R によるデータ処理

青木繁伸

2014 年 4 月 7 日

1 データ解析の例 1

```
> HEC <- read.csv("HairEyeColor.csv")
> HEC2 <- split(HEC, HEC$Sex)
> #
> lapply(HEC2, function(d) addmargins(table(d$Eye, d$Hair)))
    $Female

          Black Blond Brown Red Sum
  Blue        9    64    34   7 114
  Brown      36     4    66  16 122
  Green       2     8    14   7  31
  Hazel       5     5    29   7  46
  Sum        52    81   143  37 313

    $Male

          Black Blond Brown Red Sum
  Blue       11    30    50  10 101
  Brown      32     3    53  10  98
  Green       3     8    15   7  33
  Hazel      10     5    25   7  47
  Sum        56    46   143  34 279
> #
> with(HEC, ftable(addmargins(table(Sex, Eye, Hair)),
+                  row.vars=c("Sex", "Eye"), col.vars="Hair"))
                Hair Black Blond Brown Red Sum
    Sex    Eye
    Female Blue           9    64    34   7 114
           Brown         36     4    66  16 122
           Green          2     8    14   7  31
           Hazel          5     5    29   7  46
           Sum           52    81   143  37 313
    Male   Blue          11    30    50  10 101
           Brown         32     3    53  10  98
           Green          3     8    15   7  33
           Hazel         10     5    25   7  47
           Sum           56    46   143  34 279
    Sum    Blue          20    94    84  17 215
           Brown         68     7   119  26 220
           Green          5    16    29  14  64
           Hazel         15    10    54  14  93
           Sum          108   127   286  71 592
```

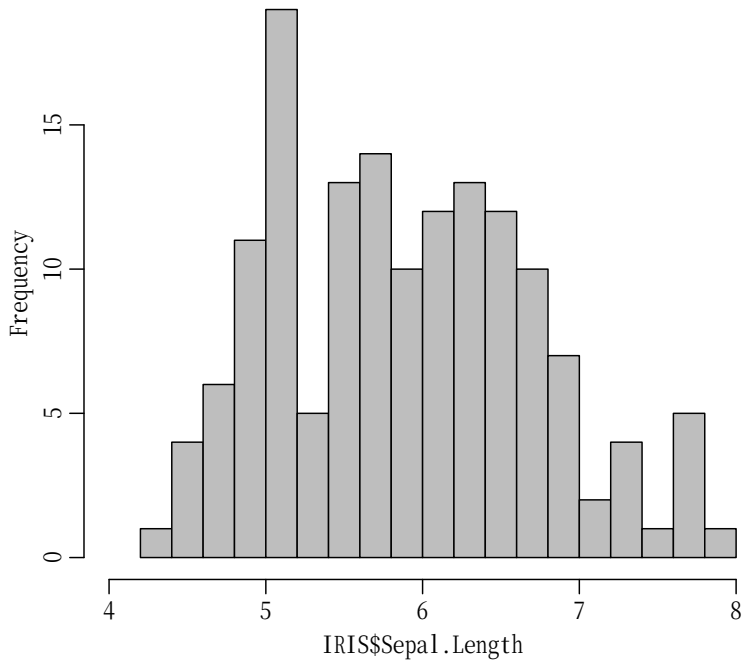

図 1　ヒストグラム

2　データ解析の例 2

```
> IRIS <- read.csv("iris.csv")
```

2.1　ヒストグラム

```
> hist(IRIS$Sepal.Length, breaks=seq(4.2, 8.0, by=0.2), right=FALSE, col="gray", xlim=c(4, 8))
```

