投野ゼミ火□

Chapter 3　Describing Data Numerically and Graphically

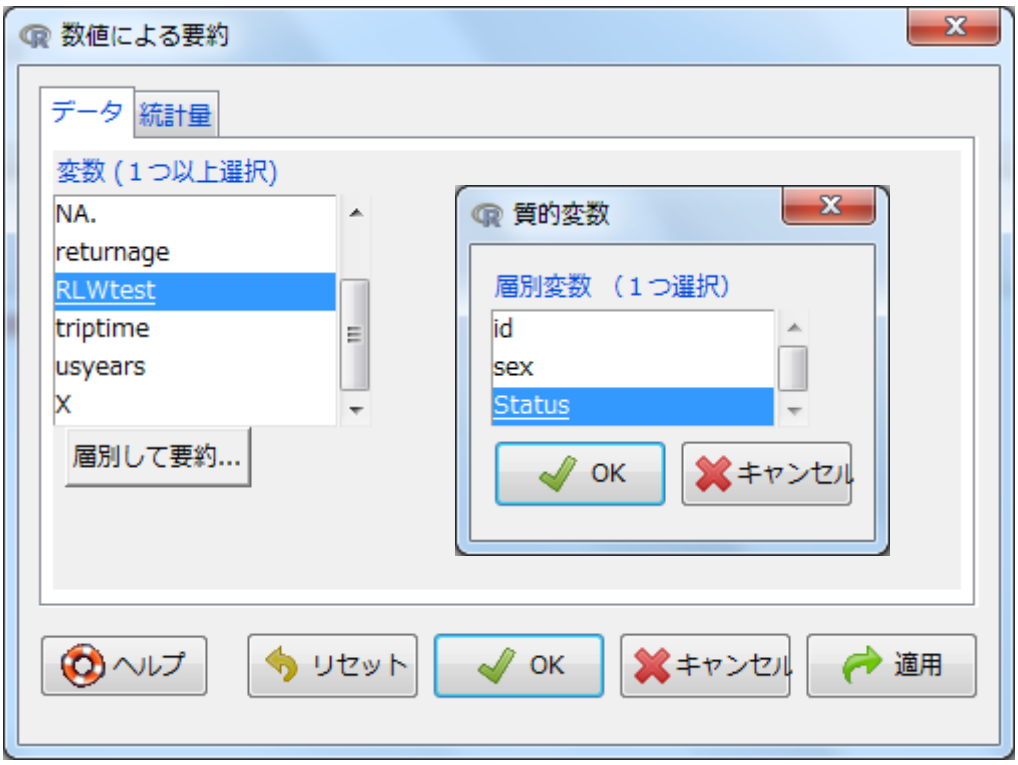
P44-59

黄鈴娥

2015/10/13

3.1 Obtaining Numerical Summaries　P44

①Data:　LarsonHall.Forgotten.sav
　Name:　forget
　R Cmd:　STATISTICS > SUMMARIES > NUMERICAL SUMMARIES
　R Code:　numSummary(forget[,"rlwtest"], groups= Status, statistics=c("mean", "sd", "quantiles"))



sum(!is.na(rlwtest)) #overall N for the variable

```
           mean       sd 0%   25% 50%   75% 100%  n
Non    67.73333 13.76054 47 56.5  67 79.00   91 15
Late   69.20000 15.37716 36 60.5  66 80.50   90 15
Early 80.85714 14.27670 47 71.5  88 90.75   95 14
```

②To get other types of statistical summaries using R Commander, choose STATISTICS > SUMMARIES > TABLE OF STATISTICS or using the tapply():

attach(forget)
tapply(rlwtest, list(Status=forget$Status), mean, na.rm=T)

```
     Non      Late     Early
67.73333 69.20000 80.85714
```

③To summarize the entire number of observations in a variable and remove any NAs for that variable:

```
sum(!is.na(rlwtest)) #overall N for the variable
```

[1] 44

```
sum(!is.na(usyears))
```

[1] 29

④To get a count of the number of participants in each group for a certain variable exclude NAs,

```
table(status[!is.na(rlwtest)]) #N count for each group
```

```
Non  Late Early
 15    15    14
```

```
table(writing$L1,writing$condition[!is.na(writing$score)])
```

Data: writing.csv

Name : writing

	correctAll	correctTarget	noCorrect
Arabic	59	76	85
Japanese	30	40	50
Russian	30	40	48
Spanish	30	40	50

⑤A quick step to get the mean, median, minimum, maximum, and Q1 and Q3 for all numeric variables in the set

R Cmd: STATISTICS > SUMMARIES > ACTIVE DATA SET.

R Code: summary(writing)

```
     score                L1                condition
Min.   : 35.0   Arabic  :220   correctAll   :149
1st Qu.: 63.0   Japanese:120   correctTarget:196
Median :103.0   Russian :118   noCorrect    :233
Mean   :121.8   Spanish :120
3rd Qu.:163.8
Max.   :373.0
```

⑥Another method of obtaining the number of observations, minimum and maximum scores, mean, median, variance, standard deviation, and skewness and kurtosis numbers is the basicStats() function from the fBasics library.

```
install.packages("fBasics")
library(fBasics)
forget$rlwtest[1:15]
basicStats(forget$rlwtest[1:15])
```

Nobs 15.000000

NAs	0.000000
Minimum	47.000000
Maximum	91.000000
1. Quartile	56.500000
3. Quartile	79.000000
Mean	67.733333
Median	67.000000
Sum	1016.000000
SE Mean	3.552955
LCL Mean	60.113002
UCL Mean	75.353665
Variance	189.352381
Stdev	13.760537
Skewness	0.062303
Kurtosis	-1.373949

3.1.1 Skewness, Kurtosis, and Normality Tests with R P47

```
non=forget$rlwtest[1:15]
#chooses rows 1–15 of the variable to subset and names them ―non‖
skewness(non, na.rm=T)
kurtosis(non, na.rm=T)
```

```
> skewness(non, na.rm=T)
[1] 0.06230273
attr(,"method")
[1] "moment"
```

The skewness help file states that the default method of computing skewness is the ―moment‖ method.

```
> kurtosis(non, na.rm=T)
[1] -1.373949
attr(,"method")
[1] "excess"
```

choosing rows 1–15 could also use the subset() command through either DATA > ACTIVE DATA SET > SUBSET ACTIVE DATA SET or non <- subset(forget, subset=1:15)

```
> non <- subset(forget, subset=1:15)
> non
 [1] 67 51 50 47 82 80 86 56 63 70 91 78 67 57 71
```

the Shapiro–Wilk and Kolmogorov–Smirnoff tests.

