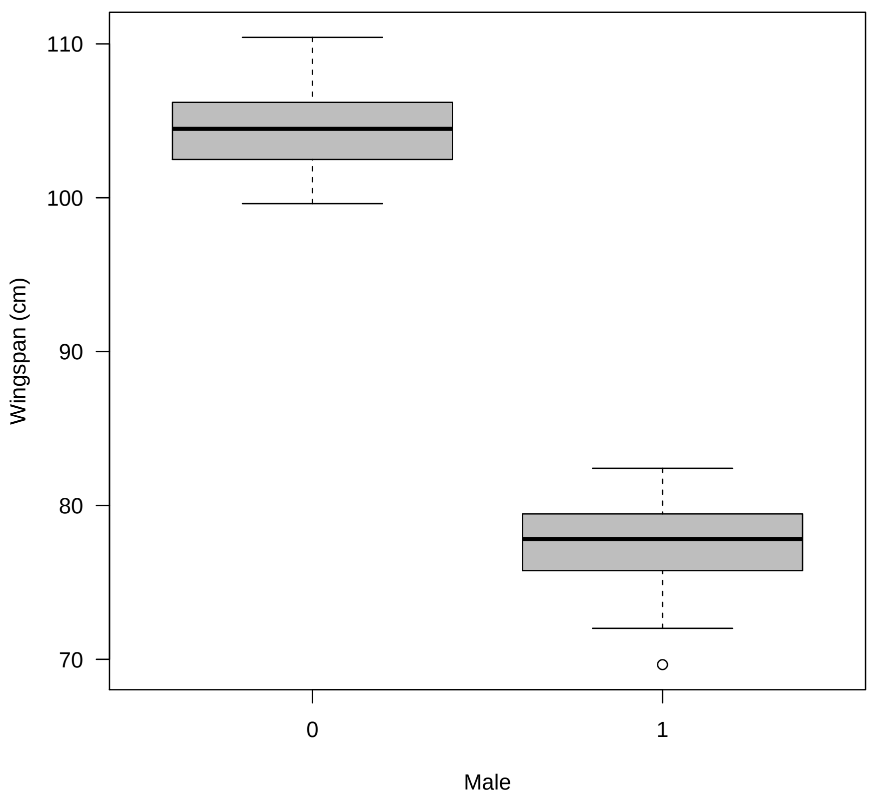
```
plot(svl, mass, col = c(rep("red", 2), rep("blue", 2), rep("green", 2)))
abline(fm$coef[1], fm$coef[4], col = "red")
abline(fm$coef[2], fm$coef[5], col = "blue")
abline(fm$coef[3], fm$coef[6], col = "green")
```



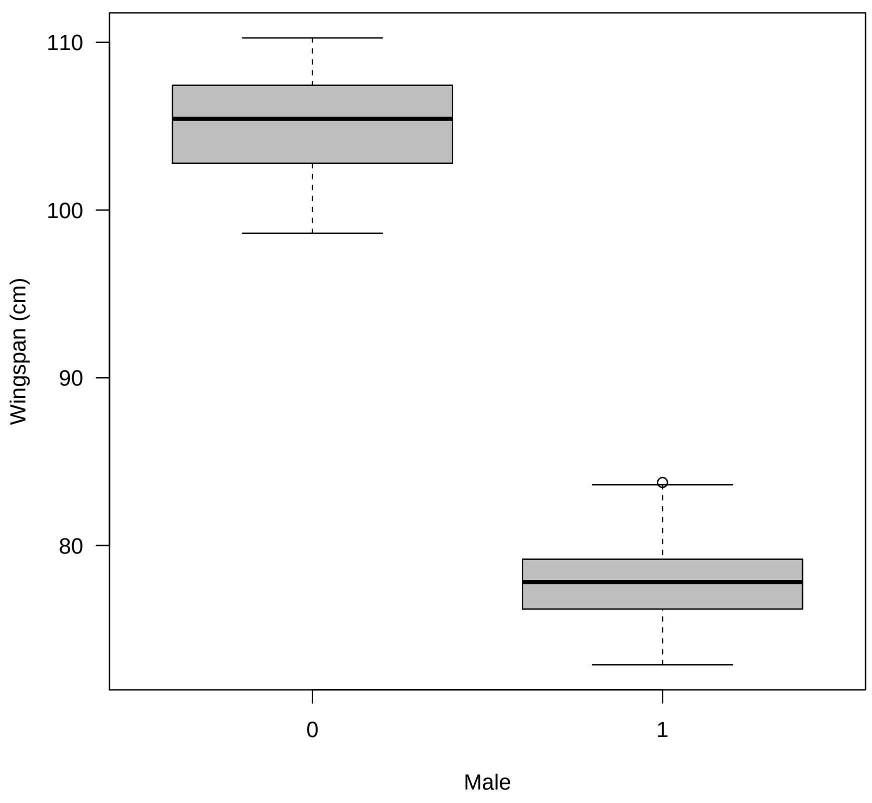
第三章真正的 T 检验

```
# data ~~~~~~~~~~~~~~~~~~
n1 <- 60  # Number of females
n2 <- 40  # Number of males
mu1 <- 105  # Population mean of females
mu2 <- 77.5  # Population mean of males
sigma <- 2.75  # Average population SD of both
n <- n1 + n2  # Total sample size
#
y1 <- rnorm(n1, mu1, sigma)  # Data for females
y2 <- rnorm(n2, mu2, sigma)  # Date for males
y <- c(y1, y2)  # Aggregate both data sets
```

```
x <- rep(c(0, 1), c(n1, n2))  # Indicator for male
boxplot(y ~ x, col = "grey", xlab = "Male", ylab = "Wingspan (cm)", las = 1)
```



```
# data
n <- n1 + n2  # Total sample size
alpha <- mu1  # Mean for females serves as the intercept
beta <- mu2 - mu1  # Beta is the difference male female
E.y <- alpha + beta * x  # Expectation
y.obs <- rnorm(n = n, mean = E.y, sd = sigma)  # Add random variation
boxplot(y.obs ~ x, col = "grey", xlab = "Male", ylab = "Wingspan (cm)", las = 1)
```



```
#
fit1 <- lm(y ~ x)  # Analysis of first data set
fit2 <- lm(y.obs ~ x)  # Analysis of second data set
summary(fit1)

##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.839 -1.813  0.269  1.845  6.035
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  104.387      0.348   300.0   <2e-16 ***
## x            -26.892      0.550   -48.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.69 on 98 degrees of freedom
## Multiple R-squared:  0.961,Adjusted R-squared:  0.96
## F-statistic: 2.39e+03 on 1 and 98 DF,  p-value: <2e-16
```

```
summary(fit2)
```

```
##
## Call:
## lm(formula = y.obs ~ x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.473 -2.001  0.137  1.973  5.977
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  105.087      0.358   293.6   <2e-16 ***
