なぜ R?

Toshiki SHIBANO

2021-02-11

目次

データ分析とは

私はデータ分析は大きく分けて次の 6 つから成り立っていると考えています.

1. 実験計画
2. データの取得
3. データの整理
4. データの可視化
5. 統計処理 (検定，モデリング)
6. 結果の吟味，考察，提案

1 と 6 に関しては解析ツールがほぼ必要なく，自分の頭とペンで行えます (実験計画における乱数などの例外はありますが).

しかしながら 2 ~ 4 に関しては，何かしらのツールが必須となっています．今の時代，グラフを手書きで描くなんてことはほぼ無いでしょう．そのツールとして 1 番に思い浮かぶのが，Excel (Microsoft) だと思います．有料ではありますが，非常に素晴らしいツールで世界中で使用されています．マウスでグラフを簡単に描くことができ，さらにピボットテーブルによる強力な集計も可能です．それ以外のツールとして挙げられるのが，SPSS (IBM) や JMP (SAS) といった非常に強力な有料ツール，プログラミング言語になりますが，R や Python なども有名だと思います．

R 言語とは

Wikipedia (https://ja.wikipedia.org/wiki/R) をまとめますと

- 日本語対応
- 様々な OS に対応したオープンソース・**フリー**ソフトウェア
- 作図・統計解析に強い
- 世界中の様々な研究者・データ分析者が利用
- 他人が開発したパッケージ (便利なツール) を DL して使用可能であり，また自分が作って他人に配ることも可能

なぜ R を使うのか

- 無料だから
- バージョンさえ合わせれば，世界中の誰がしても同じ結果が得られるから (再現性)
- 自分で思うがままに高度なこと (データの整理，作図，統計解析) が出来るから
- 世界中の先駆者が便利な機能 (パッケージ) を開発していて，それを利用できるから
- Python と比較して使い始めるまでが簡単 (と思います)

R Markdown のススメ

R はスクリプトファイル (メモ帳みたいなもの) に書いて実行することが多いと思います．しかしながら，非常に便利な R Markdown というものがあります．私も最近使い始めました (これも R Markdown で書いてます)．なぜ便利なのか，一言でまとめるなら

(実験計画 → データの取得 →) データの処理 → 可視化 → 統計解析 → 考察を一つのファイルで行える

に尽きると思います．Word などと行ったり来たりする必要がありませんし，解析したコードもそのまま載せることができます．パワーポイントもそのまま出力できるとか･･･? 初めは使い慣れるのに時間がかかるかもしれませんが，ぜひマスターしていただきたいです．

解析例

R で解析例を載せます．データは架空です．

内容ジャガイモ 3 品種 (A, B, C) の収量の差を調べたい．そこで研究室で管理している圃場を用いて，乱塊法 (4 反復) で試験した．1 試験区あたりの面積は $10m^2$ である．その収量の結果を yield.csv にまとめた．

注意点: R でデータを読み込ませて扱う場合，tidy データが必要である場合が多いです．tidy データとは

- 各変数が独立したデータであること
- 観測した値は 1 行に記録される
- 観測データの集合はテーブルを表現する

わかりにくいと思いますが，すぐに慣れると思います

実際に行っていきます．

```
# ディレクトリの設定が必要なら行う
# getwd()
# setwd()
# もしくは Session → Set Worling Directory → Choose Directory

# 試験区設定
set.seed(seed = 0) # 乱数の固定
# それぞれのブロックで A, B, C をランダムに配置する
b1 <- sample(x = c("A", "B", "C"), size = 3, replace = FALSE)
# 与えた条件 (x 引数) の中から全てサンプリングするなら size 引数を与えなくて良い
b2 <- sample(c("A", "B", "C"), replace = FALSE)
b3 <- sample(c("A", "B", "C"), replace = FALSE)
b4 <- sample(c("A", "B", "C"), replace = FALSE)

# 行列をフィールドに見立てる
test_field <- matrix(c(b1, b2, b3, b4), ncol = 4)
colnames(test_field) <- c("b1", "b2", "b3", "b4" ) # 列の名前
rownames(test_field) <- c("", "", "") # 行の名前

# test_field
knitr::kable(test_field, format = "pandoc")
```