2.2　集計表

```
> IRIS2 <- split(IRIS[1:4], IRIS$Species)
> # mean
> t(sapply(IRIS2, colMeans))
             Sepal.Length Sepal.Width Petal.Length Petal.Width
  setosa            5.006       3.428        1.462       0.246
  versicolor        5.936       2.770        4.260       1.326
  virginica         6.588       2.974        5.552       2.026
> # variance
> t(sapply(IRIS2, function(x) apply(x, 2, var)))
             Sepal.Length Sepal.Width Petal.Length Petal.Width
  setosa        0.1242490  0.14368980   0.03015918  0.01110612
  versicolor    0.2664327  0.09846939   0.22081633  0.03910612
  virginica     0.4043429  0.10400408   0.30458776  0.07543265
```

```
> # SD
> t(sapply(IRIS2, function(x) apply(x, 2, sd)))
             Sepal.Length Sepal.Width Petal.Length Petal.Width
  setosa        0.3524897   0.3790644    0.1736640   0.1053856
  versicolor    0.5161711   0.3137983    0.4699110   0.1977527
  virginica     0.6358796   0.3224966    0.5518947   0.2746501
```

3 データ解析の例 3

3.1 グラフをファイルに保存する

以下のようにすれば，描画されるグラフが一つずつ，連番名を持つファイルに保存される。
もちろん，グラフごとに個別のファイル名を付けて，サイズも変えてファイルに保存することもできる。

```
> pdf("fig%02i.pdf", width=500/72, height=400/72, onefile=FALSE)
```

3.2 分析するデータの入力

R では，データセットは，データフレーム data.frame に保管される。入力は read.table 関数による。

```
> (ap <- read.table("AirPassengers.dat", header=TRUE))
      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949 112 118 132 129 121 135 148 148 136 119 104 118
1950 115 126 141 135 125 149 170 170 158 133 114 140
1951 145 150 178 163 172 178 199 199 184 162 146 166
1952 171 180 193 181 183 218 230 242 209 191 172 194
1953 196 196 236 235 229 243 264 272 237 211 180 201
1954 204 188 235 227 234 264 302 293 259 229 203 229
1955 242 233 267 269 270 315 364 347 312 274 237 278
1956 284 277 317 313 318 374 413 405 355 306 271 306
1957 315 301 356 348 355 422 465 467 404 347 305 336
1958 340 318 362 348 363 435 491 505 404 359 310 337
1959 360 342 406 396 420 472 548 559 463 407 362 405
1960 417 391 419 461 472 535 622 606 508 461 390 432
```

3.3 データの調整

データは年ごとに 1 月から 12 月までの 12 個のデータが並んでいるので，これを一続きのベクトルとして扱うようにする。また，月はカテゴリー変数として用いる。さらに，時間変数としてデータの通し番号（1〜144）を使う。
以上の 3 つの変数をデータフレームとして用意する。

```
> AirPassengers <- as.vector(t(as.matrix(ap)))
> Month <- factor(month.abb, levels=month.abb)
> x <- 1:length(AirPassengers)
> df <- data.frame(AirPassengers, Month, x)
> head(df)
    AirPassengers Month x
1             112   Jan 1
2             118   Feb 2
3             132   Mar 3
4             129   Apr 4
5             121   May 5
6             135   Jun 6
> tail(df)
      AirPassengers Month   x
139             622   Jul 139
```