## x            -27.307      0.566   -48.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.77 on 98 degrees of freedom
## Multiple R-squared:  0.96,Adjusted R-squared:  0.959
## F-statistic: 2.33e+03 on 1 and 98 DF,  p-value: <2e-16
```

```
#
anova(fit1)
```

```
## Analysis of Variance Table
##
```

```
## Response: y
##           Df Sum Sq Mean Sq F value Pr(>F)
## x          1  17357   17357    2390 <2e-16 ***
## Residuals 98    712       7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(fit2)
```

```
## Analysis of Variance Table
##
## Response: y.obs
##           Df Sum Sq Mean Sq F value Pr(>F)
## x          1  17896   17896    2329 <2e-16 ***
## Residuals 98    753       8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#
model.matrix(fit1)
```

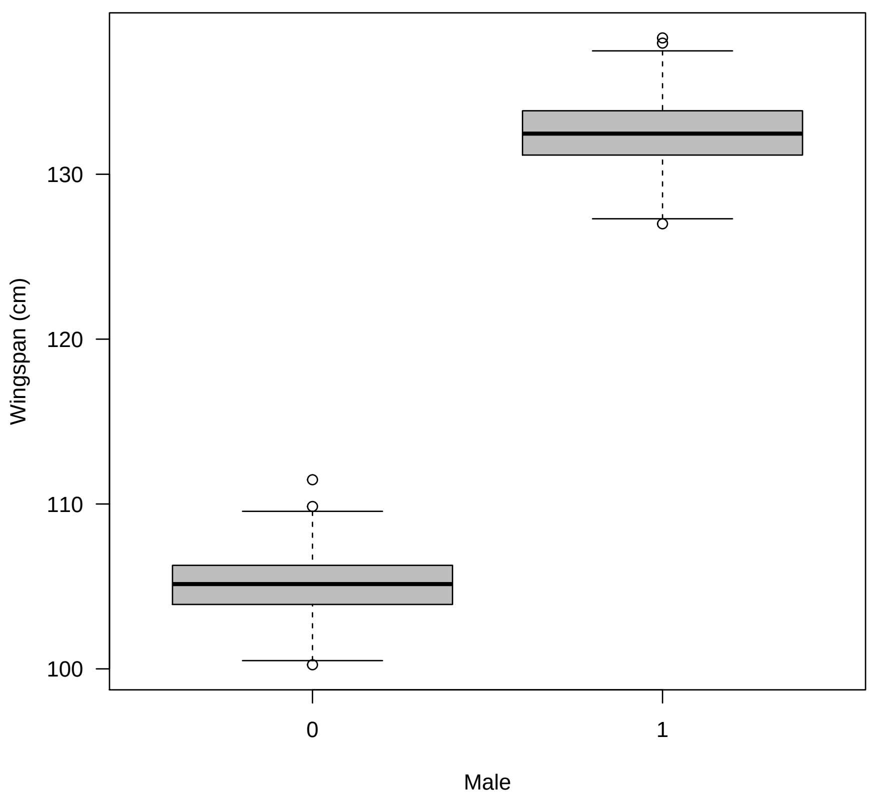
```
##     (Intercept) x
## 1             1 0
## 2             1 0
## 3             1 0
## 4             1 0
## 5             1 0
## 6             1 0
## 7             1 0
## 8             1 0
## 9             1 0
## 10            1 0
## 11            1 0
## 12            1 0
## 13            1 0
## 14            1 0
## 15            1 0
## 16                1 0
## 17                1 0
## 18                1 0
## 19                1 0
## 20                1 0
## 21                1 0
## 22                1 0
## 23                1 0
## 24                1 0
## 25                1 0
## 26                1 0
## 27                1 0
## 28                1 0
## 29                1 0
## 30                1 0
## 31                1 0
## 32                1 0
## 33                1 0
## 34                1 0
## 35                1 0
## 36                1 0
## 37                1 0
## 38                1 0
## 39                1 0
## 40                1 0
## 41                1 0
## 42                1 0
## 43                1 0
## 44                1 0
## 