

### 学術研究のためのオープンソース・ソフトウェア(5)LyX

MIYAZAKI, Kenji / 宮崎, 憲治

---

(出版者 / Publisher)

法政大学経済学部学会

(雑誌名 / Journal or Publication Title)

経済志林 / The Hosei University Economic Review

(巻 / Volume)

84

(号 / Number)

1・2

(開始ページ / Start Page)

137

(終了ページ / End Page)

163

(発行年 / Year)

2016-09-27

(URL)

<https://doi.org/10.15002/00013315>

## 【研究ノート】

# 学術研究のためのオープンソース・ ソフトウェア (5) : LyX

宮 崎 憲 治

## 1 LyXとは

ワードやパワーポイントで原稿を書いてきた人が、いきなり L<sup>A</sup>T<sub>E</sub>X (以下 latex と表記) で原稿を書くと、慣れてなくて、ついつい頻繁にコンパイルして、どのように見えるのか PDF で確認してしまう。数式を入力すると、補完機能を備えたエディタを用いても、括弧のつけ忘れなど、コマンド名の打ち間違いなどでエラーになることが多い。特に行列や表の入力が面倒である。

数式を頻繁に用いる経済学者は Scientific Word (<http://www.mackichan.com/index.html?products/sw.html~mainFrame>) を用いていることが多い。Scientific Word は、昔は英語しか入力できなかったが、現在は日本語入力が可能になっている。ただ数式の入力の仕方が latex と違うようなので、latex を使っていたことがある人ほどストレスを感じる人が多い。さらに Scientific Word は有料でオープンソース・ソフトウェアではない。

LyX (<http://www.lyx.org>) は、ワープロのような感覚で latex を操作できるオープンソース・ソフトウェアである。ワープロのような感覚といっても、厳密には同じではない。市販のワードプロセッサは、WYSIWYG、

すなわち「What You See is What You Get (見ている状態が手に入るのと同じ状態)」の原則に基づいている。それに対して, LyX の原則は WYSIWYM, すなわち「What You See Is What You Mean (見ている状態が意図している状態)」である。ワープロと違い印刷物とは完全に同じでない。ワープロはバイナリファイルを扱うけれど, LyX はテキストファイルを扱い, latex のソースコードに変換可能である。また latex と違い, LyX では画面に数式のままで表示される。つまり LyX は latex にもとづく論文を書きたいけれど, latex の数式コマンドから数式が想像できない研究者にとって非常に便利なソフトウェアである。

これまで何冊か LyX についての日本語の本が出版されている。おそらく最初に紹介されたのは小浪 (2002) であるが, 現在は絶版である。その後しばらくしてから, 北浦 (2009), 藤田 (2010) が出版された。藤田 (2010) は論文の執筆方法まで言及していて, 私と目的意識が同じであるが, 残念ながらこれも絶版である。さらにここまで挙げてきた本は platex を元に書かれている。宮崎 (2015a) で述べたように, これからは lualatex や xelatex が主流になると考えられる。

この研究ノートは最近の latex の動向を踏まえつつ, 学術研究, つまり論文作成に焦点をあてて, LyX について紹介していく。xelatex を紹介している点と, beamer および natbib での扱い方を説明している点と, knitr を用いた再現可能論文の作成方法を提示している点が上記の既存の本に言及されていない新しい点である。

LyX についてはインターネットでの情報もたくさんある。日本語については

<https://texwiki.texjp.org/?LyX>

がそれぞれの OS とコンパイルエンジンに応じた設定方法が随時更新されている。またいささか情報が古いが

<http://www.int.otaru-uc.ac.jp/lyx-howto%5D%5D>

がいまだに有益である。

この研究ノートの構成は以下の通りである。まず LyX の導入方法を述べる。次に文章の作成方法を説明する。他にも beamer の作り方, 参考文献の作成方法, knitr との連携についても言及し, 最後にまとめとして, 私の利用方法を述べる。

## 2 LyXのインストールと設定

すでに texlive 2015以上をダウンロードしているとする。texlive について宮崎 (2015a) を参照されたい。

### 2.1 LyXのインストール

Windows や Mac などでバイナリをインストールするには,

<http://www.lyx.org/WebJa.Home>

にいき, ダウンロードして実行すればよい。執筆時点の最新版は2.2である。私自身 Mac で LyX 2.2 を用いているが, 執筆時点いくつか不具合があるため2.1と併用している。近い将来安定すると思われる。

Windows で bundle 版があるが, これは latex エンジンも同時にインストールするインストーラーである。ただ日本語に対応していない MikTeX (<http://miktex.org/>) をインストールするので, 日本語を使用するなら, 別途 texlive を導入したあとで latex エンジンを含んでいないインストーラー, たとえば執筆時点での最新版である **LyX-220-Installer-2.exe** を選択しなければならない。

インストーラーを起動して, いくつかのオプションが選択できる。私は設定ファイルが編集しやすいように, 「自分専用にインストールする」を選

扱っている。なお thesaurus を追加すると全ての言語についての類語辞典がダウンロードされインストールに時間がかかることに注意されたい。デフォルトの英語の類語辞典だけが導入されれば十分である。

また Mac にインストールする際、最新の Mac OS X 10.11 だとシステム環境設定のセキュリティとプライバシーでダウンロードしたアプリケーションの実行許可について、全てのアプリケーションを許可しなければならない。

Ubuntu の場合、ターミナル上で

```
sudo apt-get install lyx
```

と入力すればよいがバージョンが古いことがある。最新版を ppa 経由で利用したければ、インストールを実行する前にターミナル上で

```
sudo apt-add-repository ppa:lyx-devel/release  
sudo apt-get update
```