統計学における、シャピロ–ウィルク検定（シャピロ–ウィルクけんてい）とは、標本 x1, ..., xn が正規母集団からサンプリングされたものであるという帰無仮説を検定する検定である。この検定方法は、サミュエル・シャピロとマーティン・ウィルクによって、1965 年に発表された。

コルモゴロフ–スミルノフ検定は統計学における仮説検定の一種であり、有限個の標本に基づいて、二つの母集団の確率分布が異なるものであるかどうか、あるいは母集団の確率分布が帰無仮説で提示された分布と異なっているかどうかを調べるために用いられる。しばしば KS 検定と略される。

```
> shapiro.test(non)   # Shapiro-Wilk normality test
data:  non
W = 0.96125, p-value = 0.7142

> ks.test(non,"pnorm") #One-sample Kolmogorov-Smirnov test
data:  non
D = 1, p-value = 1.872e-13
alternative hypothesis: two-sided
```

The nortest library provides five more tests of normality. Ricci (2005) states that the Lilliefors test is especially useful with small group sizes, and is an adjustment of the Kolmogorov–Smirnoff test.

```
install.packages("nortest")
library(nortest)
lillie.test(non)
    data:  non
    D = 0.1156, p-value = 0.8512
```

```
[71] エラー:
関数 "lillie.test" を見つけることができませんでした。
```

3.2 Application Activities with Numerical Summaries P49

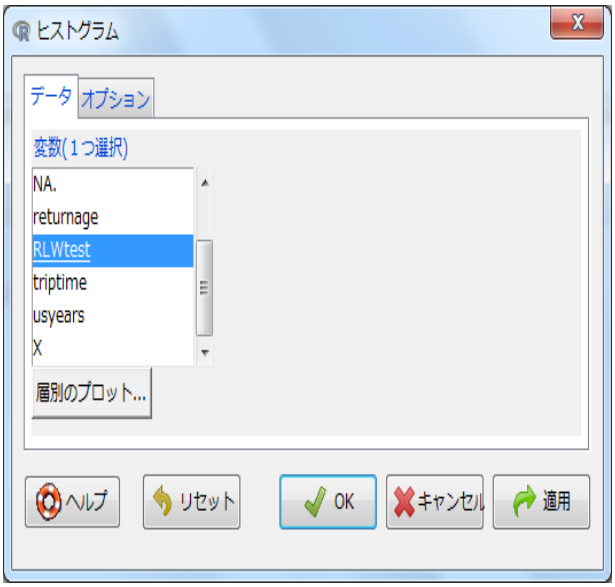
1. Import the SPSS file DeKeyser2000.sav and name it dekeyser. Summarize scores for the GJT grouped by the Status variable. Make sure you have data for the number of participants, the mean, and the standard deviation. By just eyeballing the statistics, does it look as though the groups have similar mean scores (maximum possible score was 200)? What about the standard deviation?

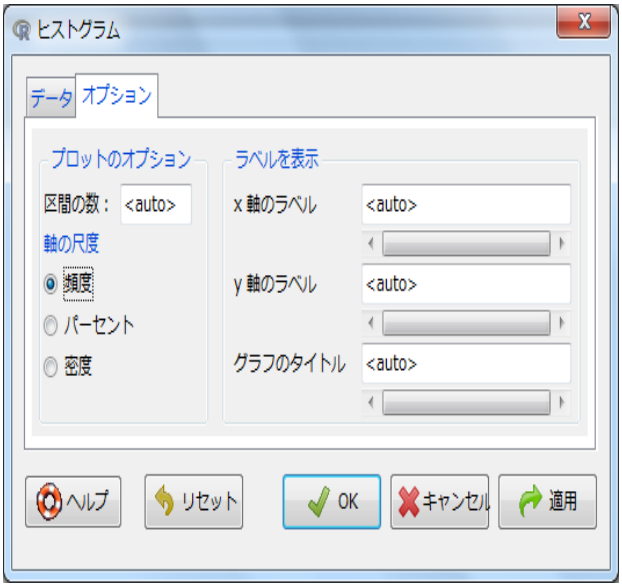
2. Import the SPSS file Obarow.sav and call it obarow (it has 81 rows and 35 columns). Summarize scores for the gain score in the immediate post-test (gnsc1.1) grouped according to the four experimental groups (trtmnt1). Each group was tested on 20 vocabulary words, but most knew at least 15 of the words in the pre-test. Obtain summaries for number of participants, mean scores, and standard deviations. Do the groups appear to have similar mean scores and standard deviations?

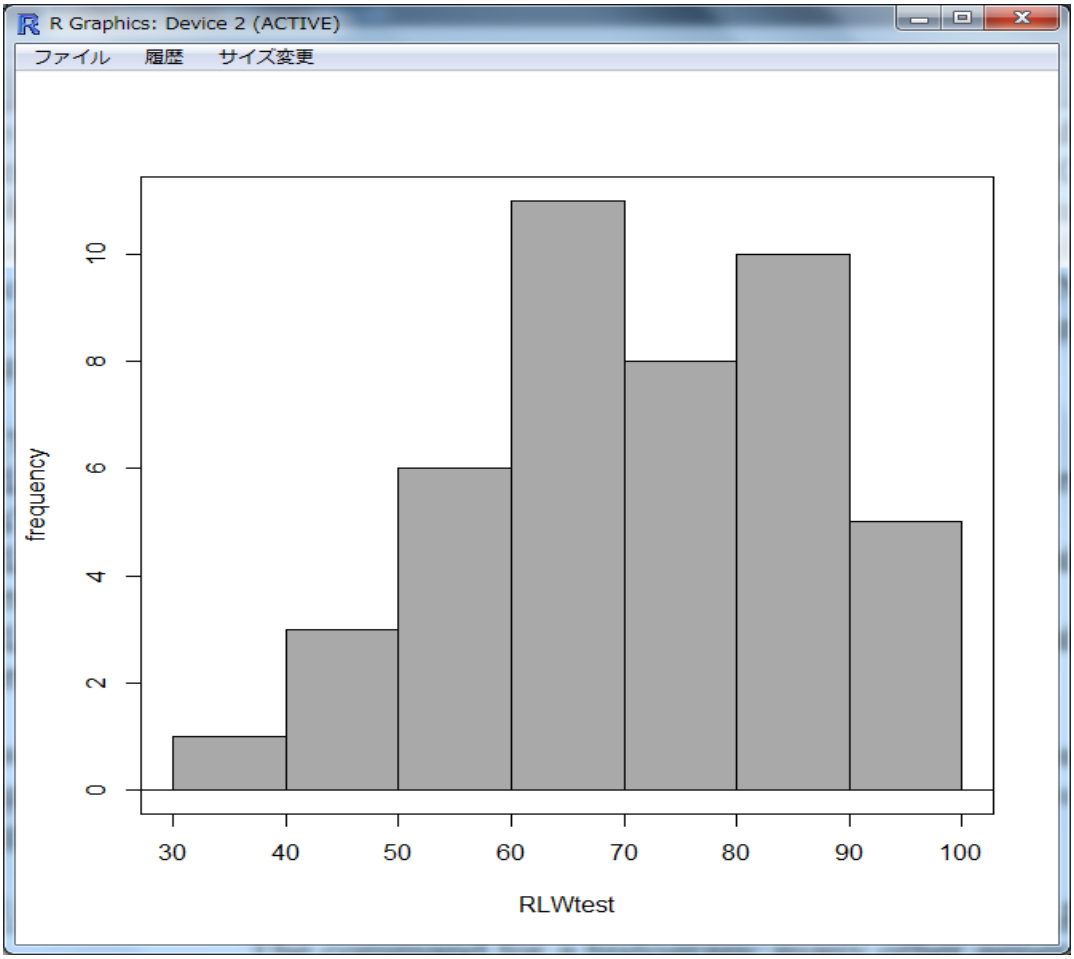
3.3 Generating Histograms, Stem and Leaf Plots, and Q-Q Plots P49

