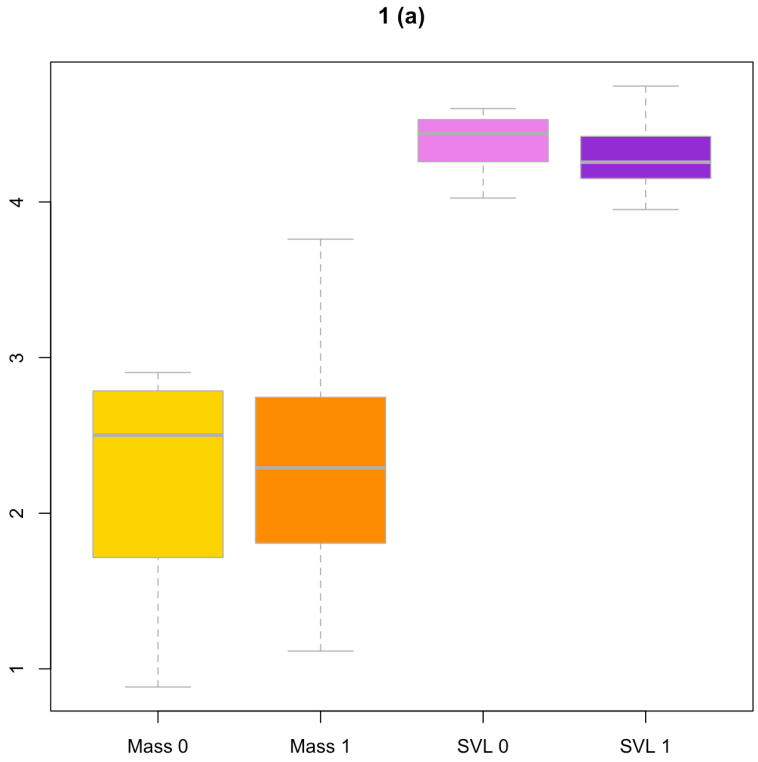
Multivariate Analysis

Midterm Exam

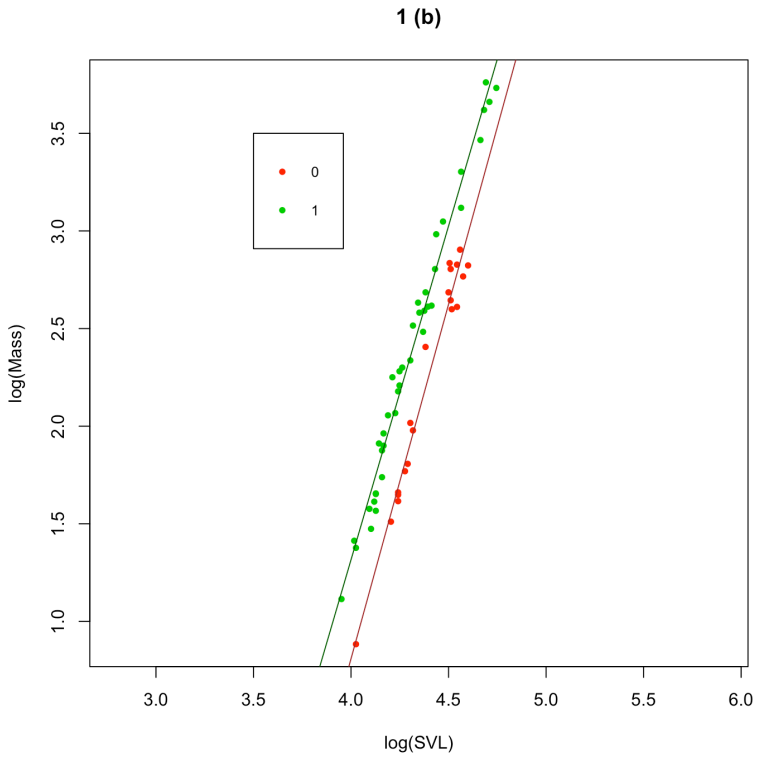
M052040003 鍾冠毅

1.

