```
# 一日目
1
   # 正規線形モデリング
2
3
  # 2013年8月28最終改訂
4
5
   # 執筆者 馬場真哉 (北大水産 M2)
   # ウェブサイト http://logics-of-blue.com/
6
   # ミスなどのご連絡は logics.of.blue★gmail.com までお願いします。
7
   # (星を@に変更)
8
9
10
   # Rによる統計モデリング
11
   # この章は全てのプログラムを記述~実行する
12
13
   14
15
   # ****
          写経はじめ
16
   # ****
17
18
19
20
   # こんにちは世界
21
   print("Hallow World")
22
23
   # print は省略可能
24
   "Hallow World"
25
   # ""を抜かすとエラー
26
27
   Hallow World
28
29
  # 簡単な計算
30
  1 + 2
  5*6
31
32
  3 + 4/2 - 1
33
   2^10
34
35 # コメント部の説明
36 # 1 + 2
```

```
37
38
    # 変数の定義
39
    # 「<-」の左右にはスペースを必ずあけること。
40
41
    # 見やすいプログラムを書く癖をつけた方がよい
42
   x <- 3
43
44
   Х
45
   x*2
46
47
    # データの読み込みと表示
48
    # データをコピーしてから実行する
49
50
    data.0.clip <- read.delim("clipboard")</pre>
51
   data.0.clip
52
   head(data.0.clip)
53
   pairs(data.0.clip)
54
    # データの読み込み方法その2
55
    # 作業ディレクトリを変更してから以下を実行
56
57
    getwd()
58
59
    # Not Run
60
   # setwd()
61
    # 任意のワーキングディレクトリを指定
    # コンソールをクリック→ファイル→ディレクトリの変更 から指定できる
62
    # getwd()で場所を確認したうえで setwd()に値をコピペで渡すと楽。
63
64
65
    # 読み込みはこの一行でOK
    data.0 <- read.csv("data0.csv")</pre>
66
67
    data.0
68
   head(data.0)
69
   pairs(data.0)
70
71
72
    # データの中身の取出し
```

```
# エラーが出る
74
    names(data.0) # data.0 に格納されているモノの名前を調べる
75
76
    data.0$food # OK
77
78
    # プロット
79
80
    # 興味の対象を「~」の左側におく
    # 興味の対象 length はえさの量によってどう変わるか?
81
82
    plot(data.0$length ~ data.0$food)
83
    # 興味の対象 length は薬の有無によってどう変わるか?
84
85
    plot(data.0$length ~ data.0$medicine)
86
87
    # ちょっと複雑なプロット
88
    # 長いプログラムは分かりやすく改行するとよい
89
90
    # インデント(左端にスペースを空ける)するとなお見やすい。
    # Google 基準ではインデントはスペース 2 つ分
91
    # ただし丸かっこで囲まれている場合は、一行目に合わせてインデントを付ける
92
93
    # ↑ただし、これはめんどくさいので私は使っていない。スペース 2 つで十分と思う。
    # 無理に Google に合わせる必要はないが、改行&インデントは実務上必須。
94
    # 私はタブを使うこともよくあるが、Google 基準では使ってはいけないらしい?
95
96
97
    # 見づらいコードを書き続けると、いつか"必ず"後悔することになるので注意。
98
    # Google に合わせる必要は皆無だが、自分なりにルールを決める必要はある。
99
100
    plot(
101
     data.0$length ~ data.0$food,
102
     col = c(2,3) [data.0\$medicine],
103
     pch=16,
104
     ylab = "Length",
105
     xlab = "food",
106
     main = "薬の有無別、体長と餌の量の関係",
107
     cex.main = 1.5,
108
     font.lab = 2
```

attach()関数は"絶対に"使わないこと。

```
109
    )
110
    # 凡例
111
112
    legend(
113
     "topleft",
     legend = c("薬あり", "薬なし"),
114
115
     col = c(2,3),
116
     bty = "n",
117
    pch = 16
118
    )
119
     # ヘルプ
120
121
    ?legend
122
     # モデルの作成
123
124
     # 応答変数:length
125
     # 説明変数:food & medicine
126
     # 興味のある対象(応答変数 length) は説明変数によってどれくらい変化するか? の
127
     モデリング
128
     lm.model.0 <- lm(length ~ food + medicine, data = data.0)</pre>
129
     lm.model.0
130
131
     # ANOVA による検定結果
    # 説明は後程
132
133
     anova(lm.model.0)
134
135
     # より詳しい結果の表示→最初は飛ばす
136
     summary(lm.model.0)
137
138
     # 予測
139
    predict(
140
     lm.model.0,
141
     newdata = data.frame(food = 50, medicine = "medicine"))
142
143
     # 予測区間つき
144
    predict(
```

```
145
       lm.model.0,
146
       newdata = data.frame(food = 50, medicine = "medicine"),
147
       interval = "prediction", level = 0.95)
148
149
     # newdata の指定なし
150
     predict(lm.model.0)
151
152
     # データの型の確認
153
     class(data.0)
154
     # ベクトルデータ
155
156
     vec <- c(1, 2, 3, 4.5, 5, 6, 7.2, 8, 9, 9.9)
157
     vec
158
     int <- 1:10
     int
159
160
161
     # 等差数列
162
     seq(from = 0.1, to = 1, by = 0.1)
163
164
     # データフレームに格納する
165
     d <- data.frame(</pre>
166
        vec = vec
167
        int = int,
168
        seq = seq(from = 0.1, to = 1, by = 0.1)
169
     )
170
171
     # データの取り出し方
172
173
     d[1,]
174
     d[,1]
175
     d[2,3]
176
    names(d)
177
     d$vec
178
     d[,c("vec")]
179
     d[,c("vec", "seq")]
180
```

```
# 予測の図示
181
182
     # newdata の作成
183
     newfood \leftarrow seq(from = min(data.0$food), to = max(data.0$food), by = 1)
184
     newfood
185
     # 薬があるときのえさと体長の関係の予測のためのデータセット
186
187
     new.1 <- data.frame(</pre>
188
     food = newfood,
189