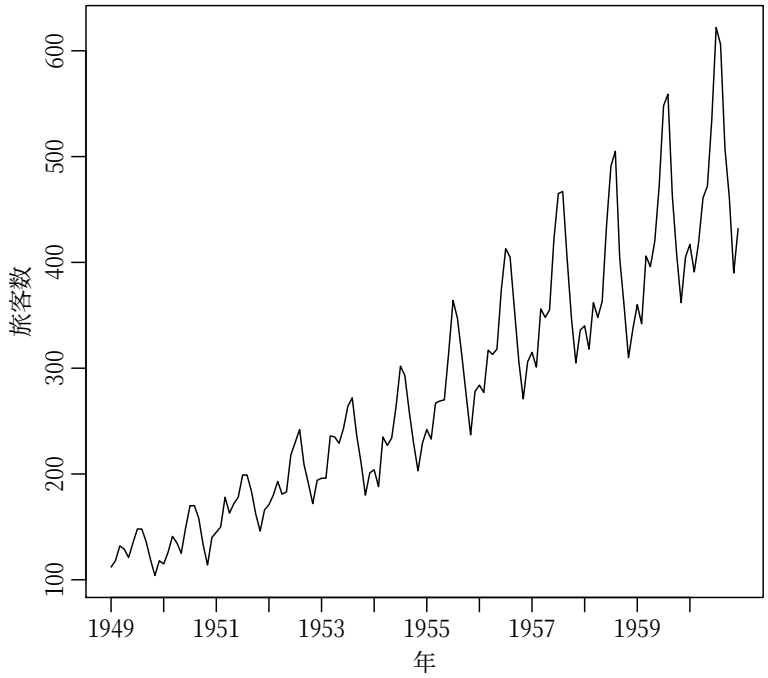

図 2　図 2 に相当するグラフ

```
140           606   Aug 140
141           508   Sep 141
142           461   Oct 142
143           390   Nov 143
144           432   Dec 144
```

3.4　航空旅客数の時系列図

横軸に時間，縦軸に旅客数をとって折れ線図を描くと，図 2 のようになる。

```
> plot(x, df$AirPassengers, type="l", xaxt="n", xlab="年", ylab="旅客数")
> axis(1, at=0:11*12+1, labels=1949:1960)
```

3.5　月ごとの統計

3.5.1　平均値など

1949 年から 1960 年までの 12 年間のデータについて，各月の平均値，分散，標準偏差を求める。

```
> mean <- colMeans(ap)
> sd <- apply(ap, 2, sd)
> var <- sd^2
> cbind(mean, var, sd)
           mean       var        sd
Jan 241.7500 10207.659 101.03296
Feb 235.0000  8031.636  89.61940
Mar 270.1667 10112.152 100.55919
Apr 267.0833 11529.356 107.37484
```

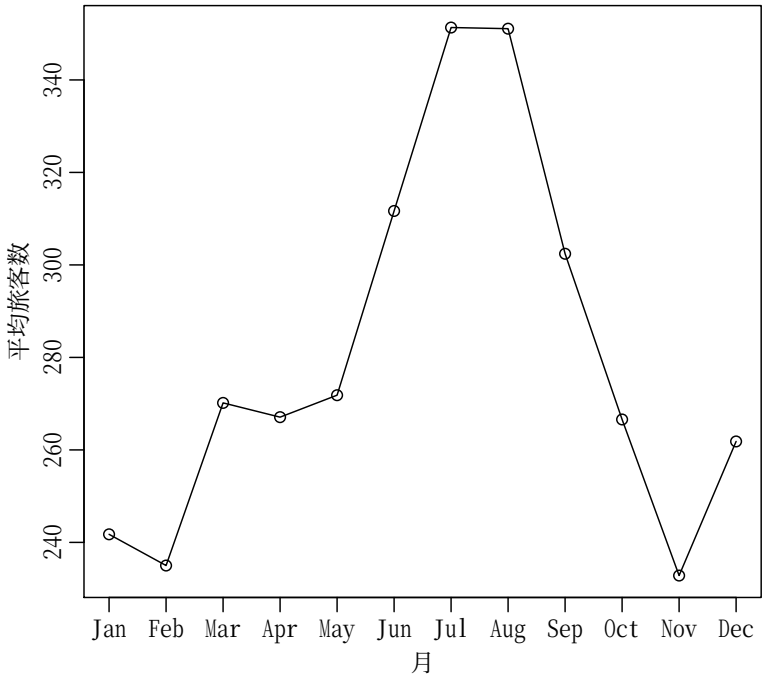

図 3　図 4 に相当するグラフ

```
May 271.8333 13165.242 114.73989
Jun 311.6667 18014.970 134.21986
Jul 351.3333 24594.788 156.82725
Aug 351.0833 24268.447 155.78333
Sep 302.4167 15364.629 123.95414
Oct 266.5833 12264.447 110.74496
Nov 232.8333  9060.333  95.18578
Dec 261.8333 10628.333 103.09381
```