3.3.1 Creating Histograms with R P49

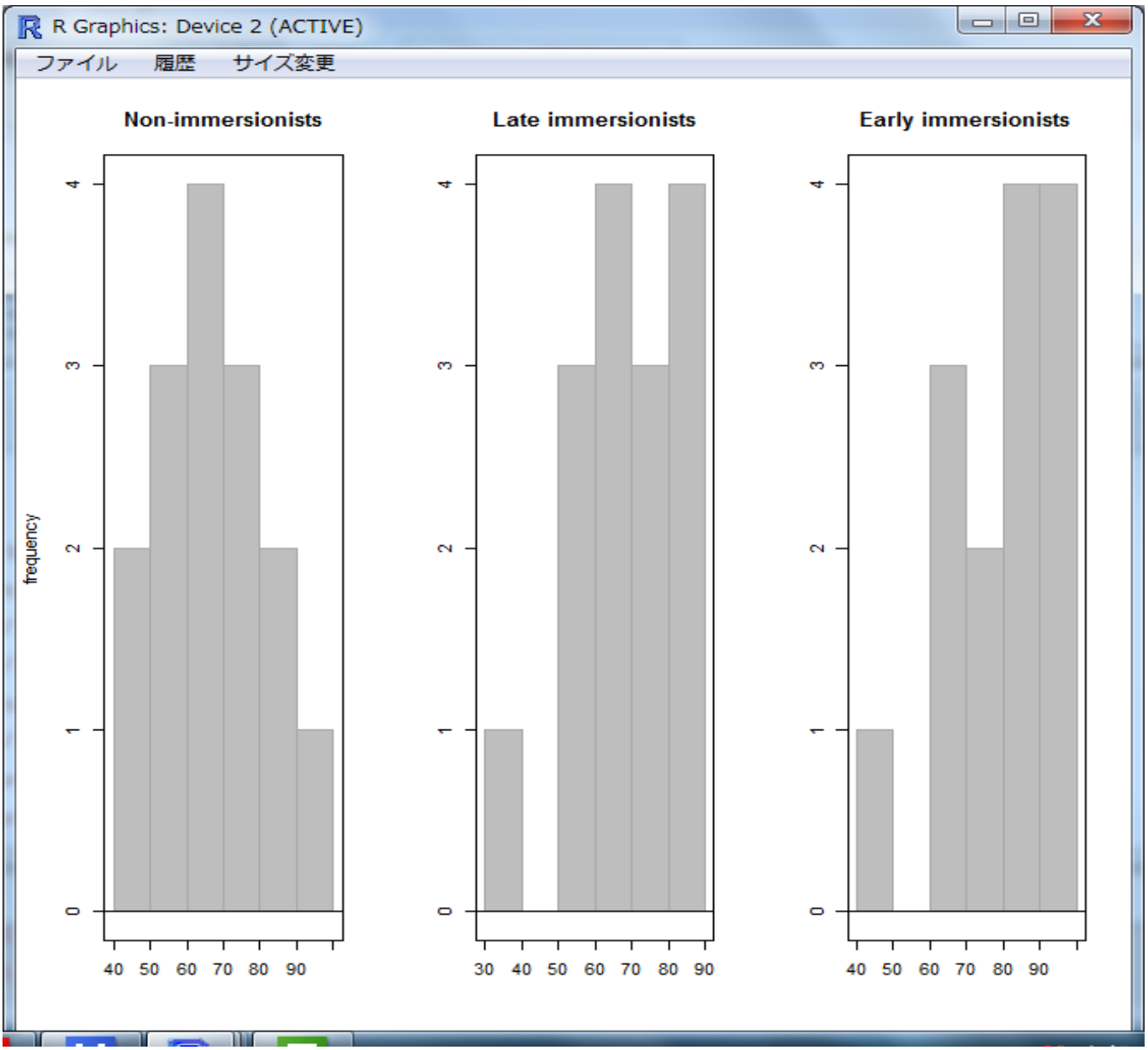
R command: GRAPHS > HISTOGRAM.





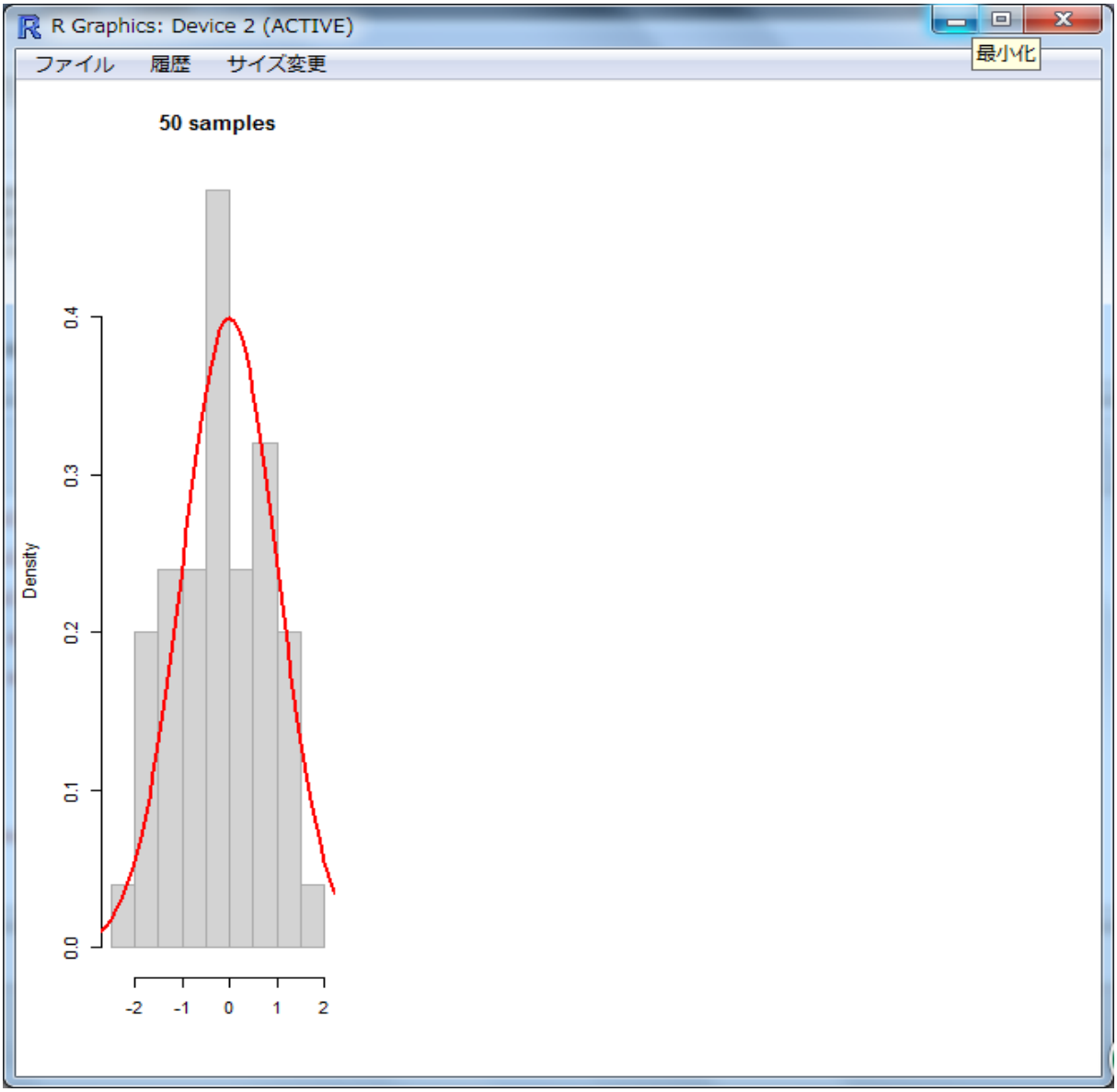


```
par(mfrow=c(1,3)) #sets the graphics display to 1 row, 3 columns
Hist(forget$rlwtest[1:15], col="gray", border="darkgray", xlab="",
main="Non-immersionists")
Hist(forget$rlwtest[16:30], col="gray", border="darkgray", xlab="", ylab="", main="Late
immersionists")
Hist(forget$rlwtest[31:44], col="gray", border="darkgray", xlab="", ylab="", main="Early
immersionists")
```