(a)



(b)



(c)

```
> df1.means
      Mass     SVL
0 10.87460 81.9750
1 13.83977 75.9125
> cov(g0)
         Mass       SVL
Mass 28.54309  65.55475
SVL  65.55475 160.93355
> cov(g1)
         Mass      SVL
Mass 121.3393 178.2155
SVL  178.2155 274.8704
> Spool
          Mass      SVL
Mass  90.94051 141.3094
SVL  141.30938 237.5462
```

(d)

```
> shapiro.test(y1)

        Shapiro-Wilk normality test

data:  y1
W = 0.97589, p-value = 0.2798

> shapiro.test(y2)

        Shapiro-Wilk normality test

data:  y2
W = 0.97477, p-value = 0.2481

> t.test(y1, y2, var.equal = T, paired = F)

        Two Sample t-test

data:  y1 and y2
t = -22.244, df = 118, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.190538 -1.832402
sample estimates:
mean of x mean of y
  2.32540   4.33687
```

(e)

No, by the variance test, we notice that p-value is to small so that the null hypothesis is rejected.

```
> var.test(y1, y2)

        F test to compare two variances

data:  y1 and y2
F = 11.853, num df = 59, denom df = 59, p-value < 2.2e-16
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
  7.080124 19.843589
sample estimates:
ratio of variances
          11.85306
```

(f)

Figure 1(a) shows that the means of responses Mass and SVL are different. Also, the factor genera affect the response. In figure 1(b), we can notice that the factor genera distribute the observation into two parallel lines. If we test the null hypothesis in (d) with the assumption of different, the p-value < 2.2e-16 is so small that we reject the equal-meaned hypothesis.

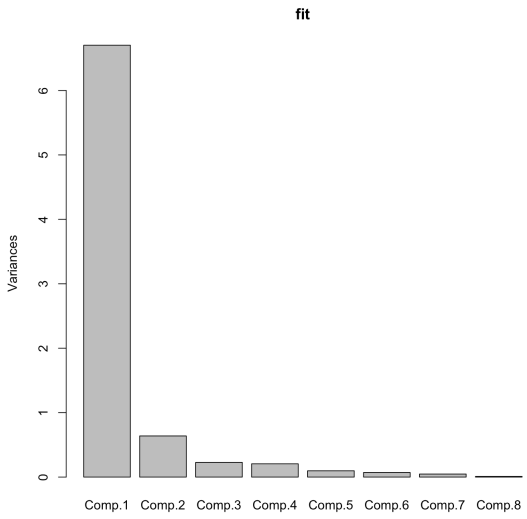
2.

(a)

```
> colMeans(df2[, -1])
     X100m.s      X200m.s      X400m.s    X800m.min   X1500m.min   X3000m.min   X10000.min Marathon.min
   10.216852    20.541481    45.829074     1.768148     3.653333    13.617593    28.535185   133.478519
> cov(df2[, -1])
                X100m.s    X200m.s    X400m.s   X800m.min  X1500m.min X3000m.min  X10000.min Marathon.min
X100m.s      0.048972921 0.11104437 0.25602156 0.008263871 0.025720126 0.12457530  0.26561286    1.3401386
X200m.s      0.111044375 0.30090342 0.66681838 0.022929210 0.066193082 0.31773382  0.68893557    3.5410381
X400m.s      0.256021558 0.66681838 2.06995573 0.057937876 0.168472956 0.85348641  1.84994074    9.1788571
X800m.min    0.008263871 0.02292921 0.05793788 0.002751223 0.007130818 0.03434829  0.07425695    0.3789048
X1500m.min   0.025720126 0.06619308 0.16847296 0.007130818 0.023033962 0.10583270  0.22970126    1.1925635
X3000m.min   0.124575297 0.31773382 0.85348641 0.034348288 0.105832704 0.57887523  1.26253347    6.4304888
X10000.min   0.265612858 0.68893557 1.84994074 0.074256953 0.229701258 1.26253347  2.81956883   14.3425380
Marathon.min 1.340138644 3.54103809 9.17885709 0.378904752 1.192563522 6.43048882 14.34253802   80.1353563
> cor(df2[, -1])
              X100m.s   X200m.s   X400m.s X800m.min X1500m.min X3000m.min X10000.min Marathon.min
X100m.s      1.0000000 0.9147554 0.8041147 0.7119388  0.7657919  0.7398803  0.7147921    0.6764873
X200m.s      0.9147554 1.0000000 0.8449159 0.7969162  0.7950871  0.7613028  0.7479519    0.7211157
X400m.s      0.8041147 0.8449159 1.0000000 0.7677488  0.7715522  0.7796929  0.7657481    0.7126823
X800m.min    0.7119388 0.7969162 0.7677488 1.0000000  0.8957609  0.8606959  0.8431074    0.8069657
X1500m.min   0.7657919 0.7950871 0.7715522 0.8957609  1.0000000  0.9165224  0.9013380    0.8777788
X3000m.min   0.7398803 0.7613028 0.7796929 0.8606959  0.9165224  1.0000000  0.9882324    0.9441466
X10000.min   0.7147921 0.7479519 0.7657481 0.8431074  0.9013380  0.9882324  1.0000000    0.9541630
Marathon.min 0.6764873 0.7211157 0.7126823 0.8069657  0.8777788  0.9441466  0.9541630    1.0000000
```