     medicine = "medicine")
190
     # 薬がない時のえさと体長の関係の予測のためのデータセット
191
192
     new.2 <- data.frame(</pre>
193
     food = newfood,
194
     medicine = "na")
195
     # 薬の有無別に予測
196
197
     pred.1 <- predict(</pre>
198
      lm.model.0, newdata = new.1,
199
      interval = "prediction", level = 0.95)
200
     pred.2 <- predict(</pre>
201
      lm.model.0, newdata = new.2,
202
      interval = "prediction", level = 0.95)
203
    # 図示
204
205
    # ちょっと複雑なプロット
206
     # これは以前作ったものと同じ。
207
     # コピペする方が早いし、正確。
     # 意地を張ると遅くなるうえに不正確になるので注意。ぜひコピペしてください。
208
209
     plot(data.0$length ~ data.0$food,
210
     col = c(2,3) [data.0\$medicine],
211
     pch=16,
212
      ylab = "Length",
213
      xlab = "food",
214
      main = "薬の有無別、体長と餌の量の関係",
215
      cex.main = 1.5,
216
     font.lab = 2
```

```
217
    )
    # 予測値
218
219
    lines(pred.1[,1] ~ newfood, col = 2, lwd = 2)
220
    lines(pred.1[,2] \sim newfood, col = 2, lwd = 1, lty = 2)
221
    lines(pred.1[,3] \sim newfood, col = 2, lwd = 1, lty = 2)
222
    lines(pred.2[,1] ~ newfood, col = 3, lwd = 2)
223
    lines(pred.2[,2] \sim newfood, col = 3, lwd = 1, lty = 2)
224
    lines(pred.2[,3] \sim newfood, col = 3, lwd = 1, lty = 2)
225
226
     # 凡例
227
    legend("topleft",
     legend = c("薬あり", "薬なし"),
228
229
     col = c(2,3),
230
     bty = "n",
231
     pch = 16)
232
233
234
     235
             写経終わり
236
     237
```

```
239
     #----
240
     # 統計の基本と t 検定
241
     #-----
242
     # 大阪データ
243
244
     osaka \leftarrow c(19,19,20,20,20,20,20,21,21,21)
245
     # 期待値
246
     19*(2/10) + 20*(5/10) + 21*(3/10)
247
     mean (osaka)
248
249
     # 標本分散
250
    ((19-20.1)^2)*(2/10) + ((20-20.1)^2) *(5/10) +
251
     ((21-20.1)^2) * (3/10)
252
253
     # 普遍分散
254
    length (osaka)
255
    ((19-20.1)^2)*(2/9) + ((20-20.1)^2) * (5/9) + ((21-20.1)^2) * (3/9)
256
    var(osaka)
257
258
     # 東京データ
259
260
    tokyo <- c(-50, 0, 0, 20, 20, 20, 20, 20, 70, 70)
261
262
     # 期待値
263
     mean(tokyo)
264
265
     # 標本分散
266
     (-50-19)^2 * (1/10) + (0-19)^2 * (2/10) +
267
     (20-19)^2 * (5/10) + (70-19)^2 * (2/10)
268
269
     # 不遍分散
270
    (-50-19)^2 * (1/9) + (0-19)^2 * (2/9) +
271
     (20-19)^2 * (5/9) + (70-19)^2 * (2/9)
272
     var(tokyo)
273
274
```

```
275
   #==========
276
    # t 検定
277
    #==============
278
279
    # ****
280
    # ★
             写経はじめ
281
    # ****
282
283
    # サンプルデータの作成
284
    d \leftarrow c(-1, -1, 0, 0, 1, 3, 5, 6, 7, 7)
285
    # 平均值
286
287
    mean(d)
288
289
    # 標準偏差
290
    sd(d)
291
292
    # サンプルサイズ
293
    length(d)
294
295
    # 標準誤差
296
    sd(d) / sqrt(length(d))
297
298
    # t 検定
299
    mean <- mean(d)</pre>
300
    std.error <- sd(d) / sqrt(length(d))</pre>
301
302
    t.value <- mean / std.error</pre>
303
    t.value
304
    (1-pt(t.value, df=length(d)-1))*2
305
306
    # Rに入っている関数を使う
307
    t.test(d)
308
309
```

```
311
    #========
312
    # 統計モデルな t 検定
313
    #=========
314
315
    # データの読み込みと表示
316
    # 作業ディレクトリを変更してから以下を実行
317
318
    # NOT RUN
319
    # setwd()
320
321
    data.0 <- read.csv("data0.csv")</pre>
322
    lm.model.0 <- lm(length ~ food + medicine, data = data.0)</pre>
323
    lm.model.0
324
325
326
    # より詳しい結果の表示
    # ここにt valueとt検定の結果が出ている。
327
328
    summary(lm.model.0)
329
330
   くくへんへんへんへんくんくくく #
331
332
    # ♪
            写経終わり
    333
334
```

```
336
    337
    # 分散分析 ANOVA
338
    339
340
    # グラフを描くだけ
341
    # サンプルデータの作成
342
343
    d2 <- data.frame(</pre>
344
    Y = c(c(1,2,3,4,5), c(4,5,6,7,8), c(7,8,9,10,11)),
345
     option = rep(c("A", "B", "C"), each=5)
346
    )
347
348
    d2
349
350
    # データの可視化
351
    plot(d2$Y ~ d2$option)
352
353
    # スライドに載せたような散布図形式のグラフを作る
    par(mar=c(5, 6, 3, 3))
354
355
    plot.default(
356
    d2\$Y \sim d2\$option,
357
     ylim=c(0,12), xlim=c(0.5,3.5),
     ylab="結果", xlab="選択肢",
358
359
    cex=2,
360
    cex.lab=2, cex.main=3,
361
    xaxt="n")
362
    axis(side=1, 1:3, LETTERS[1:3])
363
364
365
    # ****
366