To overlay the histogram with a density plot of the normal distribution (Figure 3.3) I used the following code:

```
norm.x=rnorm(50,0,1)
x=seq(-3.5, 3.5, .1)
dn=dnorm(x)
hist(norm.x, xlab="", main="50 samples", col="lightgray", border="darkgray", prob=T)
lines(x, dn, col="red", lwd=2)
```



3.3.2 Creating Stem and Leaf Plots with R　P52

幹葉表示(みきはひょうじ)とは、簡易的なヒストグラムといえる。

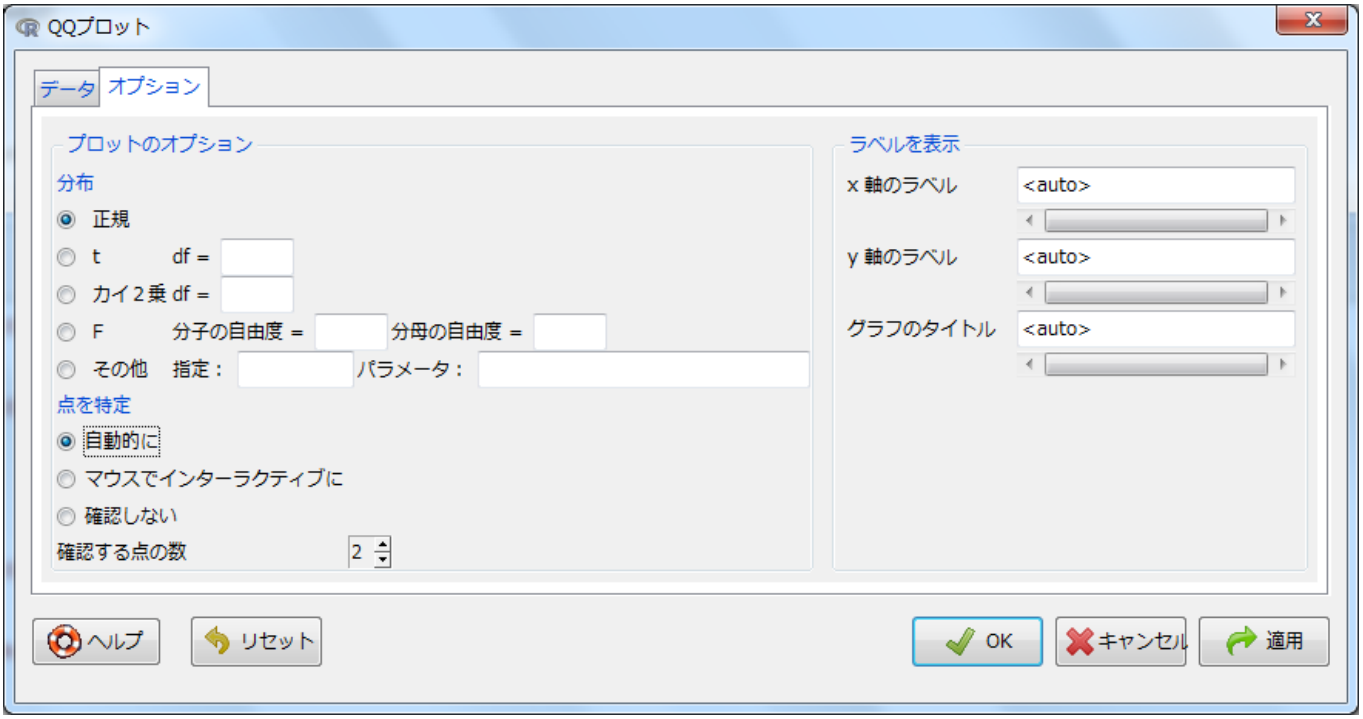
GRAPHS > STEM AND LEAF DISPLAY. or stem.leaf() creates the plot.