3.5.2　折れ線図

グラフは図 3 のようになる。

```
> plot(1:12, mean, type="o", xaxt="n", xlab="月", ylab="平均旅客数")
> axis(1, 1:12, Month)
```

3.6　年ごとの統計

3.6.1　平均値

```
> (mean <- rowMeans(ap))
     1949     1950     1951     1952     1953     1954     1955     1956
126.6667 139.6667 170.1667 197.0000 225.0000 238.9167 284.0000 328.2500
     1957     1958     1959     1960
368.4167 381.0000 428.3333 476.1667
```

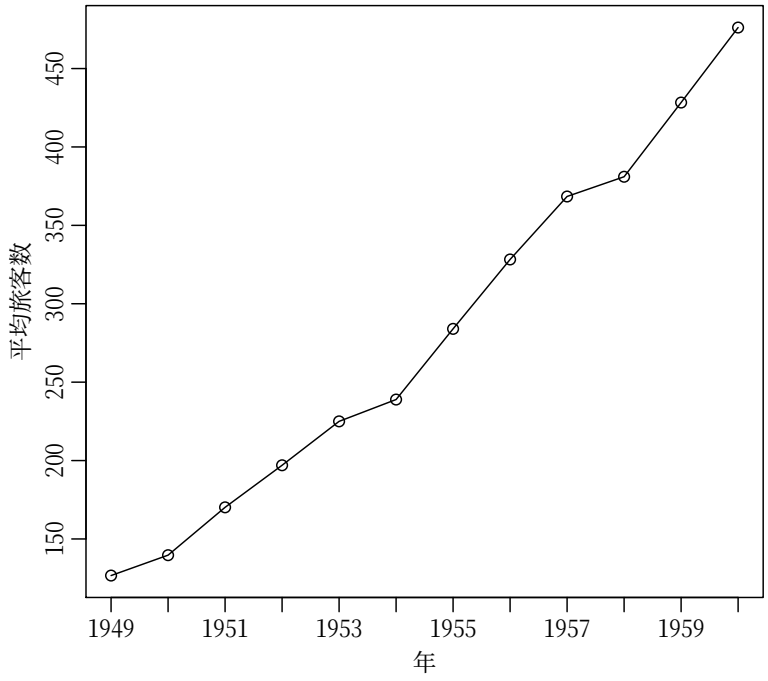

図 4　図 6 に相当するグラフ

3.6.2　**折れ線図**

グラフは図 4 のようになる。

```
> plot(1:12, mean, type="o", xaxt="n", xlab="年", ylab="平均旅客数")
> axis(1, 1:12, 1949:1960)
```

3.6.3　**理論曲線へのあてはめ**

非線形最小二乗法により,

$$\text{年平均値} = a\ \exp(b\,x)$$

という指数曲線へ当てはめる。グラフは図 5 のようになる。

```
> ans <- nls(mean ~ a*exp(b*1:12), start=list(a=117, b=0.12))
> summary(ans)
    Formula: mean ~ a * exp(b * 1:12)

    Parameters:
       Estimate Std. Error t value Pr(>|t|)
    a 1.241e+02  4.960e+00   25.01 2.39e-10 ***
    b 1.140e-01  4.258e-03   26.77 1.22e-10 ***
    ---
    Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

    Residual standard error: 12.94 on 10 degrees of freedom
```

```
Formula: mean ~ a * exp(b * 1:12)

Parameters:
   Estimate Std. Error t value Pr(>|t|)
a 1.241e+02  4.960e+00   25.01 2.39e-10 ***
b 1.140e-01  4.258e-03   26.77 1.22e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.94 on 10 degrees of freedom

Number of iterations to convergence: 4
Achieved convergence tolerance: 7.335e-07
```