    # ★
             写経はじめ
367
    # *****
368
369
    # サンプルデータの作成
370
    d2 <- data.frame(</pre>
371
    Y = c(c(1,2,3,4,5), c(4,5,6,7,8), c(7,8,9,10,11)),
```

```
372
    option = rep(c("A", "B", "C"), each=5)
373
374
375
    d2
376
377
    # データの可視化
378
    plot(d2$Y ~ d2$option)
379
380
    # Yの総平均
381
    mean(d2$Y)
382
    # option ごとの Y の期待値の算出
383
384
    tapply(d2$Y, d2$option, mean)
385
386
    # Rで分散分析を実装する
387
    lm.model.anova <- lm(Y ~ option, data=d2)</pre>
388
389
    # 係数の確認
390
    lm.model.anova
391
    summary(lm.model.anova)
392
393
    # 分散分析による予測
394
    predict(lm.model.anova, data.frame(option=c("A", "B", "C")))
395
396
    # 予測値は option 別の期待値に等しい
397
    tapply(d2$Y, d2$option, mean)
398
399
400
    401
    # 分散分析モデルにおける検定
402
    # 検定スライドを見せてから実施
403
    404
405
    # 予測値の変化の大きさ
406
    d2.effect <- data.frame(</pre>
407
     Y = c(rep(3,5), rep(6,5), rep(9,5)),
```

```
408
     option = rep(c("A", "B", "C"), each=5)
409
410
411
    d2.effect
412
     # 偏差平方和(分散の分子の部分)
413
414
     sum((d2.effect$Y-mean(d2.effect$Y))^2)
415
     # 偏差平方和を自由度で割る
416
417
     # 自由度 = 自由に動ける値
418
     # d2.effect の中身は3,6,9の3種類のみ。
     # また d2.effect の期待値は元データの期待値と等しくなるという制約がある
419
420
     # 自由に動ける値は 「3 - 1 = 2」
    # 予測値の差の大きさを表す分散
421
422
     sum((d2.effect$Y-mean(d2.effect$Y))^2) / 2
423
424
    effect <- sum((d2.effect$Y-mean(d2.effect$Y))^2) / 2</pre>
425
    effect
426
427
     # ノイズの大きさ
     # 単に実データから予測値を引いただけ
428
429
    # ↓これが予測誤差(ノイズ)
430
    d2$Y - d2.effect$Y
431
432
     # データフレームにまとめる
433
    d2.noise <- data.frame(</pre>
434
     Y = d2\$Y - d2.effect\$Y,
     option = rep(c("A", "B", "C"), each=5)
435
436
437
438
     d2.noise
439
     # 偏差平方和(分散の分子の部分)
440
     sum( (d2.noise$Y-mean(d2.noise$Y))^2 )
441
442
443
   # 自由度
```

```
# 誤差というモノの制約=合計値は O になる
444
445
    # そうしないと、単なるノイズ以上の意味を持ってしまうから。
    # ゆえに、option ごとに合計値が 0 にならなければならない。
446
    # サンプルサイズから option の個数 (この場合 3) を引く。
447
448
    length(d2.noise$Y)-3
449
    # ノイズの大きさを表す分散
450
    sum((d2.noise\$Y-mean(d2.noise\$Y))^2)/12
451
452
453
    noise <- sum( (d2.noise$Y-mean(d2.noise$Y))^2 ) / 12</pre>
454
    noise
455
    # F比
456
457
   # Effect は option の数-1 で割っている
458
    # Noise はサンプルサイズ-option 数で割っている
    # ゆえに、サンプルサイズが多くなるほど Effect と Noise の比は大きくなる。
459
460
    effect
461
    noise
462
463
    # F比
464
    F.ratio <- effect/noise
465
    F.ratio
466
467
    # 検定
468
    # option によって有意に予測は変わらないことがわかっているデータを集める部分は
469
    # 数式を使って省略する。
    # ここにも自由度が使われている。サンプルサイズが効いているということ。
470
471
    1 - pf(F.ratio, 2, 12)
472
473
474
    # 関数を使って計算
475
    lm.model.anova <- lm(Y ~ option, data=d2)</pre>
476
    anova(lm.model.anova)
477
    summary(lm.model.anova)
478
```

```
480
481
    # ♪ 写経終わり
    482
483
484
    485
    # ナイーブな予測 (NULL.model) との比較
    # ナイーブ予測と比べて予測誤差は減少したのか?
486
487
    488
489
    # NULL.model
490
    lm.model.NULL <- lm(Y ~ 1, data=d2)
491
    lm.model.NULL
492
493
    # ナイーブ予測の結果
494
    predict(lm.model.NULL)
495
496
    # ナイーブ予測残差の計算
    # 」これが予測誤差(ノイズ)
497
498
    d2$Y - 6
499
    # データフレームにまとめる
500
501
    d2.noise.NULL <- data.frame(</pre>
502
    Y = d2\$Y - 6,
503
    option = rep(c("A", "B", "C"), each=5)
504
    )
505
506
    d2.noise.NULL
507
508
509
    # ナイーブ予測の残差平方和
510
    sum( (d2.noise.NULL$Y - mean(d2.noise$Y))^2 )
511
512
    # F比
    # 予測残差の減少度 ÷ option 付き ANOVA 予測の予測残差の大きさ
513
514
    sum.sq.ANOVA <- sum( (d2.noise$Y - mean(d2.noise$Y))^2 )</pre>
515
    sum.sq.NULL <- sum( (d2.noise.NULL$Y - mean(d2.noise$Y))^2 )</pre>
```

```
516
517
    sum.sq.ANOVA
518
    sum.sq.NULL
519
520
    # 自由度の差
521
    3 - 1
522
523
     # 分散分析モデルの予測誤差の分散 (再掲)
524
    noise.ANOVA <- sum( (d2.noise$Y-mean(d2.noise$Y))^2 ) / 12</pre>
525
    noise.ANOVA
526
    # 見やすいように括弧を増やしている。本来は(sum.sq.NULL - sum.sq.ANOVA )/ 2
527
528
    / Noise.ANOVA
    # 残差の差分を自由度の差分で割ったものが分子
529
530
     # option 入りの ANOVA モデルの予測残差の分散が分母
531
    F.ratio2 <- ((sum.sq.NULL - sum.sq.ANOVA)/2) / noise.ANOVA
532
    F.ratio2
533