b1	b2	b3	b4
B	C	A	B
A	A	C	C
C	B	B	A

```
# データの読み込み
# OS によってコードが変わる
yield <- read.csv(file = "yield.csv", header = TRUE)

# データの表示
yield
```

variety	block	yield
A	1	312.6295
A	2	296.7377
A	3	313.2980
A	4	312.7243
B	1	264.1464
B	2	244.6005
B	3	250.7143
B	4	257.0528
C	1	249.9423
C	2	274.0465
C	3	257.6359
C	4	242.0099

```
# variety 列と block 列を character 型から factor 型に変更する
# 後の TukeyHSD を使った多重比較のため
yield$variety <- factor(yield$variety)
yield$block <- factor(yield$block)
```

```
# 作図
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
```

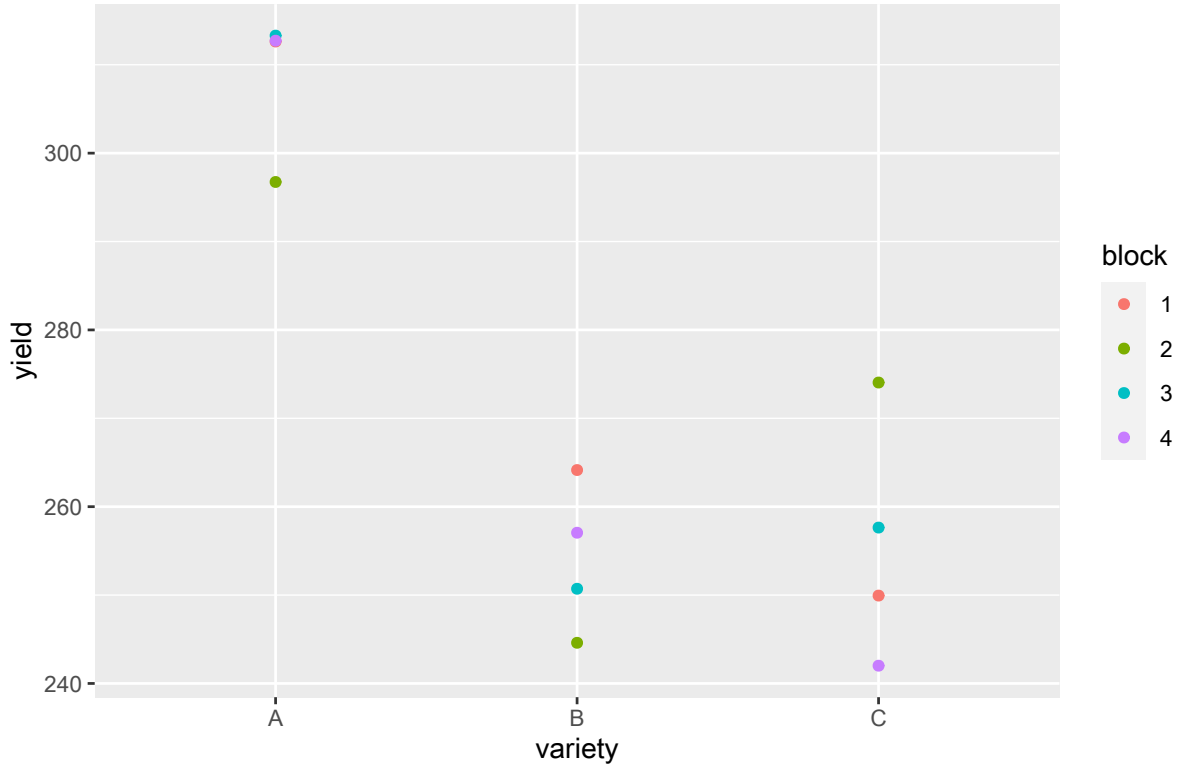
```
## v tibble  3.0.1     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------------------------------- tid
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# 日本語フォントを表示出来るように設定
theme_set(theme_gray(base_family = "HiraMinProN-W3"))
# ポストフォントスクリプトのデータベースにフォントファミリ"HiraMinProN-W3"が見つかりません
# という警告が出てるけど，いけてるからヨシ！
# 原因が分からないので調べておきます.

# x軸に品種，y軸に収量
p <- ggplot(data = yield,
            mapping = aes(x = variety, y = yield))
p1 <- p + geom_point(mapping = aes(color = block)) +
    labs(title = "Fig.1 各品種の収量")
p1
```

