

卒業論文

メインタイトル

—サブタイトル—

学科とか

コースとか

おなまえ

目次

1	問題設定	1
2	先行研究の検討	2
2.1	word で卒論の問題点 . . . . .	2
2.2	先行事例 . . . . .	2
3	執筆方法	3
3.1	本文 . . . . .	3
3.2	表 . . . . .	3
3.3	図 . . . . .	4
3.4	引用 . . . . .	4
3.5	注 . . . . .	4
3.6	文献リスト . . . . .	5
3.7	レイアウト . . . . .	5
3.8	フォント . . . . .	5
	引用文献	i

## 1 問題設定

あらゆる分野の計量的研究において、コードの管理というのは、誰しものが頭を悩ます問題である。そこで、本稿では **Rmarkdown** から直接的に **pdf** を出力し、解析とレポート生成を同時に行うことで、この問題に対する解決法を提示する<sup>1)</sup>。

### [注]

<sup>1)</sup> 注はこのような形で記入する。執筆規定にもよるが、今回の卒論は章末に注釈を置くことになっているのでこのような形式になっている（詳細は後述）。

## 2 先行研究の検討

### 2.1 word で卒論の問題点

なぜ word で卒論を書きたくないか。大きな理由は2点ある。 第一に、レイアウトを考えるのが面倒だからである。word で執筆する場合、excel でなどから図表をインポートする時に、どこに設置するかのようなことをいちいち考えなければならない<sup>2)</sup>。

第二に、コピペ汚染の問題である。計量的研究では、SPSS, stata といった統計ソフトで解析した結果を excel にコピペし、excel 上で罫線を書く、桁数を調整するなどの整形作業を行い、そこから word にコピペする、という作業が必要である。しかし、この方法では、解析をやり直すたびに、コピペ、図表の整形をやり直さなければならない。

そんなもの、全ての分析が終わった後に図表を挿入すれば良いだけではないか、という声が聞こえてきそうである。しかし、分析が終わったと自分で思っても、予定変更が起こることは頻繁に起こる（一度書いたことがある人ならばわかるはず）。

以上の理由から、word での執筆を少なくとも私はやりたくないと考えた。

### 2.2 先行事例

word を避けて論文を書こうという試みは、社会科学系においてもいくつか存在する。本稿はこれらの先行事例を参考に、自分なりにブラッシュアップしたものを共有している。インスピレーションを与えてくれた先行事例の数々には、篤く感謝申し上げる。

まず、Mugiyama (2017) では、Latex を用いて執筆する方法が紹介されている。Latex は数式に強く、理系の論文では良く使用されている。しかし、本稿の目的は解析とレポートの同時化であり、この方法では根本的な解決には至っていない。

次に、Nigimitama (2019) では、本稿と同じく Rmarkdown から pdf を出力する方法が紹介されている。本稿はこれをベースに、卒論執筆規定が求める調整を施している。

## [注]

<sup>2)</sup> ただし、今回紹介する方法でも、図表の位置をまったく気にしなくてよくなるわけではない。

## 3 執筆方法

### 3.1 本文

本文は普通に書けばよい。

改段落は文末で半角スペース 2 つ。

### 3.2 表

Latex 対応は開発途上ではあるが、最近 `gt` を好んで使っている（複雑な表でなければ `kable` でも良い）。`knitr` ではキャプションを `kable` 関数の中に書くことで実装できるが、`gt` はまだ対応していないらしい。詳しくは Github の issue を参照。卒論では `level3` の見出しを使ったが、あまりいい方法ではない。

また、1 ページに収めたいのに表がデカすぎる場合、`\scalefont` という latex のパッケージで大きさを変更できる。

表 1：iris の記述統計量

variables	n	mean	sd	min	max
setosa					
Petal.Length	50	1.46	0.17	1.0	1.9
Petal.Width	50	0.25	0.11	0.1	0.6
Sepal.Length	50	5.01	0.35	4.3	5.8
Sepal.Width	50	3.43	0.38	2.3	4.4
versicolor					
Petal.Length	50	4.26	0.47	3.0	5.1
Petal.Width	50	1.33	0.20	1.0	1.8
Sepal.Length	50	5.94	0.52	4.9	7.0
Sepal.Width	50	2.77	0.31	2.0	3.4
virginica					
Petal.Length	50	5.55	0.55	4.5	6.9
Petal.Width	50	2.03	0.27	1.4	2.5
Sepal.Length	50	6.59	0.64	4.9	7.9
Sepal.Width	50	2.97	0.32	2.2	3.8

### 3.3 図

図 1 が iris の散布図である。

図の大きさは、`out.width`、`out.height`、`fig.width`、`fig.height` の 4 つのパラメータでいじることになる。最初の二つで描画領域を指定し、後の二つで図の比率をいじるイメージ。実際にいろいろやってみて感覚を掴んでほしい。