534
    1 - pf(F.ratio2, 2, 12)
535
536
537
     # ナイーブ予測と分散分析モデル予測との予測誤差の比較
538
    anova(lm.model.NULL, lm.model.anova)
539
540
     # 普通の ANOVA
541
    anova(lm.model.anova)
542
     # 分散分析とは予測残差の違いを検定するものだ、とご理解ください
543
544
545
546
547
548
549
```

```
550
     #========
551
     # 回帰分析
     #========
552
553
554
     # ****
555
              写経はじめ
     # *****
556
557
    # サンプルデータ
558
559
    d3 <- data.frame(</pre>
560
     Y = c(3, 5, 4, 6, 7),
561
     X = 1:5
562
     )
563
564
     d3
565
    nrow(d3)
566
    plot(d3\$Y \sim d3\$X)
567
568
     summary(lm(Y \sim X, data=d3))
569
     lm(Y \sim X, data=d3)
570
     anova(lm(Y \sim X, data=d3))
571
572
573
     # optim で最小二乗法
574
575
    # 自作関数の作成
576
    # 予測するための関数
577
    yosoku <- function(a, b, X) {</pre>
    Yhat <- a*X + b
578
579
     return(Yhat)
580
    }
581
582
     # 2*3 + 1
583
     yosoku(a=2, b=1, X=3)
584
585
```

```
# 予測結果の残差平方を出す関数
586
587
     zansa <- function(Y, Yhat){</pre>
588
      zansa <- (Y - Yhat)^2</pre>
589
     return(zansa)
590
     }
591
592
     # (2-3)^2
593
     zansa(2, 3)
594
595
596
    # 残差平方和計算
    # for ループを使用する(本当は使わなくていい。勉強用)
597
598
     zansa2 <- numeric()</pre>
599
     a <- 2
600
     b <- 2
601
     for ( i in 1:nrow(d3)) {
602
      Yhat <- yosoku(a, b, d3$X[i])</pre>
603
      zansa2[i] <- zansa(d3$Y[i], Yhat)</pre>
604
     zansa2
605
606
     sum.zansa2 <- sum(zansa2)</pre>
607
     sum.zansa2
608
609
     plot(d3$Y ~ d3$X,ylim=c(0, 15),main=c("a=2, b=2の時の予測結果"))
610
     abline (a=2, b=2)
611
612
     # 一つの関数にまとめる
613
614
    OLS <- function(para){
615
      zansa2 <- numeric()</pre>
616
      for ( i in 1:nrow(d3)) {
617
        Yhat <- yosoku(para[1], para[2], d3$X[i])</pre>
618
        zansa2[i] <- zansa(d3$Y[i], Yhat)</pre>
619
620
       sum.zansa2 <- sum(zansa2)</pre>
621
      return(sum.zansa2)
```

```
622
   }
623
624
    OLS(c(2, 2))
625
626
     # optim で最適化
627
    optim(c(2, 2), OLS)
628
629
    # 答え合わせ
630
631
     lm.model <- lm(Y ~ X, data=d3)
632
    lm.model
633
    # 検定
634
635
    anova(lm.model)
636
637
638
    639
    # 実はこの関数を使った方が早い
640
    # これを写経する必要はありません
641
     # しかし、実務上はこちらのプログラムを使った方が計算速度が速くなります。
642
643
    OLS2 <- function(para){
644
     Yhat <- yosoku(para[1], para[2], d3$X)</pre>
645
     zansa2 <- zansa(d3$Y, Yhat)</pre>
646
     sum.zansa2 <- sum(zansa2)</pre>
647
     return(sum.zansa2)
648
    }
649
650
     OLS2(c(2, 2))
651
652
    optim(c(2, 2), OLS2)
653
654
    # 関数の仕組み
655
656
    bai <- function(x) {</pre>
657
     return(2*x)
```

```
658
   }
659
660
    # 実はこの書き方でも OK
661
    # bai <- function(x) return(2*x)</pre>
662
663
    bai(2)
664
    sample.data <- 1:10</pre>
665
    bai(sample.data)
666
    # このやり方はエラー
667
668
    bai(1,2,3,4,5,6,7,8,9,10)
669
670
    ###################################
671
672
    673
             写経終わり
    674
675
676
677
678
    # 回帰分析における分散分析
679
    #-----
680
681
    # 係数
682
    lm.model <- lm(Y ~ X, data=d3)
683
    lm.model
684
685
    lm.model$coef
686
687
    # 予測値の計算
688
    yosokuti <- yosoku(lm.model$coef[2], lm.model$coef[1], d3$X)
689
    yosokuti
690
691
    # 予測値の変動の大きさ=xがyに与える影響の大きさの計算
692
    heihou <- zansa(mean(yosokuti), yosokuti)</pre>
693
    heihou
```

```
694
695
     Effect <- sum(heihou) / 1</pre>
696
     Effect
697
698
     # 予測残差の大きさ = ノイズの大きさ
699
     gosa <- zansa(d3$Y, yosokuti)</pre>
700
     gosa
701
702
     Error <- sum(gosa) / 3</pre>
703
     Error
704
     # 分散分析
705
706
     F.value <-Effect / Error
707
     F.value
708
709
     1 - pf(F.value, 1, 3)
710
711
     anova(lm.model)
712
713
714
     # おまけ NULL モデル (ナイーブ予測) との比較
715
     lm.model.NULL <- lm(Y ~ 1, data=d3)
716
     predict(lm.model.NULL)
717
     mean(d3$Y)
718
719
     anova(lm.model.NULL, lm.model)
720
     anova(lm.model)
721
722
```

```
724
    725
    # パラメトリックブートストラップ検定
726
    727
728
    # ****
729
             写経はじめ
    # *****
730
731
732
    # 回帰分析の時に使ったデータの再掲
733
    d3 <- data.frame(</pre>
734