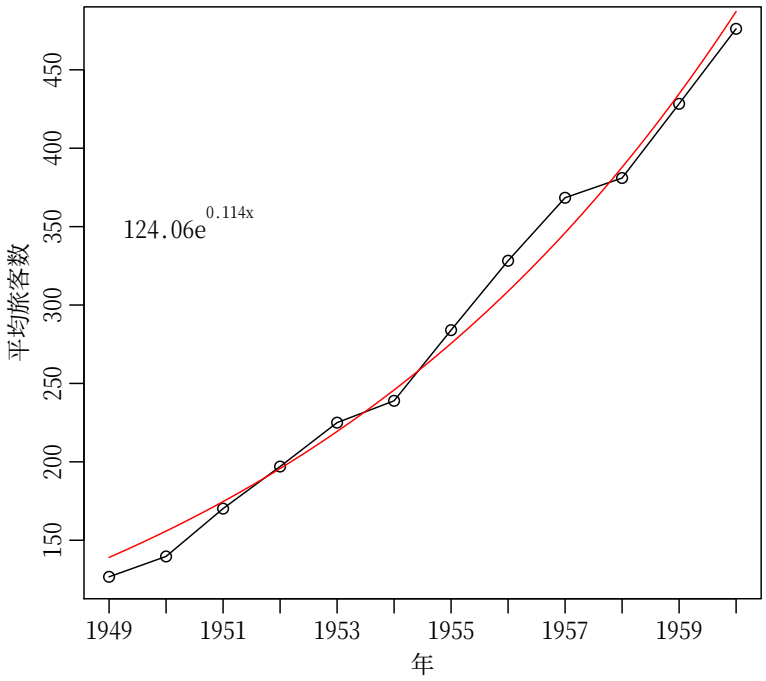

図 5　図 7 に相当するグラフ

```
   Number of iterations to convergence: 4
   Achieved convergence tolerance: 7.335e-07
> a <- coef(ans)[1]
> b <- coef(ans)[2]
> x1 <- seq(1, 12, length=200)
> y1 <- a*exp(b*x1)
> lines(x1, y1, col="red")
> text(1, 350, substitute(a*e^{b* x}, list(a=round(a, 2), b=round(b, 4))), pos=4)
```

3.6.4　年増加率

```
> (rate <- mean[-1]/mean[-12]) # 年ごとの
     1950      1951      1952      1953      1954      1955      1956      1957
1.102632 1.218377 1.157689 1.142132 1.061852 1.188699 1.155810 1.122366
```

```
        1958     1959     1960
    1.034155 1.124234 1.111673
> (mean.rate <- exp(mean(log(rate)))) # 平均増加率（幾何平均）
    [1] 1.127928
> cbind(年平均値=mean, 予測式から=a*exp(b*1:12),
+       平均増加率から=mean[1]*mean.rate^(0:11))
          年平均値   予測式から 平均増加率から
    1949 126.6667 139.0411    126.6667
    1950 139.6667 155.8294    142.8709
    1951 170.1667 174.6447    161.1482
    1952 197.0000 195.7319    181.7636
    1953 225.0000 219.3652    205.0163
    1954 238.9167 245.8521    231.2437
    1955 284.0000 275.5370    260.8263
    1956 328.2500 308.8063    294.1934
    1957 368.4167 346.0925    331.8291
    1958 381.0000 387.8809    374.2794
    1959 428.3333 434.7148    422.1604
    1960 476.1667 487.2037    476.1667
```