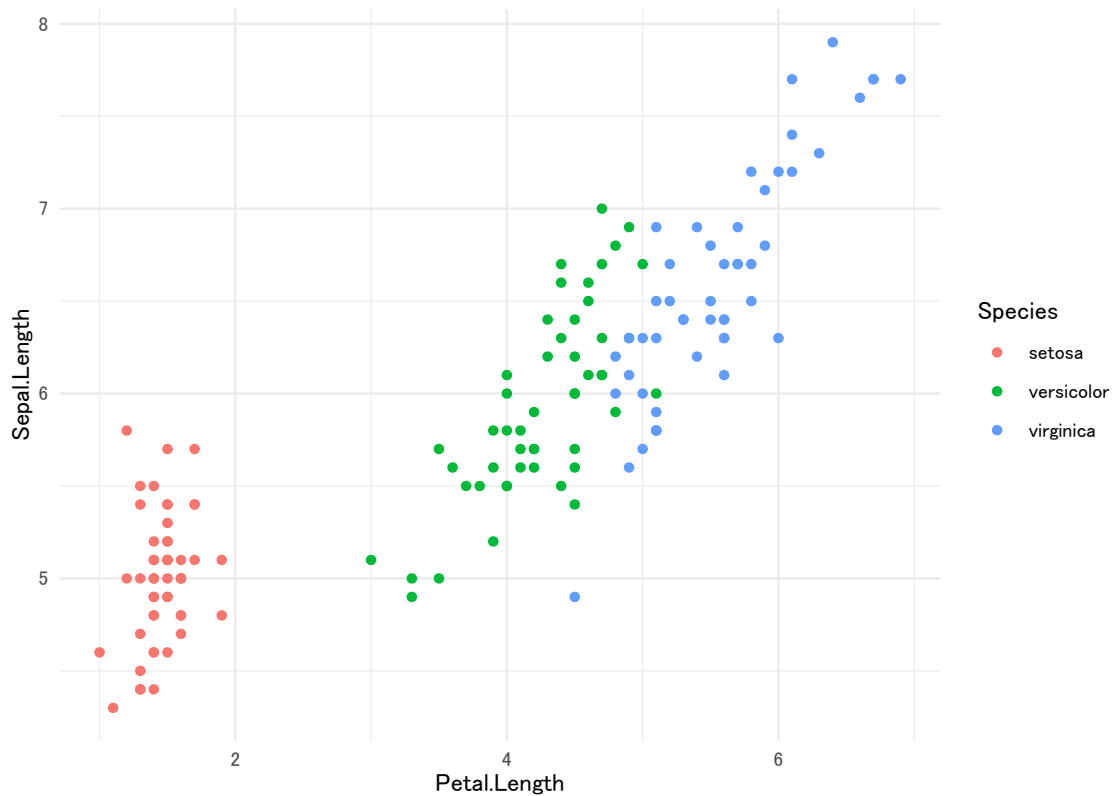


図 1: iris の散布図

### 3.4 引用

`bibtex` という便利なものがあるが、今回は利用していない。そもそもこれが便利なのはスタイルファイルなどが公式に提供されている場合であり、`latex` の `l` も知らない人が多い社会科学系では、強みを十分に生かすことができない。

### 3.5 注

既に登場しているが、注をつけたい場所に `[^番号]` と打ち、その後 `[^番号]:文章` という形で注釈を書く<sup>3)</sup>。実際に出力される注番号は、ここで記入する番号とは異なり、`latex` 側が

勝手に順番・番号振りをやってくれる。注釈を書いたら、`theendnotes` というコマンドを、注釈を挿入したいところを書く。すると、コマンドまでの注釈がまとめて吐き出される。

なお、デフォルトでは脚注になってしまうので、`preamble` で、定義を色々いじっている (hnsn1202 2013)。そちらも参照のこと。

### 3.6 文献リスト

引用のところでも書いたが、`Bibtex` は使用していない。また、引用文献はぶら下げインデントをする慣習があるので、`hangall` という環境を `preamble` で定義している (hakuoku 2018)。

### 3.7 レイアウト

`geometry` パッケージを `preamble` に書き、余白を設定する。

一行当たりの文字数は、`\mojiparline` というマクロを `preamble` で定義している (ebal・北川 2020)。ページ当たりの行数は、`\linesparpage` というマクロを `preamble` で定義しているのでそれを使うか、`word` で行数を設定した時の行送りの値を参考にするという手もある？

### 3.8 フォント

どのフォントを使うかは、`preamble` で指定する。`luatexja-fontspec` というパッケージを使い、`\setmainfont` で英数字、`\setmainjfont` で日本語のフォントを設定できる。なお、本稿は MS 明朝 + Times New Roman で作成している。詳しくは公式 vignette(<http://mirrors.ibiblio.org/CTAN/macros/luatex/generic/luatexja/doc/luatexja-ja.pdf>)を参照。

## [注]

<sup>3)</sup> 例えばこんな感じ。

## 引用文献

- ebal・北川弘典, 2020, 「LuaLaTeX-j<sub>a</sub> の`\ltjgetparameter` について」 (2021 年 1 月 10 日取得 <https://oku.edu.mie-u.ac.jp/tex/mod/forum/discuss.php?d=2907>).
- hakuoku, 2018, 「カスタム字下げ・地寄せ」 (2021 年 1 月 10 日取得 [https://hakuoku.github.io/agakuTeX/tutorial/5\\_1flush/](https://hakuoku.github.io/agakuTeX/tutorial/5_1flush/)).
- hnsn1202, 2013, 「LaTeX で脚注を文書末もしくは章末に書く手順」 (2021 年 1 月 10 日取得 <http://hnsn1202.hateblo.jp/entry/2013/02/06/170617>).
- Mugiyama Ryota, 2017, 「修論 LaTeX テンプレート (jsbook 使用)」 (2021 年 1 月 10 日取得 <http://ryotamugiyama.com/2017/02/17/texttemplate/>).
- Nigimitama, 2019, 「[R] RMarkdown を PDF にして論文を書くときのテンプレート」 (2021 年 1 月 10 日取得 <https://nigimitama.hatenablog.jp/entry/2019/01/13/021447>).