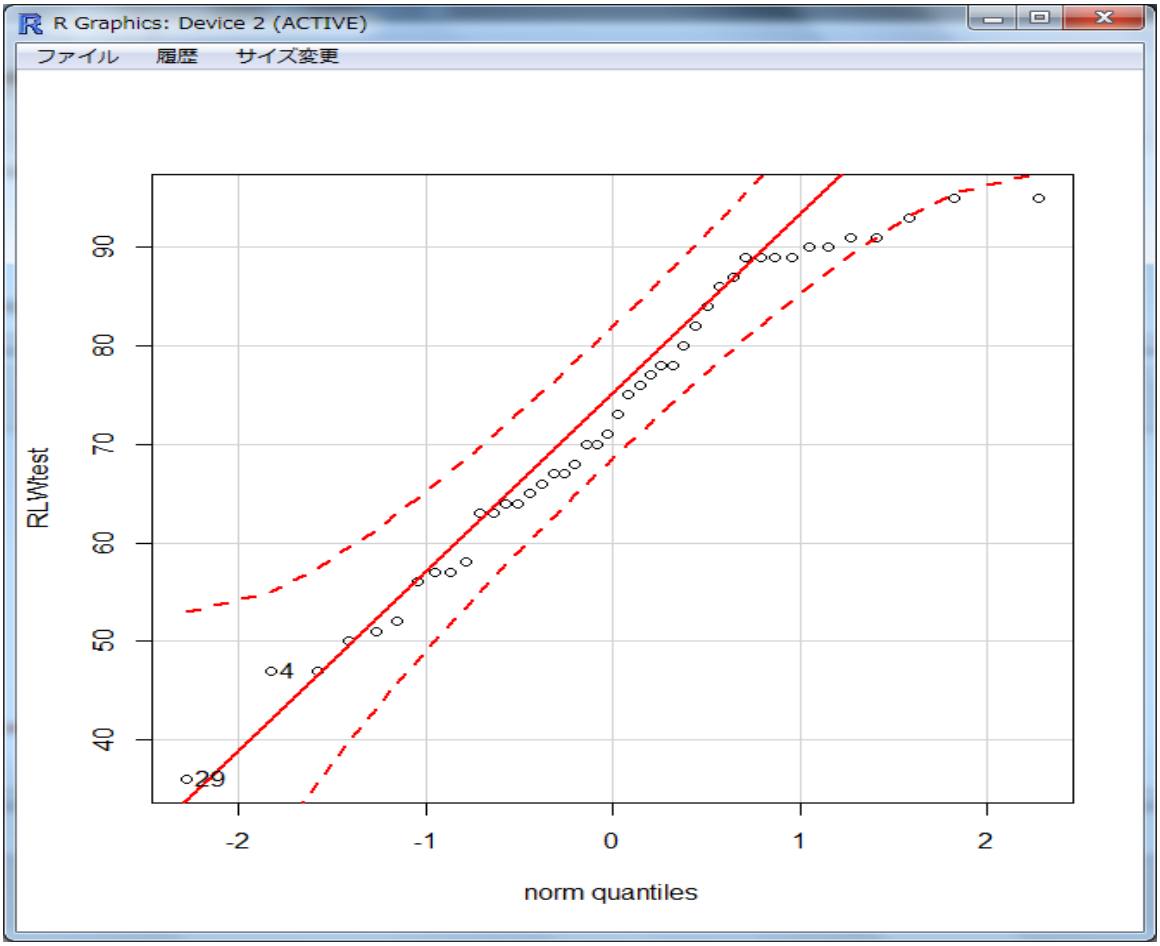
```
> with(forget, stem.leaf(rlwtest, na.rm=TRUE))

1 | 2: represents 12
 leaf unit: 1
                n: 44
    1       3.  | 6
            4*      |
    3       4.  | 77
    6       5*      | 012
   10       5.  | 6778
   14       6*      | 3344
   19       6.  | 56778
  (4)       7*      | 0013
   21       7.  | 56788
   16       8*      | 024
   13       8.  | 679999
    7       9*      | 00113
    2       9.  | 55
```