Fig.1 各品種の収量




```
# データの集計
# 品種ごとに平均および標準偏差を求める
yield_by_variety <-
  yield %>%
  group_by(variety) %>%
  summarise(N = n(),
            mean = mean(yield),
            sd = sd(yield),
            .groups = "drop")

yield_by_variety
```

variety	N	mean	sd
A	4	308.8474	8.078538
B	4	254.1285	8.393439
C	4	255.9087	13.671605

```r
p <- ggplot(data = yield_by_variety,
            mapping = aes(x = mean, y = variety, color = variety))

p2 <- p + geom_pointrange(mapping = aes(xmin = mean-sd, xmax = mean+sd)) +
    guides(color = FALSE) +
    labs(x = "yield", title = "Fig.2 各品種の平均収量 (±SD)")

# ブロックごとに平均および標準偏差を求める
yield_by_block <-
  yield %>%
  group_by(block) %>%
  summarise(N = n(),
            mean = mean(yield),
            sd = sd(yield),
            .groups = "drop")
yield_by_variety
```

variety	N	mean	sd
A	4	308.8474	8.078538
B	4	254.1285	8.393439
C	4	255.9087	13.671605

```r
p <- ggplot(data = yield_by_block,
            mapping = aes(x = mean, y = block, color = block))
p3 <- p + geom_pointrange(mapping = aes(xmin = mean - sd, xmax = mean + sd)) +
    guides(color = FALSE) +
    labs(x = "yield", title = "Fig.3 各反復の平均収量 (±SD)")

p_all <- ggpubr::ggarrange(p2, p3)
p_all
```

Fig.2 各品種の平均収量(±SD)

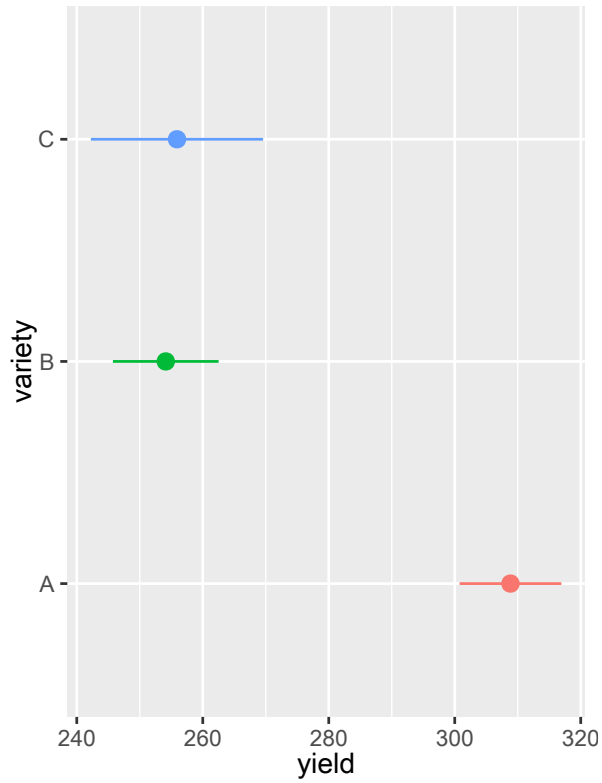
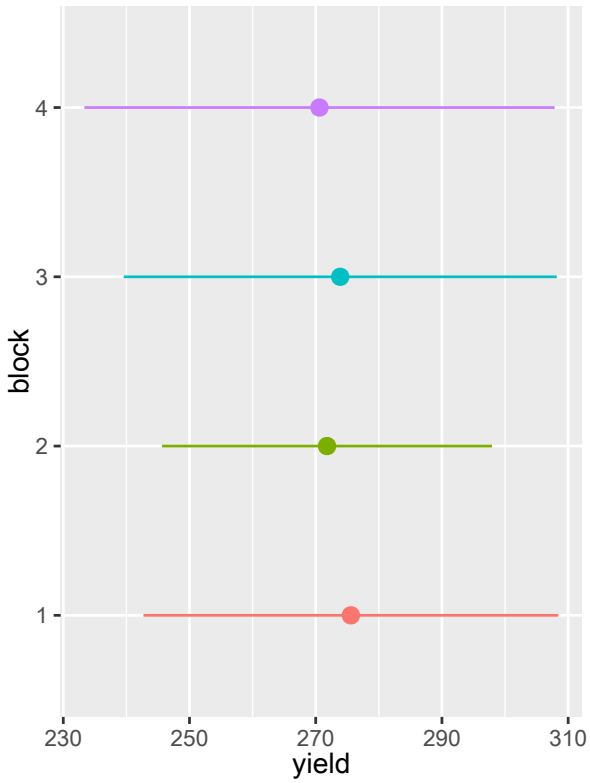