45                1 0
## 46                1 0
## 47                1 0
## 48                1 0
## 49                1 0
## 50                1 0
## 51            1 0
## 52            1 0
## 53            1 0
## 54            1 0
## 55            1 0
## 56            1 0
## 57            1 0
## 58            1 0
## 59            1 0
## 60            1 0
## 61            1 1
## 62            1 1
## 63            1 1
## 64            1 1
## 65            1 1
## 66            1 1
## 67            1 1
## 68            1 1
## 69            1 1
## 70            1 1
## 71            1 1
## 72            1 1
## 73            1 1
## 74            1 1
## 75            1 1
## 76            1 1
## 77            1 1
## 78            1 1
## 79            1 1
## 80            1 1
## 81            1 1
## 82            1 1
## 83            1 1
## 84            1 1
## 85            1 1
## 86              1 1
## 87              1 1
## 88              1 1
## 89              1 1
## 90              1 1
## 91              1 1
## 92              1 1
## 93              1 1
## 94              1 1
## 95              1 1
## 96              1 1
## 97              1 1
## 98              1 1
## 99              1 1
## 100             1 1
## attr(,"assign")
## [1] 0 1
```

```
model.matrix(fit2)
```

```
##     (Intercept) x
## 1             1 0
## 2             1 0
## 3             1 0
## 4             1 0
## 5             1 0
## 6             1 0
## 7             1 0
## 8             1 0
## 9             1 0
## 10            1 0
## 11            1 0
## 12            1 0
## 13            1 0
## 14            1 0
## 15            1 0
## 16              1 0
## 17              1 0
## 18              1 0
## 19              1 0
## 20              1 0
## 21              1 0
## 22              1 0
## 23              1 0
## 24              1 0
## 25              1 0
## 26              1 0
## 27              1 0
## 28              1 0
## 29              1 0