    Y = c(3, 5, 4, 6, 7),
735
    X = 1:5
736
    )
737
738
    # モデルの作成
739
    lm.model.NULL <- lm(Y ~ 1, data=d3)
740
    predict(lm.model.NULL)
741
742
    lm.model <- lm(Y \sim X, data=d3)
743
    anova(lm.model)
744
745
    # シミュレーションデータの作成
746
    # NULL モデルが正しいと仮定してシミュレーションする
747
748
749
    simulate(lm.model.NULL, 1)
750
751
    sim.data <- cbind(</pre>
752
     simulate(lm.model.NULL,1),
753
     1:5)
754
755
    colnames(sim.data) <- c("Y","X")</pre>
756
    sim.data
757
    mean(sim.data$Y)
758
    plot(sim.data$Y ~ sim.data$X)
759
```

```
760
761
762
     # パラメトリックブートストラップ検定
763
     # 本来は N.sim=10000 くらいで行うことが望ましいが
     # PC のスペックに自信のない方は N.sim=1000 に変えてください
764
     # apply 系の関数を使うともっと早くできます。興味がある方は試してみてください。
765
766
767
     set.seed(1)
768
     Nsim <- 10000
769
     sim <- simulate(lm.model.NULL, Nsim)</pre>
770
     sim.F.value <- numeric()</pre>
771
     for(i in 1:Nsim){
772
     sim.data <- cbind(sim[i], 1:5)</pre>
773
     colnames(sim.data) <- c("Y", "X")</pre>
774
      model <- lm(Y ~ X, data=sim.data)</pre>
775
     sim.F.value[i] <- summary(model)$fstatistic[1]</pre>
776
    }
777
778
     sim.F.value
779
     max(sim.F.value)
780
781
     # 見やすいヒストグラムを作る
782
     F.value <- subset(sim.F.value, sim.F.value < 20)</pre>
783
784
     # F比のヒストグラム
785
     hist(F.value, xlim=c(0, 20))
786
     # 0~1 の範囲内にある F 比は 10000 個中 6000 個くらい
787
     # 1~2 の範囲内にある F 比は 10000 個中 1400 個くらい
788
789
     # 2~3 の範囲内にある F 比は 10000 個中 650 個くらい
790
     # ヒストグラムの数字バージョン
791
792
     stem(F.value)
793
794
     # 実際の F 値と比較
795
     # この F 比よりも大きな F 比が出た回数を数えればよい
```

```
796
     lm.model <- lm(Y \sim X, data=d3)
797
     lm.model
798
     summary(lm.model)
799
     summary(lm.model)$fstatistic[1]
800
801
    # 数え方
    # 論理演算
802
803
    2<3
804
     2>3
805
806
     # 相手がたくさんあっても大丈夫
807
     c(1,2,3,4,5,6,7,8,9,10)>6
808
809
    # TRUE の個数がわかる
810
     sum(c(1,2,3,4,5,6,7,8,9,10)>6)
811
812
813
     # 実データから計算された F比
814
     summary(lm.model)$fstatistic[1]
815
816
     # PB 検定の p 値
817
     sum(sim.F.value>=summary(lm.model)$fstatistic[1]) / Nsim
818
819
     # 理論的に計算された p 値とほぼ等しい事を確認
820
     anova(lm.model)
821
822
     # 偉人の数式とシミュレーションの比較
823
824
    # pf はF比がある値を下回る確率を表す
825
     # dfはF比の確率分布を表す
826
     # 確率分布とは、面積が1になるように標準化されたヒストグラムのこと
827
    x < - seq(0, 20, by=0.1)
     F < -df(x, 1, 3)
828
829
830
831
     hist(F.value,xlim=c(0,20),ylim=c(0,1),prob=T,breaks=50)
```

```
832
    lines(x, F, col=2, lwd=2)
833
834
    835
             写経終わり
836
     837
838
     839
     # スライドのグラフを作るためのスクリプト
840
841
842
    # 正規分布
843
    x < - seq(-10, 10, by=0.01)
844
    N1 <- dnorm(x, mean=0, sd=1)
    N2 \leftarrow dnorm(x, mean=2, sd=1)
845
846
    N3 \leftarrow dnorm(x, mean=0, sd=2)
847
848
    plot(0,0, type="n", xlim=c(-7,7), ylim=c(0,0.5),
     xlab="",ylab="確率",main="正規分布",cex.main=2)
849
850
    lines(N1 \sim x, col=1, lwd=4)
851
    lines (N2 \sim x, col=2, lwd=4)
852
    lines (N3\simx, col=3, lwd=4)
853
    abline(v=0)
854
    legend(
855
     "topleft",
856
     lwd=4,
857
     col=c(1,2,3),
858
     btv="n",
     legend=c("平均0、分散1","平均2、分散1","平均0、分散4"),
859
860
     cex=1.5
861
    )
862
863
864
865
866
867
```

```
868
    869
    # 正規線形モデルの再確認
870
    871
872
    # ****
            写経はじめ
873
    # *****
874
875
876
    # データの読み込み
877
    # setwd()
878
    data0 <- read.csv("data0.csv")</pre>
879
    head(data0)
880
    # 分散分析・回帰分析で解説したように、両者はやってることがほぼ同じ
881
882
    # 説明変数がカテゴリデータか定量データかということは気にする必要がない
883
    summary(data0)
884
    pairs(data0)
885
   # モデルの作成
886
887
    # 応答変数:length
888
    # 説明変数:food & medicine
    # 興味のある対象(応答変数 length) は説明変数によってどれくらい変化するか?
889
890
    model0 <- lm(length ~ food + medicine, data = data0)</pre>
891
    model0
892
893
    # 分散分析結果の再確認
    # すべての出力の意味が分かるはず
894
    # 変数を加えることによって予測残差は"有意に"減ったか?
895
896
    anova(model0)
897
898
    # 要約結果の再確認
899
    summary(model0)
900
901
   # Call:モデルの式
902
    # Residuals:残差
```

Coefficients:係数。t検定によるパラメタの有意性の検定結果