Fig.3 各反復の平均収量(±SD)



- 収量の軸が 0 スタートでないことに注意.
- Fig.1, 2 より品種間に差はありそう．A が一番大きくてついで B, C か.
- Fig.3 より反復間で大きなばらつきはなさそう．→ 環境要因の差は少ないか.

実際に検定をかけよう．乱塊法を使ってるので分散分析を行う

```
# 乱塊法を行う
# 交互作用は考えない
anova_RB <- aov(yield ~ variety + block, data = yield)

# R Markdown で出力を整えるためのコード
# anova_RB
# でよい
res_RB <- broom::tidy(anova_RB)
res_RB
```

term	df	sumsq	meansq	statistic	p.value
variety	2	7733.10504	3866.55252	25.107495	0.0012159
block	3	43.87657	14.62552	0.094971	0.9600323

term	df	sumsq	meansq	statistic	p.value
Residuals	6	923.99960	153.99993	NA	NA

```
res_tukey <- TukeyHSD(aov(yield$yield ~ yield$variety))

res_tukey$`yield$variety`
```

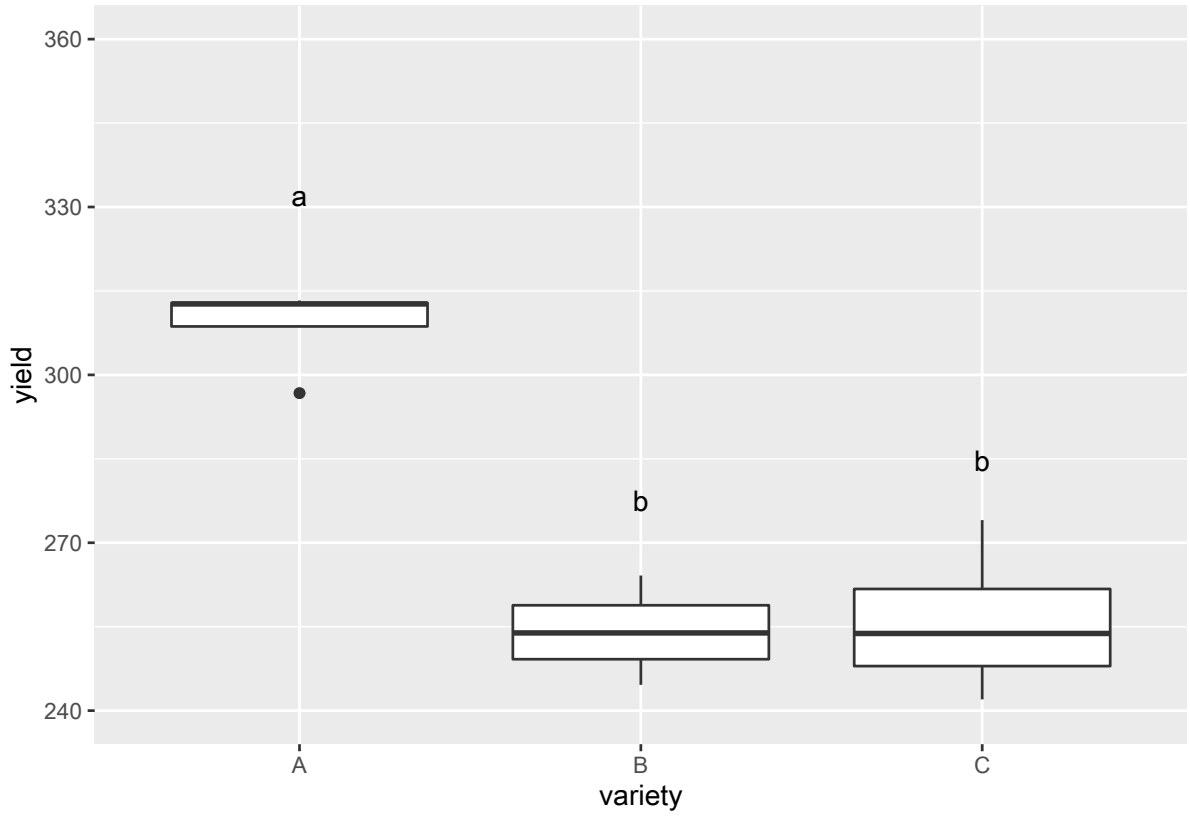
```
##           diff       lwr       upr        p adj
## B-A -54.718864 -75.19226 -34.24547 0.0001018444
## C-A -52.938698 -73.41210 -32.46530 0.0001317481
## C-B   1.780166 -18.69323  22.25356 0.9681473384
```