## 30              1 0
## 31              1 0
## 32              1 0
## 33              1 0
## 34              1 0
## 35              1 0
## 36              1 0
## 37              1 0
## 38              1 0
## 39              1 0
## 40              1 0
## 41              1 0
## 42              1 0
## 43              1 0
## 44              1 0
## 45              1 0
## 46              1 0
## 47              1 0
## 48              1 0
## 49              1 0
## 50              1 0
## 51            1 0
## 52            1 0
## 53            1 0
## 54            1 0
## 55            1 0
## 56            1 0
## 57            1 0
## 58            1 0
## 59            1 0
## 60            1 0
## 61            1 1
## 62            1 1
## 63            1 1
## 64            1 1
## 65            1 1
## 66            1 1
## 67            1 1
## 68            1 1
## 69            1 1
## 70            1 1
## 71            1 1
## 72            1 1
## 73            1 1
## 74            1 1
## 75            1 1
## 76            1 1
## 77            1 1
## 78            1 1
## 79            1 1
## 80            1 1
## 81            1 1
## 82            1 1
## 83            1 1
## 84            1 1
## 85            1 1
## 86              1 1
## 87              1 1
## 88              1 1
## 89              1 1
## 90              1 1
## 91              1 1
## 92              1 1
## 93              1 1
## 94              1 1
## 95              1 1
## 96              1 1
## 97              1 1
## 98              1 1
## 99              1 1
## 100             1 1
## attr(,"assign")
## [1] 0 1
```

现在来看一下怎么使用 bayesian 方法实现呢？

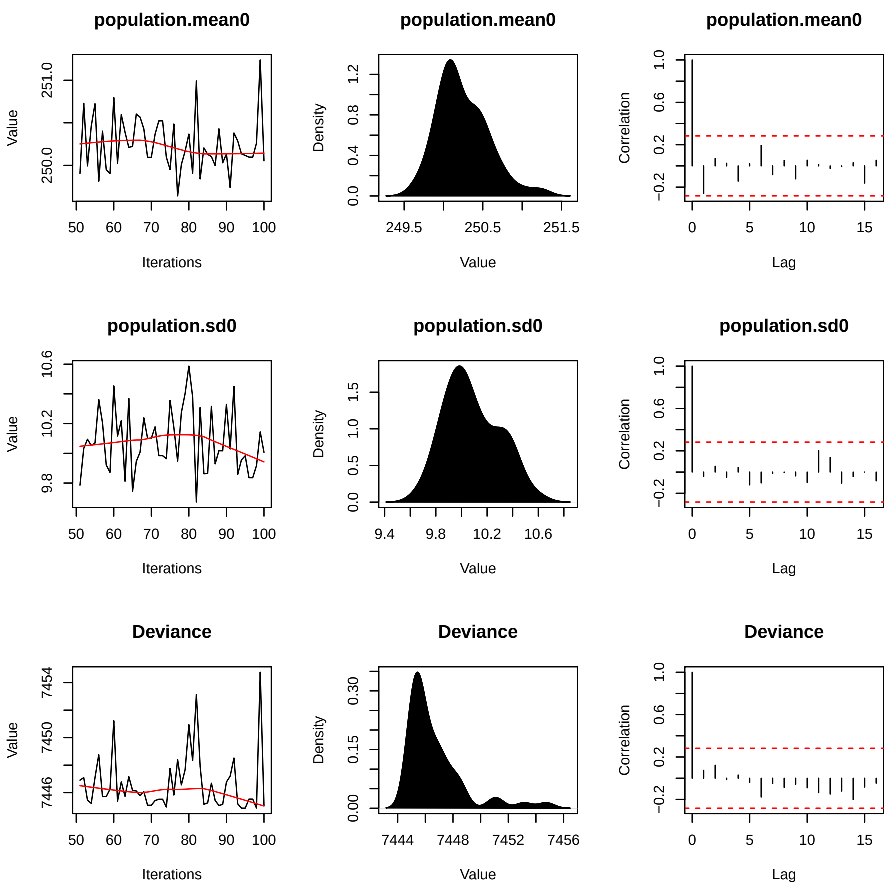
```
## 2.2 Data ~~~~~~~~~~~~~~~~~~~
require(LaplacesDemon)
##################################################### data ~~~~~~~~~~~~~~~~~~
n1 <- 600  # Number of females
n2 <- 400  # Number of males
mu1 <- 105  # Population mean of females
mu2 <- 77.5  # Population mean of males