```
# →パラメタは有意に 0 と異なるか?
904
905
    # Residual standard error 残差の標準誤差
906
    # Multiple R-squared:
    # 予測の変動の大きさ/(予測の変動の大きさ+残差の変動の大きさ)
907
908
    # →モデルの当てはまりの良さの指標
909
    # Adjusted R squared:自由度調整済みR2。
    # →変数が多いと R2 が増えて当たり前。これを補正したもの。
910
911
    # F-statistic ナイーブ予測と比べて予測残差はどれだけ減ったか?
912
913
914
   # 🎝
915
            写経終わり
    くくへんへんへんくんくんくんくん #
916
917
918
    #-----
919
    # シミュレーションと予測区間
920
    #-----
921
    set.seed(1)
922
    x.sim <- rnorm(mean=5, sd=10, n=100)
923
    x.sim
924
    hist(x.sim)
925
    y.sim <- x.sim * 0.8 + 5 + rnorm(mean=0, sd=10, n=100)
926
927
    plot(y.sim ~ x.sim)
928
929
    model.sim <- lm(y.sim ~ x.sim)</pre>
930
    summary(model.sim)
931
932
    # 予測
933
    new.sim <- data.frame(x.sim=seq(min(x.sim), max(x.sim), by=0.1))</pre>
934
    new.sim
935
936
    # 信頼区間
937
    # 同じ確率分布から得られたデータを例えば100データセット集める。
938
    # 一つのデータセットにつき 100 サンプルあるから、データは1万個
939
    # 全部回帰分析で線を引っ張る
```

```
# 引っ張られた 100 本の線のうち、95 本の線が中に入る区間
940
941
     sinrai <- predict(</pre>
942
      model.sim, newdata=new.sim, se.fit=T, interval="confidence")
943
     sinrai
944
945
     plot(y.sim ~ x.sim)
946
     lines(sinrai$fit[,1] ~ new.sim$x.sim, lwd=2)
947
     lines(sinrai$fit[,2] ~ new.sim$x.sim, lwd=2, col=4)
948
     lines(sinrai\fit[,3] ~ new.sim\fix.sim, lwd=2, col=2)
949
950
     # 予測区間
951
     # 線を引く
     # そのあと、同じ確率分布から得られたデータを例えば100個集める。
952
     # その 100 個のデータのうち、95 このデータはこの範囲内に収まる
953
954
     # 予測値から左右対称になっていることに注意
955
956
     yosoku <- predict(</pre>
957
      model.sim, newdata=new.sim, se.fit=T, interval="predict")
958
     yosoku
959
     yosoku$residual.scale
960
     # ↑この値が Ysim のノイズの標準偏差 (10) に近いことを確認
961
962
     plot(y.sim ~ x.sim)
963
     lines(yosoku$fit[,1] ~ new.sim$x.sim, lwd=2)
964
     lines(yosoku$fit[,2] ~ new.sim$x.sim, lwd=2, col=4)
965
     lines(yosoku$fit[,3] ~ new.sim$x.sim, lwd=2, col=2)
966
967
968
     # 両方載せたグラフ
969
     plot(y.sim ~ x.sim)
970
     lines(sinrai\fit[,1] ~ new.sim\fix.sim, lwd=2)
971
     lines(sinrai\fit[,2] ~ new.sim\fix.sim, lwd=2, col=4)
972
     lines(sinrai\fit[,3] ~ new.sim\forall x.sim, lwd=2, col=2)
     lines(yosoku$fit[,2] ~ new.sim$x.sim, lwd=2, col=4, lty=2)
973
     lines(yosoku$fit[,3] ~ new.sim$x.sim, lwd=2, col=2, lty=2)
974
975
```

```
976
977
      # 中心極限定理
978
979
      980
981
      # サイコロの合計値は正規分布になるか?
982
      N.sample <- 1
983
      saikoro <- sample(1:6, size=N.sample, replace = T)</pre>
984
      saikoro
985
      # 一回サイコロを投げることを 1000 回実行
986
987
      N.sim < -1000
988
     kekka <- numeric()</pre>
989
      for(i in 1:N.sim){
990
      kekka[i] <- sample(1:6, size=N.sample, replace = T)</pre>
991
      }
992
      kekka
993
      hist(kekka, breaks=0:6)
994
995
      # 五回サイコロを投げることを 1000 回実行
996
997
     N.sample <- 5
998
     N.sim < -1000
999
     kekka <- numeric()</pre>
1000
      for(i in 1:N.sim){
1001
      kekka[i] <- sum(sample(1:6, size=N.sample, replace = T))</pre>
1002
      }
1003
      kekka
1004
      hist(kekka, breaks=min(kekka):max(kekka))
1005
1006
      # 十回サイコロを投げることを 1000 回実行
1007
1008
      N.sample <- 10
1009
     N.sim < -1000
1010
     kekka <- numeric()</pre>
1011
     for(i in 1:N.sim){
```

```
1012
       kekka[i] <- sum(sample(1:6, size=N.sample, replace = T))</pre>
1013
1014
      kekka
1015
      hist(kekka, breaks=min(kekka):max(kekka))
1016
1017
1018
      # 十回サイコロを投げることを 10000 回実行
1019
      N.sample <- 10
1020
      N.sim <- 10000
1021
      kekka <- numeric()</pre>
1022
      for(i in 1:N.sim){
1023
      kekka[i] <- sum(sample(1:6, size=N.sample, replace = T))</pre>
1024
      }
1025
      kekka
1026
      hist(kekka, breaks=min(kekka):max(kekka))
1027
      # 単体では全く正規分布ぽくないサイコロの目が
1028
      # 合計値をとると正規分布によく似た確率分布(ヒストグラム)になっている。
1029
1030
1031
1032
```

```
1034
1035
     #----
     # モデル選択と AIC
1036
1037
1038
     # ****
1039
              写経はじめ
1040
     # *****
1041
1042
1043
     # データの読み込み
     # ディレクトリを変更した後で実行する
1044
     data1 <- read.csv("data1.csv")</pre>
1045
1046
     # データの確認
1047
1048
     head(data1)
1049
     names (data1)
1050
     levels(data1$option1)
1051
     levels(data1$option2)
1052
     summary(data1)
1053
1054
     pairs(data1, panel=panel.smooth)
1055
1056
1057
     # 4 つの説明変数を全て加えたモデルを作成
1058
     lm.model.1 <- lm(Y ~., data=data1)
1059
     summary(lm.model.1)
1060
     anova(lm.model.1)