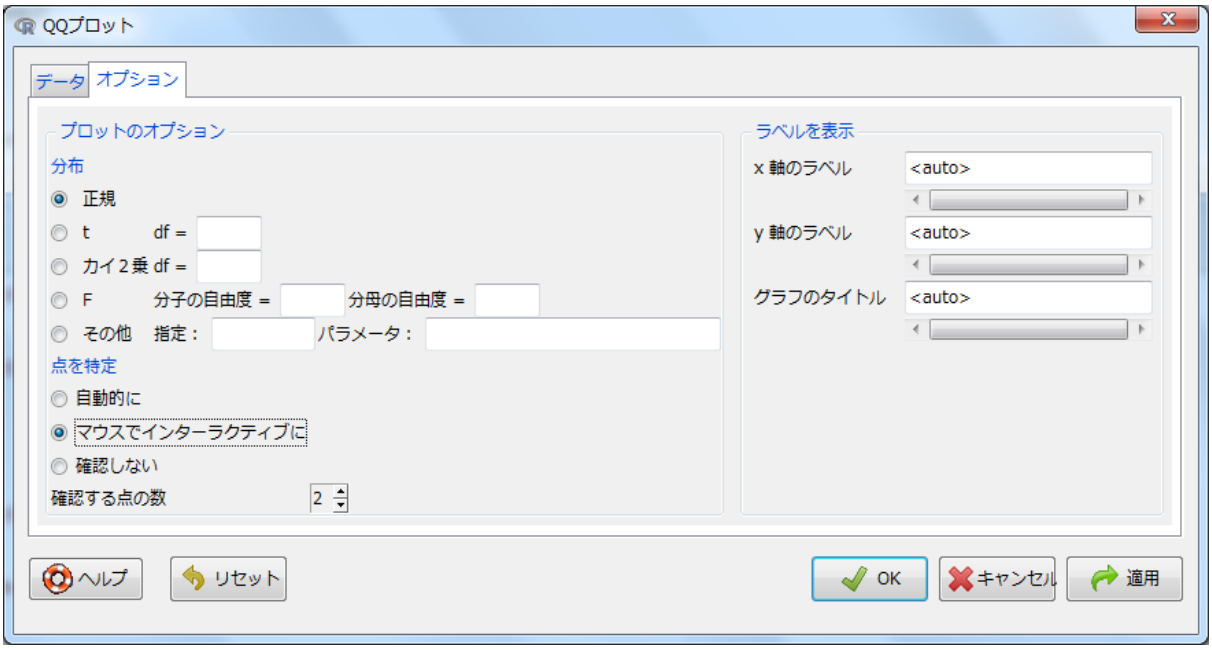
3.3.3. Creating Q-Q Plots with R　P52

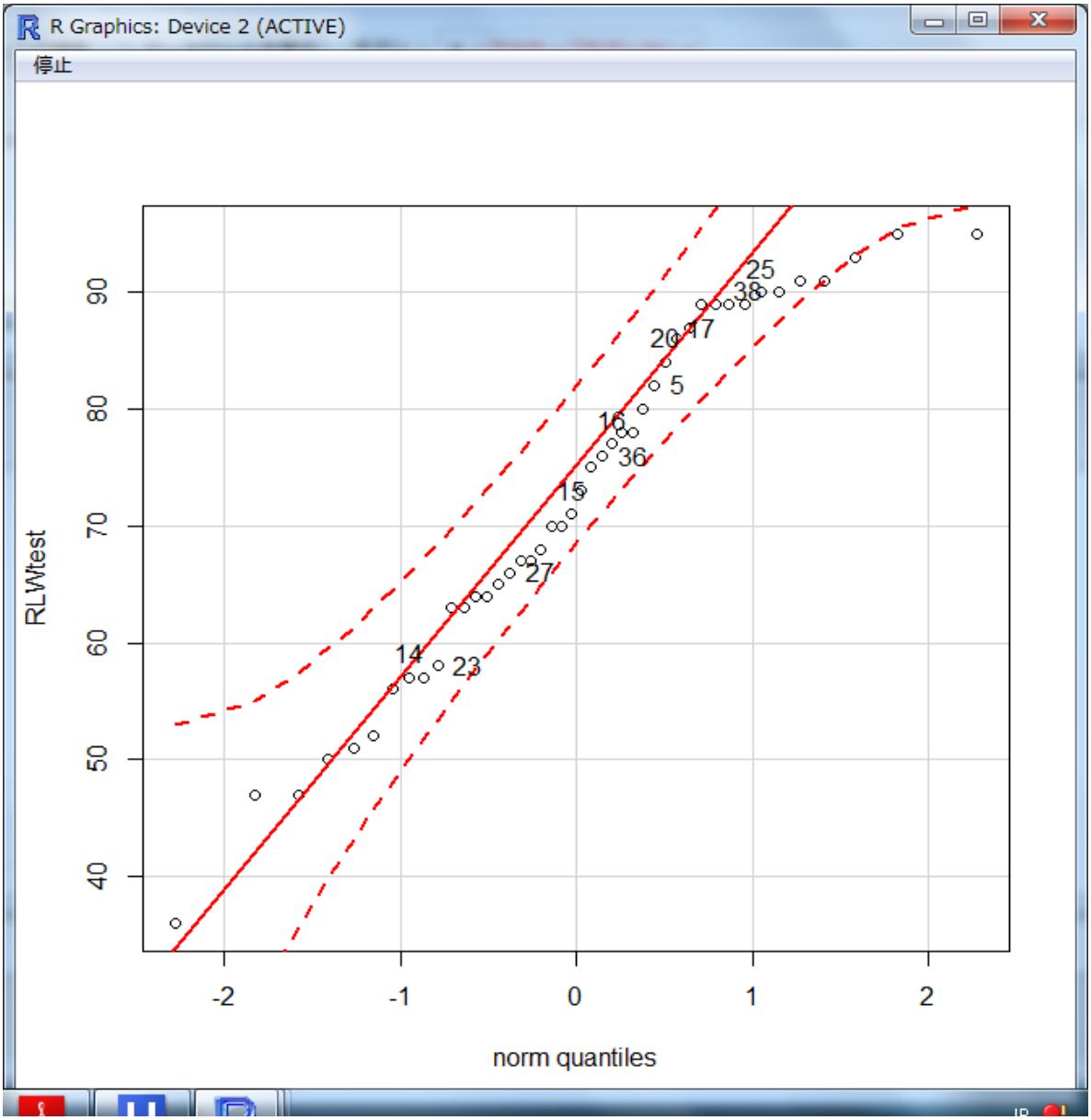
プロットが一直線上に並べば、観測値は正規分布に従っていると考えられる。

GRAPHS > QUANTILE-COMPARISON PLOT

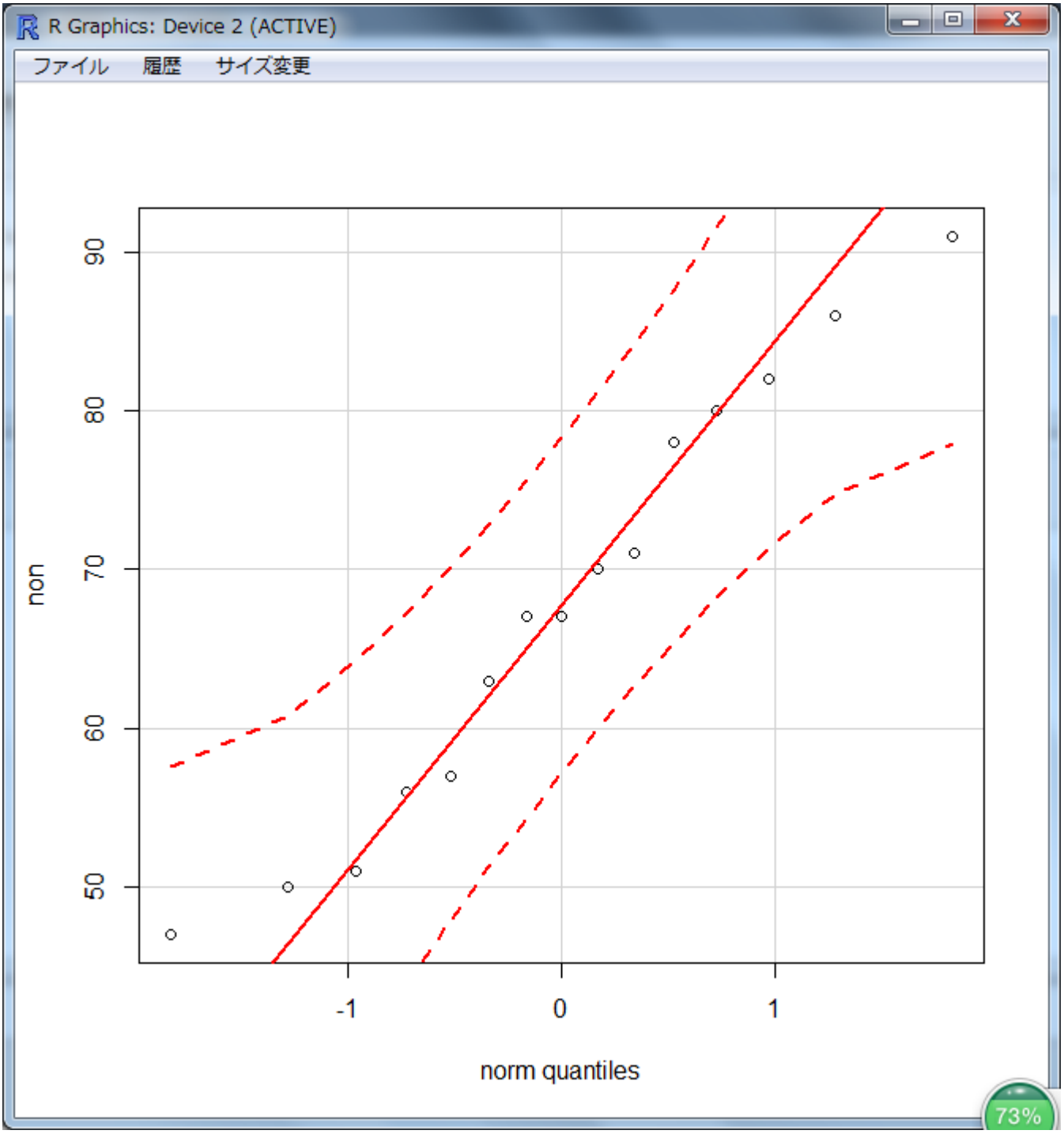





QQプロット
データ
オプション
プロットのオプション
分布
正規
t df =
カイ2乗 df =
F 分子の自由度 = 分母の自由度 =
その他 指定： パラメータ：
点を特定
自動的に
マウスでインターラクティブに
確認しない
確認する点の数
2
ラベルを表示
x 軸のラベル
<auto>
y 軸のラベル
<auto>
グラフのタイトル
<auto>
ヘルプ
リセット
OK
キャンセル
適用