3.7 旅客数の予測

3.7.1 予測式を求める

月，時間変数を独立変数，旅客数を従属変数として重回帰分析を行う。

```
> ans <- lm(AirPassengers~Month+x, df)
```

```
> summary(ans)
    Call:
    lm(formula = AirPassengers ~ Month + x, data = df)

    Residuals:
        Min      1Q  Median      3Q     Max
    -42.121 -18.564  -3.268  15.189  95.085

    Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
    (Intercept)  63.50794    8.38856   7.571 5.88e-12 ***
    MonthFeb     -9.41033   10.74941  -0.875 0.382944
    MonthMar     23.09601   10.74980   2.149 0.033513 *
    MonthApr     17.35235   10.75046   1.614 0.108911
    MonthMay     19.44202   10.75137   1.808 0.072849 .
    MonthJun     56.61502   10.75254   5.265 5.58e-07 ***
    MonthJul     93.62136   10.75398   8.706 1.17e-14 ***
    MonthAug     90.71103   10.75567   8.434 5.32e-14 ***
    MonthSep     39.38403   10.75763   3.661 0.000363 ***
    MonthOct      0.89037   10.75985   0.083 0.934177
    MonthNov    -35.51996   10.76232  -3.300 0.001244 **
    MonthDec     -9.18029   10.76506  -0.853 0.395335
    x             2.66033    0.05297  50.225  < 2e-16 ***
    ---
    Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

    Residual standard error: 26.33 on 131 degrees of freedom
    Multiple R-squared:  0.9559,        Adjusted R-squared:  0.9518
    F-statistic: 236.5 on 12 and 131 DF,  p-value: < 2.2e-16
```

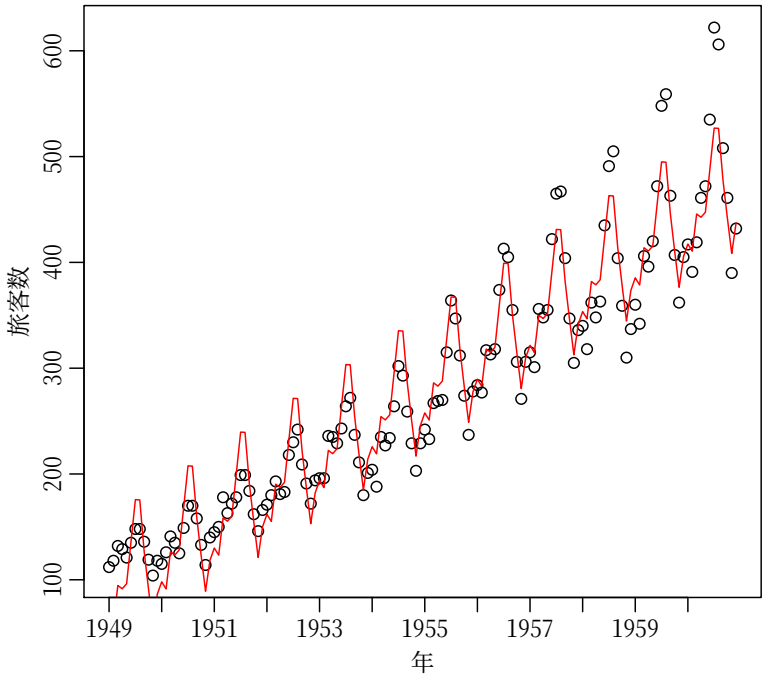

図 6　図 11 に相当するグラフ

3.7.2　あてはめ結果

あてはめ結果のグラフは図 6 のようになる。

```
> plot(x, AirPassengers, type="p", xaxt="n", xlab="年", ylab="旅客数")
> lines(x, ans$fitted.values, col="red")
> axis(1, at=0:11*12+1, labels=1949:1960)
```

3.7.3　予測値と実測値の散布図

予測値を横軸，実測値を縦軸に取って描いたグラフは図 7 のようになる。

```
> plot(ans$fitted.values, AirPassengers, xlab="予測値", ylab="実測値")
> abline(0, 1, col="red")
```