1061
     # option2 が不要なようなので、これを削除したモデルを作成
1062
1063
     lm.model.2 <- update(lm.model.1, ~ .-option2)</pre>
1064
     lm.model.2
1065
     # 変数を切った時、予測誤差は有意に増加したか?
1066
1067
     anova(lm.model.1, lm.model.2)
1068
     # option2 を切ったモデルで再度分散分析
```

```
1070
      anova(lm.model.2)
1071
1072
1073
      # x1 もいらなさそうだったので削除する
1074
      lm.model.3 <- update(lm.model.2, ~ .-x1)</pre>
1075
1076
      # 変数を切った時、予測誤差は有意に増加したか?
      anova(lm.model.2, lm.model.3)
1077
1078
1079
      # 二つの変数を削除した後のモデルでもう一度分散分析
1080
      anova(lm.model.3)
1081
1082
1083
      # 念のためx2も削除してみる
1084
      lm.model.4 <- update(lm.model.3, ~ .-x2)</pre>
1085
      # x2 という変数がなくなった時、予測残差は有意に増えたか?
1086
1087
      anova(lm.model.3, lm.model.4)
1088
1089
      # Best なモデルは lm.model.3 に決定
1090
      best.model.anova <- lm.model.3</pre>
1091
      summary(best.model.anova)
1092
1093
      # ベストじゃない変数全部入りモデルとの比較
1094
      summary(lm.model.1)
1095
1096
      # AICによるモデル選択
1097
1098
      # install.packages("MuMIn") をした後で実行
1099
      library(MuMIn)
1100
      # 変数を抜き差しして、すべての変数の組み合わせにおいて AIC を算出して比較する
1101
      list <- dredge(lm.model.1, rank="AIC")</pre>
1102
1103
      list
1104
1105
      all.model <- get.models(list)</pre>
```

```
1106
    best.model.AIC <- all.model[1]</pre>
1107
    # AICと検定の比較
1108
   # 両方同じ結果になった
1109
    best.model.AIC
1110
1111
    best.model.anova
1112
1113
   へんへんくんくんくんくんくんく #
1114
   # ♪ 写経終わり ♪
1115
    1116
1117
```

```
1119
     #===========
1120
     # 8章 Type II ANOVA
1121
     #===============
1122
1123
     # *****
              写経はじめ
1124
     # *****
1125
1126
1127
1128
     # データの読み込み
1129
     # ディレクトリを変更した後で実行する
1130
     data2 <- read.csv("data2.csv")</pre>
1131
     # データの確認
1132
1133
     # 売上 (sell) を上げるためにはどうすればよいかを知りたい、という設定
     # experience:チラシを配る店員さんの経験年数
1134
1135
     # n.sheets:配れたチラシの枚数
     # time:チラシを配った時間帯(昼か夜か)
1136
     # sex:チラシを配った店員さんの性別
1137
1138
     head(data2)
1139
     pairs (data2, panel=panel.smooth)
1140
     summary(data2)
1141
1142
     # やってはいけない解析方法
1143
     # 統計モデルを使わずに、一つ一つ検定していく
1144
     # 経験・チラシ配布枚数は売り上げに貢献しているか? @t検定
1145
1146
     summary(lm(sell ~ experience, data=data2))
1147
     summary(lm(sell ~ n.sheets, data=data2))
1148
1149
     #経験・チラシ配布枚数は売り上げに貢献しているか? @分散分析
1150
     anova(lm(sell ~ experience, data=data2))
     anova(lm(sell ~ n.sheets, data=data2))
1151
1152
     # チラシを配った人の性別と、配った時間は売り上げに貢献しているか? @t 検定
1153
```

t.test(data2\$sell ~ data2\$sex)

```
1155
     t.test(data2$sell ~ data2$time)
1156
1157
      # 4つの変数を一つにまとめて統計モデリングする
1158
1159
      sell.model1 <- lm(sell ~., data = data2)</pre>
1160
      # 係数が変数を一つずつ入れた時と大きく異なっていることに注意
1161
      # 特に experience の値がおかしい
1162
1163
      sell.model1$coef
1164
      # 分散分析する。
1165
1166
      anova(sell.model1)
1167
      # 各係数を t 検定する
1168
1169
      summary(sell.model1)
1170
      # 説明変数の関係性
1171
1172
     par(mfrow=c(1,2))
1173
     plot(data2$n.sheets ~ data2$experience)
1174
     plot(data2$sex ~ data2$time)
1175
     par(mfrow=c(1,1))
1176
1177
```

```
1179
      #=========
1180
      # Type II ANOVA
1181
      #==========
1182
1183
      # install.packages("car") をしたあとに実行
1184
      library(car)
1185
1186
      # 普通の分散分析 (Type I ANOVA)
1187
      anova(sell.model1)
1188
1189
      # Type II ANOVA
      Anova(sell.model1,type=c("II"))
1190
1191
1192
      # 性別はいらなさそうだったので切る
1193
      sell.model2 <- update(sell.model1, ~.-sex)</pre>
1194
      anova(sell.model1, sell.model2)
1195
1196
      Anova (sell.model2)
1197
1198
      # さらに経験もいらなさそうだったので切る
1199
      sell.model3 <- update(sell.model2, ~.-experience)</pre>
1200
      anova(sell.model2, sell.model3)
1201
1202
      Anova(sell.model3)
1203
      # これがベストモデル
1204
1205
      summary(sell.model3)
1206
1207
1208
      # AIC
1209
1210
      library(MuMIn)
1211
      sell.model.list <- dredge(sell.model1, rank="AIC")</pre>
1212
      sell.model.list
1213
1214
      all.model.sell <- get.models(sell.model.list)</pre>
```

```
1224
     1225
     # 交互作用
1226
     1227
1228
     # *****
              写経はじめ
1229
     # *****
1230
1231
1232
1233
     # データの読み込み
1234
     # ディレクトリを変更した後で実行する
1235
     data3 <- read.csv("data3.csv")</pre>
1236
1237
1238
     # データの確認
1239
     head (data3)
1240
     summary(data3)
1241
     pairs(data3, panel=panel.smooth)