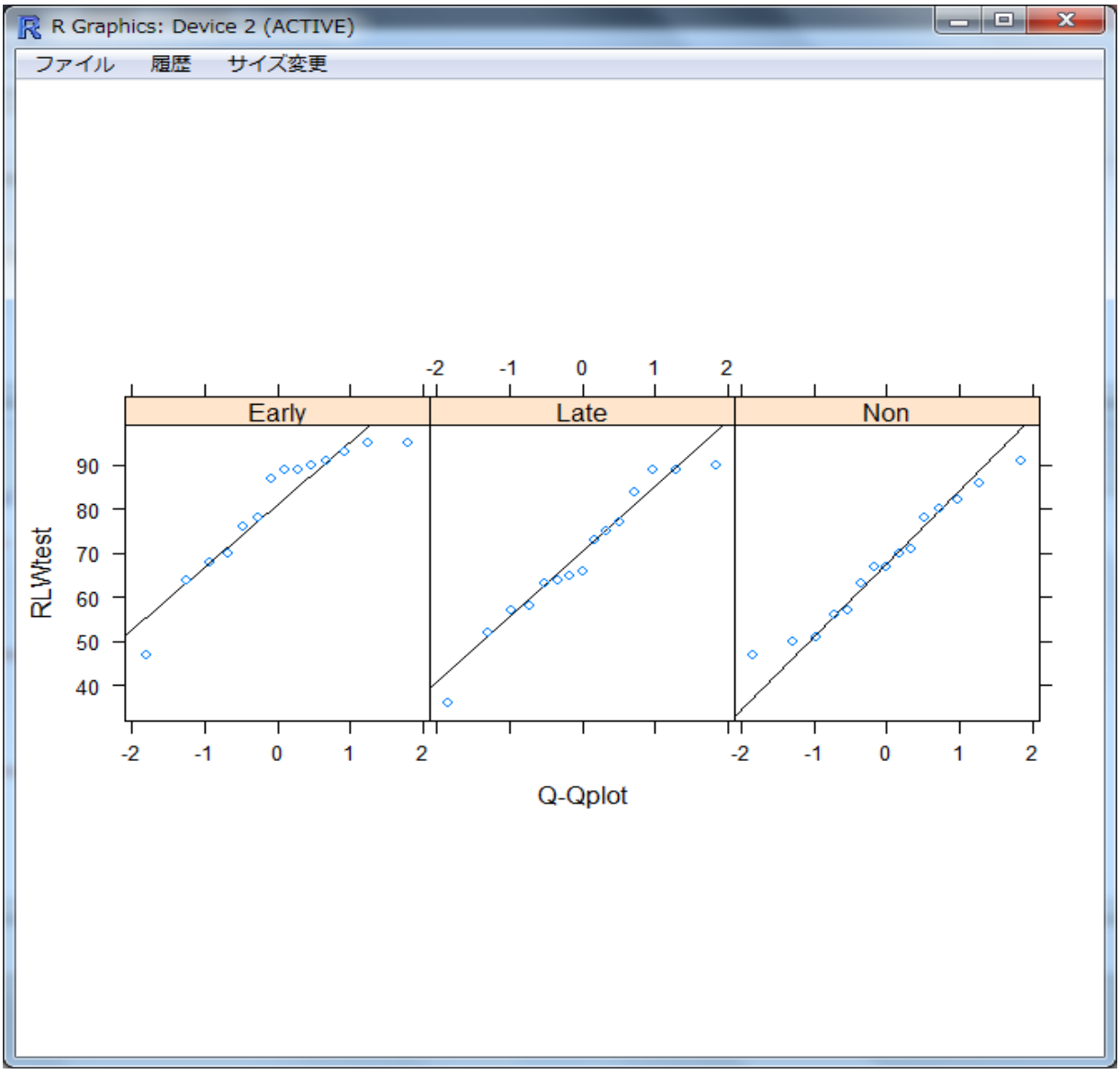

R Graphics: Device 2 (ACTIVE)
停止
RLWtest
norm quantiles
40
50
60
70
80
90
-2
-1
0
1
2
25
38
47
20
5
16
36
15
27
14
23

qq.plot(non, dist="norm", labels=F)



To produce a set of Q-Q plots for a variable split into groups without manually subsetting the data yourself, you can use the Lattice graphics library.

```
library(lattice)
qqmath(~rlwtest|Status, aspect="xy", data=forget,layout=c(3,1), xlab="Q-Qplot",
prepanel=prepanel.qqmathline,
panel=function(x, ...){
panel.qqmathline(x,...)
panel.qqmath(x,...)
})
```



3.4 Application Activities for Exploring Assumptions

P55 参照

3.5 Imputing Missing Data P56

Data: LafranceGottardo.sav
Name: lafrance
R Code:

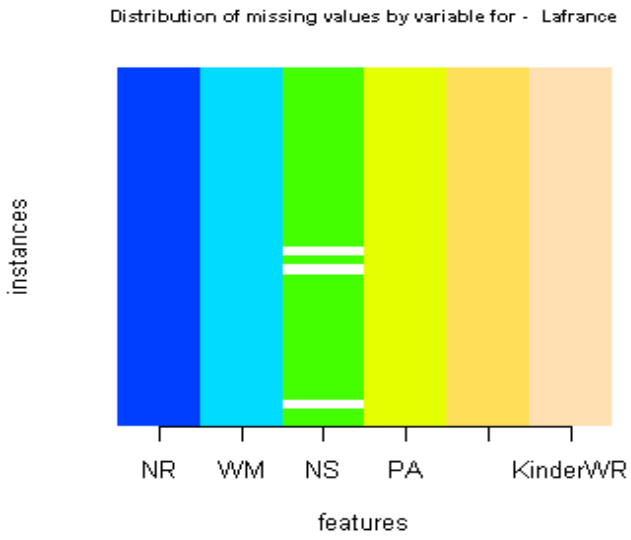
```
library(mice)
imp<-mice(lafrance)
complete(imp) #shows the completed data matrix
implafrance<-complete(imp) #name the new file to work with
library(dprep)
```

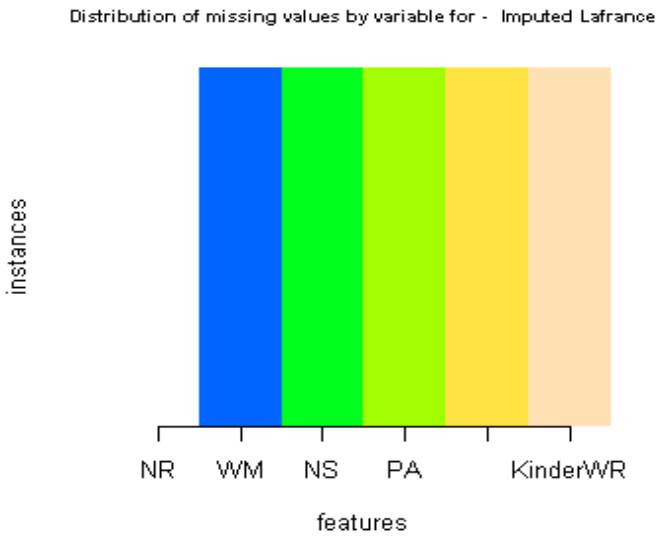
※ https://cran.r-project.org/web/packages/dprep/index.html

Package 'dprep' was removed from the CRAN repository.