sigma <- 1.75  # Average population SD of both
#
n <- n1 + n2  # Total sample size
alpha <- mu1  # Mean for females serves as the intercept
beta <- mu1 - mu2  # Beta is the difference male female
x <- rep(c(0, 1), c(n1, n2))
E.y <- alpha + beta * x  # Expectation
y.obs <- rnorm(n = n, mean = E.y, sd = sigma)  # Add random variation
boxplot(y.obs ~ x, col = "grey", xlab = "Male", ylab = "Wingspan (cm)", las = 1)
```

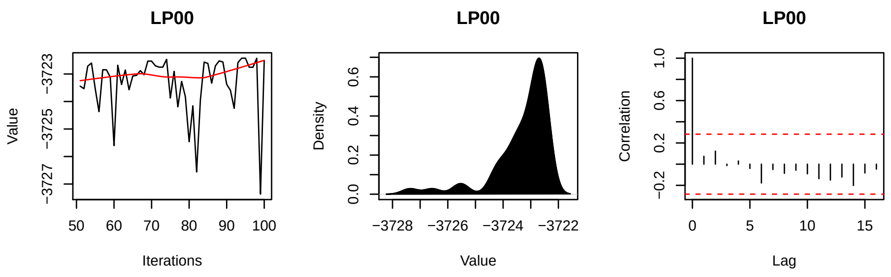


```
######################################################
parm.names <- as.parm.names(Initial.Values)

## Error: replacement has length zero

mon.names <- c("LP00")
Data <- MyData <- list(N = n, mon.names = mon.names, parm.names = parm.names,
    x = x, y = y.obs)
## 2.3. Model ~~~~~~~~~~~~~~~~~
Model <- function(parm, Data) {
    ### Parameters
    alpha <- parm[1]
    beta <- parm[2]
```

```
    sigma <- parm[3]
    ### Log(Prior Densities)
   alpha.prior <- dnormv(x = alpha, mean = 0, var = 1000, log = TRUE)
    beta.prior <- dnorm(x = beta, mean = 0, sd = sigma, log = TRUE)
    sigma.prior <- dhalfcauchy(x = sigma, scale = 25, log = TRUE)
    ### Log-Likelihood
    mu <- alpha + beta * Data$x
    LL <- sum(dnorm(x = Data$y, mean = mu, sd = sigma, log = TRUE))
    ### Log-Posterior
    LP <- LL + alpha.prior + beta.prior + sigma.prior
   Modelout <- list(LP = LP, Dev = -2 * LL, Monitor = c(LP), yhat = rnorm(length(mu),
        mu, sigma), parm = parm)
    return(Modelout)
}
## 2.4. Initial Values~~~~~~~~~~~~~~
parm <- Initial.Values <- c(alpha_11 = 100, beta_11 = 20, sigma_11 = 1)
## 2.5 MCMC ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ MCMC settings
ni <- 10000  # Number of draws from posterior (for each chain)
st <- 4000  # Steps when status message should be given
nt <- 50  # Thinning rate #  Abate autocorrelation
# Run LaplacesDemon
out <- LaplacesDemon(Model, Data = MyData, Initial.Values, Iterations = ni,
    Status = st, Thinning = nt)