1242
1243
     # 気温や天気がビールの利益に与える影響は?
     # beer:ビールによる利益
1244
1245
     # temperature:気温
1246
     # weather:天気
1247
1248
     # ダメな分析手法、変数を別個に入れて検定
1249
     summary(lm(beer ~ temperature, data=data3))
1250
     t.test(data3$beer ~ data3$weather)
1251
     # 変数を一緒にしてモデリング
1252
1253
     model.beer0 <- lm(beer~., data=data3)</pre>
1254
     library(car)
1255
     Anova (model.beer0)
1256
1257
     # 晴れの日、雨の日でデータを分割
1258
     fine <- subset(data3, weather=="fine")</pre>
1259
     rain <- subset(data3, weather=="rain")</pre>
```

```
1260
1261
      # お天気別気温の影響
1262
     par(mfrow=c(1,2))
1263
     plot(fine$beer~fine$temperature, main="晴れ")
1264
     plot(rain$beer~rain$temperature, main="雨")
1265
     par(mfrow=c(1,1))
1266
1267
      # 交互作用を入れる
1268
      # model \mathcal{O} formula
      # どれを使っても、今回は、結果は同じ
1269
      # + で区切られたのが主効果(交互作用じゃないもの)
1270
1271
      #:でつながったものが交互作用
1272
     lm(beer ~ temperature + weather + temperature:weather, data=data3)
1273
     lm(beer \sim temperature * weather , data = data3)
     lm(beer \sim (.)^2, data = data3)
1274
1275
1276
      # 交互作用入りのモデルを使って再度検定
1277
     model.beer <- lm(beer ~ (.)^2, data = data3)
1278
     summary(model.beer)
1279
1280
      # anova
1281
     anova(model.beer)
1282
1283
      # Type II ANOVA
1284
     Anova(model.beer, type="II")
1285
1286
      # weather の部分が Type I ANOVA と全くいっしょ
      # 主効果の検定をする際に、交互作用を無視してしまっている。
1287
1288
1289
      # Type III ANOVA
1290
      # 交互作用が入っている時はこっちを使った方がよいかも
1291
      # 交互作用に着目する場合は Type III ANOVA を使うべき
1292
      # 主効果の影響を主に見ているのならば、Type II ANOVA のままでよいという説もある。
     Anova(model.beer, type="III")
1293
1294
1295
    # 3つの ANOVA の比較
```

```
1296
      anova(model.beer)
1297
      Anova(model.beer, type="II")
1298
      Anova(model.beer, type="III")
1299
1300
      # 交互作用入りモデルの係数
1301
      model.beer$coef
1302
1303
      # AIC でモデル選択
1304
      library(MuMIn)
1305
      dredge(model.beer, rank="AIC")
1306
1307
      # 結果の図示
1308
1309
      plot(data3$beer ~ data3$temperature,
1310
       col=c(1,2)[data3$weather],
1311
       pch = 16,
1312
       xlab = "気温",
       ylab = "ビールによる利益",
1313
       main = "お天気別ビールによる利益",
1314
1315
       font.lab = 2
1316
1317
      legend("bottomright", pch = 16, col = c(1,2), legend = c("fine","rain"))
1318
1319
      # 回帰直線を引く
1320
      # 各係数の使い方に注意
1321
      abline(
1322
       model.beer$coef[1],
1323
       model.beer$coef[2],
1324
       lwd = 2)
1325
      abline(
1326
       model.beer$coef[1] + model.beer$coef[3],
1327
        model.beer$coef[2] + model.beer$coef[4],
        lwd = 2, col = 2)
1328
1329
1330
      # 晴れの日の傾きと切片
1331
      model.beer$coef[1]
```

```
1332
     model.beer$coef[2]
1333
      # 雨の日の傾きと切片
1334
1335
     model.beer$coef[1] + model.beer$coef[3]
1336
     model.beer$coef[2] + model.beer$coef[4]
1337
1338
     1339
     # Þ
               写経終わり
1340
1341
     1342
1343
      #==========
      # おまけ
1344
      # 両方カテゴリデータの時
1345
1346
      #==========
1347
1348
      data <- data.frame(Y = c(0,5,3,10,1,6,2,13),
1349
       sisaku A = rep(c("a.not", "act", "a.not", "act"), 2),
1350
       sisaku B = rep(c("a.not", "a.not", "act", "act"), 2))
1351
1352
     data
1353
1354
      model <-lm(Y \sim (.)^2, data=data)
1355
      model$coef
1356
1357
      model.2 <- lm(Y ~ sisaku A, data=data, subset=(sisaku B == "a.not"))</pre>
1358
     model.2$coef
1359
1360
     model.3 <- lm(Y ~ sisaku A, data=data, subset=(sisaku B == "act"))</pre>
1361
     model.3$coef
1362
1363
1364
     model$coef
1365
     model.2$coef
     model.3$coef
1366
1367
```

```
# subset の意味
1368
1369
      data
1370
      subset(data, data$sisaku B == "a.not")
1371
      subset(data, data$sisaku B == "act")
1372
1373
      ################
      # もっとわかりやすい(が、ほんとはダメな)例
1374
1375
1376
      data <- data.frame(Y = c(1, 5, 3, 10),
1377
        sisaku A = rep(c("a.not", "act", "a.not", "act"), 1),
1378
        sisaku B = rep(c("a.not", "a.not", "act", "act"), 1))
1379
1380
      data
1381
1382
      model <-lm(Y \sim (.)^2, data=data)
1383
      model$coef
1384
1385
      model.2 <- lm(Y ~ sisaku A, data=data, subset=(sisaku B == "a.not"))</pre>
1386
      model.2$coef
1387
1388
      model.3 <- lm(Y ~ sisaku A, data=data, subset=(sisaku B == "act"))</pre>
1389
      model.3$coef
1390
1391
1392
      model$coef
1393
      model.2$coef
      model.3$coef
1394
1395
1396
      subset(data, data$sisaku B == "a.not")
1397
      subset(data, data$sisaku B == "act")
1398
      # 本当はこんなデータ使ってはいけない。あくまで、わかりやすさを優先した解説
1399
1400
      anova(model)
1401
1402
      # お疲れ様でした
```