Formerly available versions can be obtained from the archive.

```
imagmiss(lafrance, name="Lafrance")
imagmiss(implafrance, name="Lafrance imputed")
```

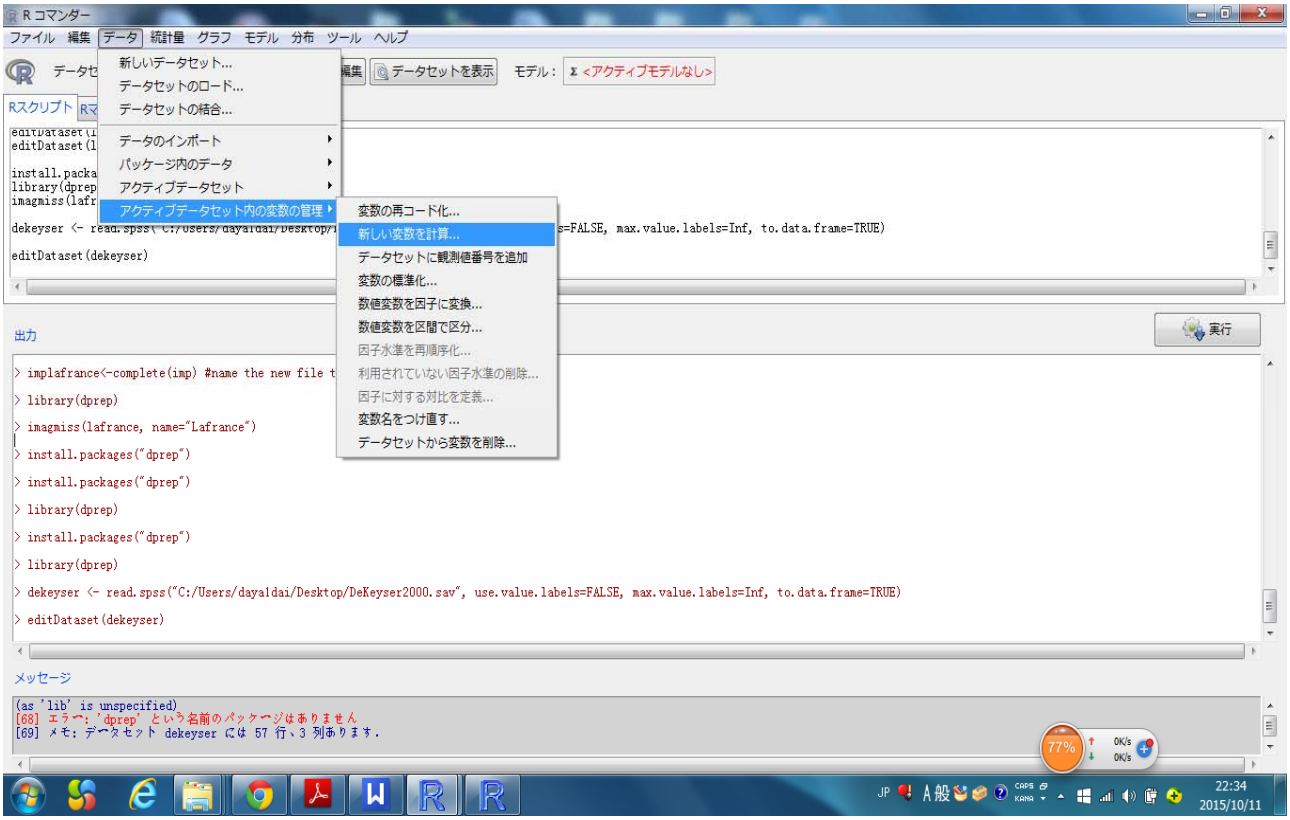


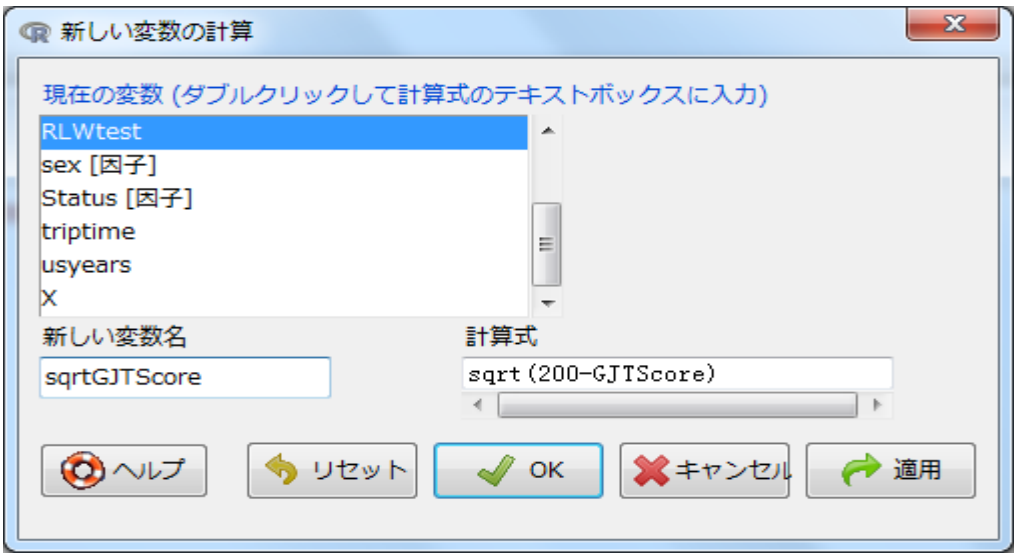


3.6 Transformations P57

Data: DeKeyser2000.sav
Naming: dekeyser
R command: DATA > MANAGE ACTIVE VARIABLES IN DATA SET > COMPUTE NEW VARIABLE.
R code: dekeyser$sqrtGJTScore <- with(dekeyser, sqrt(200-GJTScore))

The commands for transformations are not listed in the dialogue box itself. You will also have to add your own parentheses to the variable to show that the function,





※新しい変数名 sqrtGJTScore ； 計算式 sqrt(200-GJTScore)

```
with(dekeyser, Hist(GJTScore, scale="frequency", breaks="Sturges", col="darkgray"))
with(dekeyser, Hist(sqrtGJTScore, scale="frequency", breaks="Sturges", col="darkgray"))
```