##
## Laplace's Demon was called on Wed Oct 30 18:04:59 2013
##
## Performing initial checks...
## WARNING: The length of Initial Values differed from Data$parm.names.

## Warning: NAs produced

## Generating initial values due to a non-finite posterior.

## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
```

```
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning:  NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced
## Warning: NAs produced




## Warning: NAs produced
## Error: The deviance is a missing value!

# Have a look at some summary statistics
out



                                                                .Values,
## population.mean0   population.sd0
##            2.835            2.835
##
## Covariance (Diagonal) History: (NOT SHOWN HERE)
## Deviance Information Criterion (DIC):
##          All Stationary
## Dbar    7956     7449.2
## pD   1433069      278.3
## DIC  1441024     7727.5
##
## Delayed Rejection (DR): 0
## Initial Values:
## [1] 42.79 51.29
##
## Iterations: 5000
## Log(Marginal Likelihood): NA
## Minutes of run-time: 0.02
## Model: (NOT SHOWN HERE)
## Monitor: (NOT SHOWN HERE)
## Parameters (Number of): 2
## Periodicity: 5001
## Posterior1: (NOT SHOWN HERE)
## Posterior2: (NOT SHOWN HERE)
## Recommended Burn-In of Thinned Samples: 10
## Recommended Burn-In of Un-thinned Samples: 500
## Recommended Thinning: 400
## Status is displayed every 1000 iterations
## Summary1: (SHOWN BELOW)
## Summary2: (SHOWN BELOW)
## Thinned Samples: 100
## Thinning: 50
##
##
## Summary of All Samples
##                 Mean     SD   MCSE   ESS      LB  Median     UB
## population.mean0  239.16  39.27  14.56  7.461   86.407   250.2   250.8
## population.sd0     18.86   28.52  11.47 12.477    9.633   10.1   118.7
## Deviance       7955.58 1692.97 666.52 11.773  7444.874  7446.4 13105.7
## LP           -3977.78  846.49 333.26 11.773 -6552.836 -3723.2 -3722.4
##
##
## Summary of Stationary Samples
##                      Mean      SD      MCSE ESS        LB   Median
## population.mean0   250.23  0.4124   0.04023  90   249.668   250.22
## population.sd0      10.11  0.4844   0.05074  90     9.624    10.06
## Deviance          7449.19 23.5941   2.45715  90  7444.873  7446.20
## LP               -3724.59 11.7971 333.26228  90 -3726.344 -3723.09
##                         UB
## population.mean0    250.78
## population.sd0       10.58
## Deviance           7452.71
## LP                -3722.43

# alpha= 105, beta= 27.5, sigma= 1.75 Plotting output
plot(out, BurnIn = 50, Data, PDF = F)
```


population.mean0
population.mean0
population.mean0
Value
Iterations
Density
Value
Correlation
Lag
population.sd0
population.sd0
population.sd0
Value
Iterations
Density
Value
Correlation
Lag
Deviance
Deviance
Deviance
Value
Iterations
Density
Value
Correlation
Lag
250.0
251.0
249.5
250.5
251.5
9.8
10.2
10.6
9.4
10.6
7446
7450
7454
7444
7448
7452
7456
0.00
0.15
0.30
−0.2
0.2
0.6
1.0
0
5
10
15
50
60
70
80
90
100



LP00
Value
Iterations
−3723
−3725
−3727
50
60
70
80
90
100
LP00
Density
Value
0.0
0.2
0.4
0.6
−3728
−3726
−3724
−3722
LP00
Correlation
Lag
1.0
0.6
0.2
−0.2
0
5
10
15