(b) Two components would be enough.

```
> summary(fit) # print variance accounted for
Importance of components:
                          Comp.1     Comp.2     Comp.3     Comp.4     Comp.5      Comp.6      Comp.7      Comp.8
Standard deviation     2.5890713 0.79900570 0.47699528 0.45370605 0.31237388 0.265871985 0.216661141 0.098584288
Proportion of Variance 0.8379112 0.07980126 0.02844056 0.02573115 0.01219718 0.008835989 0.005867756 0.001214858
Cumulative Proportion  0.8379112 0.91771251 0.94615307 0.97188422 0.98408140 0.992917386 0.998785142 1.000000000
```



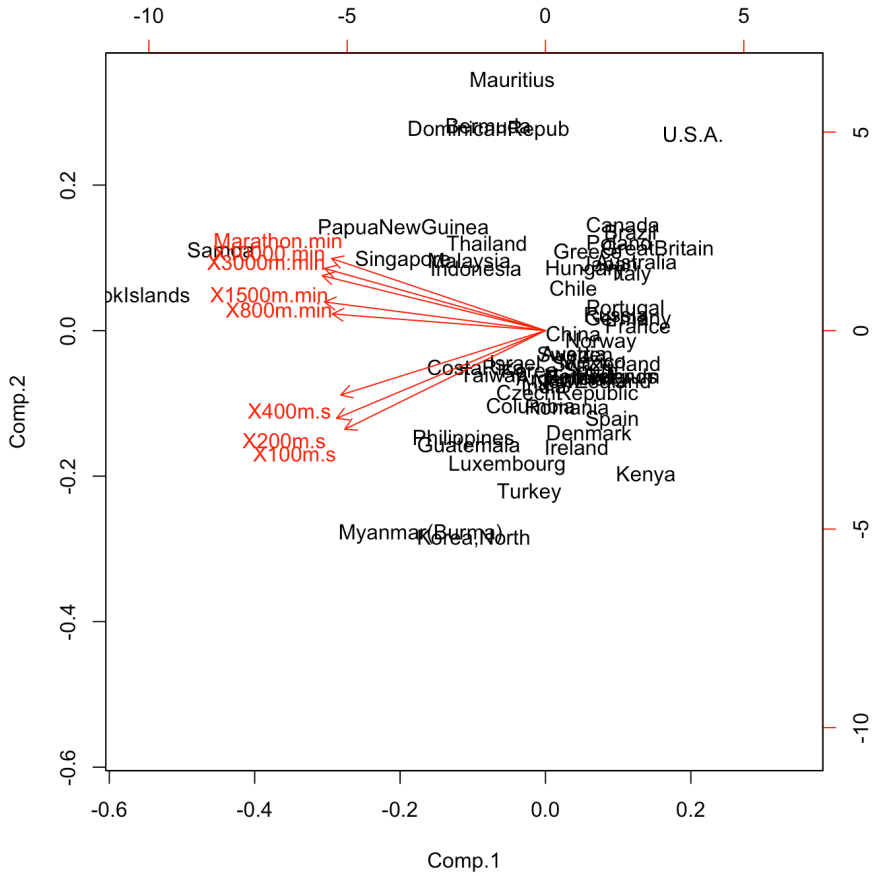
(c)

Check the figure besides. Considering the component 1, the U.S.A. performs the best and Cook Islands and N.Korea perform the worst. It correspond to our intuitive notion that the U.S.A. always perform the best in the Olympics.

```
> loadings(fit) # pc loadings

Loadings:
             Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
X100m.s      -0.332 -0.529  0.344  0.381 -0.300 -0.362  0.348
X200m.s      -0.346 -0.470         0.217  0.541  0.349 -0.440
X400m.s      -0.339 -0.345        -0.851 -0.133         0.114
X800m.min    -0.353        -0.783  0.134  0.227 -0.341  0.259
X1500m.min   -0.366  0.154 -0.244  0.233 -0.652  0.530 -0.147
X3000m.min   -0.370  0.295  0.183                -0.359 -0.328  0.706
X10000.min   -0.366  0.334  0.244                -0.273 -0.351 -0.697
Marathon.min -0.354  0.387  0.335         0.338  0.375  0.594

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.125  0.125  0.125  0.125  0.125  0.125  0.125  0.125
Cumulative Var  0.125  0.250  0.375  0.500  0.625  0.750  0.875  1.000
```



(d)

The loadings of component 1 are all negative, because of the observations are the records of the minutes or seconds the athletes perform. Thus, the less the better. It's different from the results of the "the-more-the-better"data.