3.7.4　将来予測

将来予測のグラフは図 8 のようになる。

```
> plot(x, AirPassengers, type="l", xaxt="n", xlab="年", ylab="旅客数",
+      xlim=c(1, 12*17), ylim=c(0, 700))
> new.x <- (12*12+1):(12*17)
> new <- data.frame(x=new.x, Month=factor(month.abb, levels=month.abb))
> pred <- predict(ans, new)
> lines(new.x, pred, col="red")
```

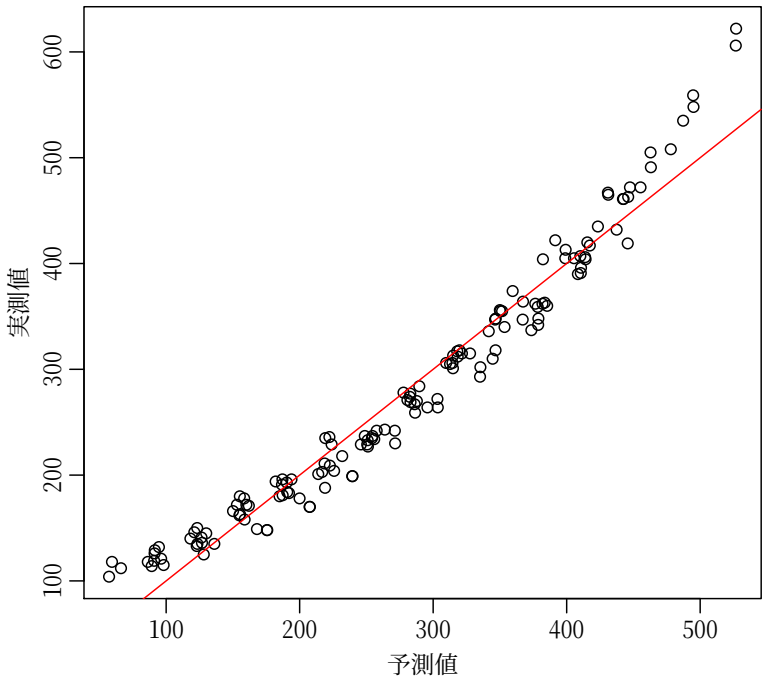

図 7　図 12 に相当するグラフ

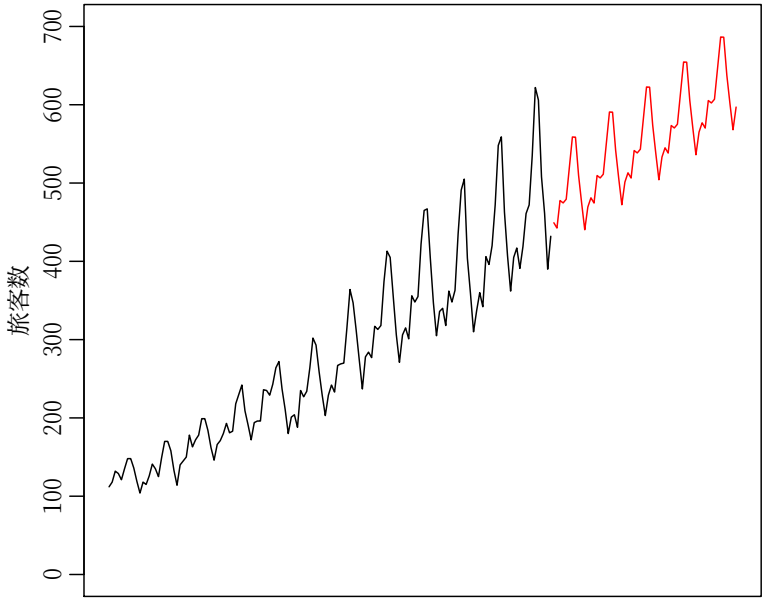

図 8　図 13 に相当するグラフ

3.8 ファイル出力の終了

ファイル出力を終了する。

```
> dev.off()
```