A と B，A と C に有意差があり，B と C には有意差がなかった．

今回のような圃場条件では，A が一番取れる品種だろう．

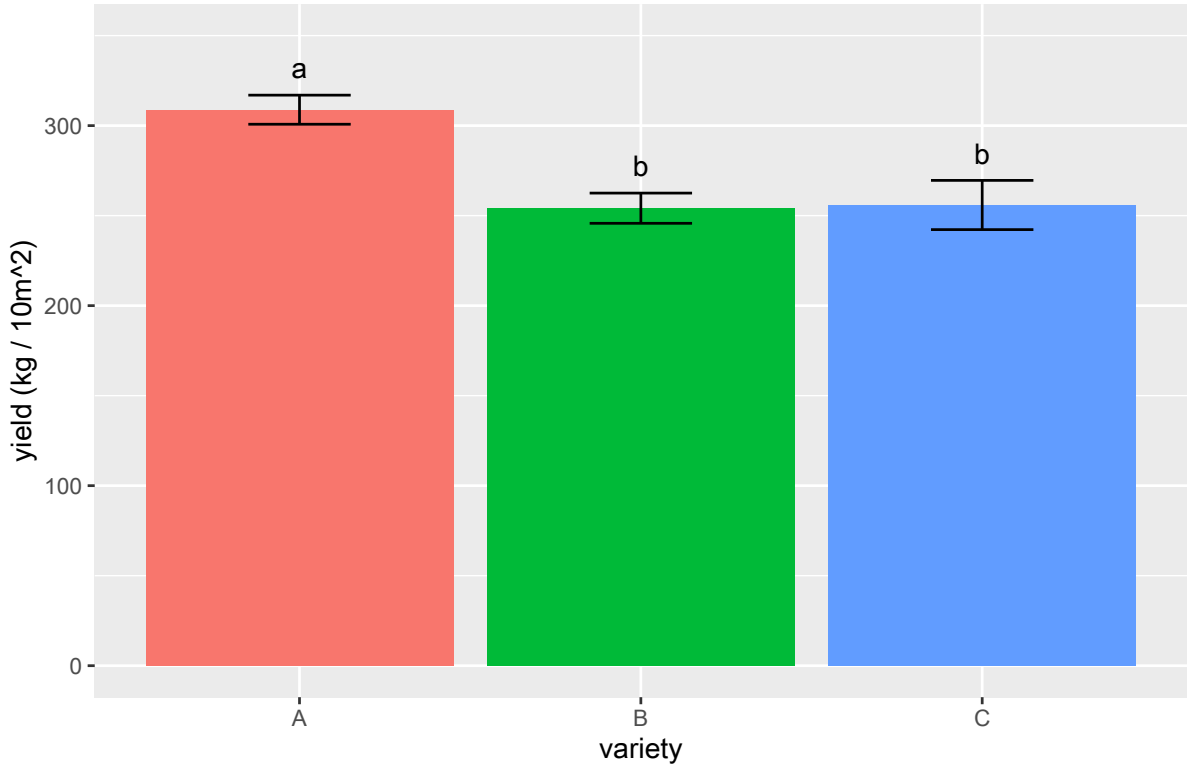
有意差の結果を踏まえてもう一度グラフを描く．

```
# 箱ひげ図
p <-  ggplot(data = yield,
           mapping = aes(x = variety, y = yield))
# annotate はプロットに文字を描くための関数です
# 文字の場所を決める
position_text <- yield_by_variety$mean + yield_by_variety$sd + 15
p + geom_boxplot() +
    scale_y_continuous(limits = c(240, 360)) +
    annotate("text", x = "A", y = position_text[1], label = "a") +
    annotate("text", x = "B", y = position_text[2], label = "b") +
    annotate("text", x = "C", y = position_text[3], label = "b")
```




```
# 棒グラフ
p <- ggplot(data = yield_by_variety,
            mapping = aes(x = variety, y = mean, fill = variety))
position_text <- yield_by_variety$mean + yield_by_variety$sd + 15
p + geom_col() +
    geom_errorbar(mapping = aes(ymin = mean-sd, ymax = mean+sd), width = 0.3) +
    scale_y_continuous(limits = c(0, 350)) +
    annotate("text", x = "A", y = position_text[1], label = "a") +
    annotate("text", x = "B", y = position_text[2], label = "b") +
    annotate("text", x = "C", y = position_text[3], label = "b") +
    labs(y = "yield (kg / 10m^2)",
         title = "Fig. 各品種の平均収量 (±SD)") +
    theme(legend.position = "")
```

Fig. 各品種の平均収量(±SD)



補足

今回のデータは R を使って作成しました.

品種 A, B, C について

- A: 平均 300，標準偏差 10 の正規分布
- B: 平均 260，標準偏差 10 の正規分布
- C: 平均 250，標準偏差 10 の正規分布

です．ブロック間で変動はありません.

```
set.seed(seed = 0)
va <- rnorm(n = 4, mean = 300, sd = 10)
vb <- rnorm(n = 4, mean = 260, sd = 10)
vc <- rnorm(n = 4, mean = 250, sd = 10)
# データフレームにまとめる
df <- data.frame("variety" = c(rep("A", 4), rep("B", 4), rep("C", 4)),
                 "block" = c(rep(1:4, 3)),
```

```
                  "yield" = c(va, vb, vc))
df
```

variety	block	yield
A	1	312.6295
A	2	296.7377
A	3	313.2980
A	4	312.7243
B	1	264.1464
B	2	244.6005
B	3	250.7143
B	4	257.0528
C	1	249.9423
C	2	274.0465
C	3	257.6359
C	4	242.0099

```
# csv ファイルに書き出し
write.csv(df, "yield.csv", row.names = FALSE)
```

文献

おそらくネットを調べればたくさん出てくると思います．以下私が読んだ，もしくは流し見をして良かった印象がある本をあげます．

R と統計の本

- R によるやさしい統計学 by 山田剛史・杉澤武俊・村井潤一郎
- R による統計解析 by 青木繁伸
- データ解析のための統計モデリング入門一般化線形モデル・階層ベイズモデル・MCMC by 久保拓哉
- R と Stan ではじめるベイズ統計モデリングによるデータ分析入門 by 馬場真哉

可視化

- データ分析のためのデータ可視化入門 (原文: Data Visualization: A Practical Introduction) by キーラン・ヒーリー (訳: 瓜生真也・江口哲史・三村喬生)
- Google 流資料作成術 (R は使ってない) by コール・ヌッスバウマー・ナフリック (訳: 村井瑞枝)

実験計画法など

- 入門実験計画法 by 永田靖
- 統計的多重比較法の基礎 by 永田靖

インターネットサイト

- R Tips (http://cse.naro.affrc.go.jp/takezawa/r-tips/r.html)
- biostatistics (https://stats.biopapyrus.jp)
- Qiita (エラーで困った時やこのグラフどうやって描くんや？って時に行きつくことが多い)

文献などは別の機会にきちんとまとめたいです.