を実行する。

## 2.2 bxjsarticleの導入

latex で日本語を用いるにはコンパイルエンジンを platex にして、スタイルファイルを jsarticle を用いるのが定番であったが、最近 xelatex, lualatex, pdfflatex, uplatex が注目されている。texlive 2015 以降なら、bxjsarticle を用いれば、xelatex, lualatex, pdfflatex, uplatex といった、ユニコードを扱うコンパイルエンジンで日本語表示が可能になる。ここではまずどのようにすれば LyX に導入できるか述べ、そのための設定を示す。

まず `bxjsarticle.layout` ファイルを新規作成して以下の内容を入力する。

```
##% Do not delete the line below; configure depends on this
# \DeclareLaTeXClass{Japanese Article (bxjsarticle)}
# \DeclareCategory{Articles}
# Japanese new article textclass definition file.

# This style provides japanese features
Format 60
Provides japanese 1

# Input general definitions
Input article.layout
```

真ん中あたりのフォーマットの番号は、LyX のバージョンによって違う。たとえばバージョン2.2ならば60になる。

このファイルを Mac OS Xならフォルダ `~/Library/Application Support/LyX-2.2/layouts/` にコピーする。linux なら `~/.lyx/layouts` である。Windows なら `%APPDATA%\Roaming\LyX2.2\layouts\` である。その後メニューバーの [ツール (T)] をクリックし、出てきたメニューの [再初期設定 (R)] をクリックして LyX を再起動する。

なお `bxjsarticle` 以外のスタイルファイル `bxjsbook`, `bxjsreport` も同様にすればよい。これらを LyX に導入するには以下を参照されたい。

<https://texwiki.texjp.org/?LyX%2F%E8%A8%AD%E5%AE%9A#g3fd040c>

そこには `ltsarticle` および `ltsbook` についての設定も説明されている。

## 2.3 環境設定

LyX 全体の設定は、Mac だと [LyX] をクリックし、出てきたメニューの [環境設定...] で実行する。そうすると図 1 が出てくる。Mac 以外だと [ツール] をクリックする。



図 1 環境設定画面

表示速度が遅いと感じるなら、[操作性] → [表示] において、[Display Graphics] のチェックボックスのチェックを外し、[Instant Preview] も「無効」にする。もしチェックをいれていて表示されていないときは、画像変換ソフト ImageMagick (<http://www.imagemagick.org>) が導入されていないので適宜導入する必要がある。

[パス]で実行ファイルを指定することができる。自動的にパスが指定されているが、上手いかなければここで指定する。El Captain 以降の Mac OS では latex の実行ファイルは /Library/TeX/texbin になっていることに注意されたい。

[言語設定] → [言語] において babel を off にする。まず [言語パッケージ] を「任意設定」し、右側の入力ボックスを空欄にする。次に [言語を大域に設定] のチェックボックスのチェックを外す。もしひとつのファイルに多言語を導入するのなら、この設定を行わない。

[出力] → [Latex] において次のように変更する。まず [Latex フォントエンコーディング] が「T1」でチェックが入っていることを確認する。

さらに `bibtex` がユニコードで動くよう `upbibtex` に変更する。具体的には [書誌情報の生成] の [処理子] を任意設定にして、すぐ下のコマンドを `upbibtex` とする。また [日本語処理子] の入力ボックスを空欄にする。

さらに索引作成も `mendex` でなくユニコード対応にした `upmendex` ほうがよい。ただ `texlive 2016`以降からしか含まれていない。`texlive 2016`以降なら、[索引情報の生成] の [処理子] を任意設定にして、すぐ下のコマンドを `upmendex -c -q` とする。また [日本語処理子] の入力ボックスを空欄にする。`texlive 2015`を用いているのなら、論文作成には索引を使うことはほとんどないので、`mendex -c -q` のままでよいだろう。

[ファイル処理] → [変換子] で次の2つを変更する。まず内部でユニコードに対応した `uplatex` で実行するために、[Latex (pLaTeX) → DVI] の変換子を `uplatex` に、追加フラグを `latex=uplatex` に修正する。次に `latex` のソースコードを日本語であっても読み込めるように [Latex (plain) → LyX] の変換子で `tex2lyx-f $$i $$o` となっているのを `tex2lyx -e utf8 -f $$i $$o` に修正する。

ところで [ツール] → [環境構成] を実行したときに、いくつかはデフォルトに戻ってしまうことがあるので注意されたい。私の環境では `upbibtex` と `uplatex` と `tex2lyx` の設定が元に戻ってしまう。その場合、再度設定する必要がある。

## 2.4 文書の設定

以上のように設定してメニューバーの [ファイル] をクリックし、出てきたメニューの [新規] で実行する。その新しいファイルに何かアルファベットの文章を入力し、入力し終わったら、[文章] → [PDF (pdflatex)] を選択すれば、英語のみの文章は PDF に変換され、表示されるはずである。その他の細かい設定や日本語表示するためにはそれぞれの文書ごとの設定をおこなう必要がある。

出力エンジンごとの設定をおこなうため、メニューバーの [文書] をク





図2 文書設定画面

リックし、出てきたメニューの「設定」をクリックする。そうすると図2が出てくる。ここでは xelatex, lualatex, uplatex, pdflatex について説明する。なおこれらのエンジンの違いによる出力の違いについては以下を参照されたい。

<https://texwiki.texjp.org/?BXjscls>

## 2.4.1 xelatex

[文章] → [設定] → [文章クラス] の [文章クラス] で「Japanese Article (bxjsarticle)」を選択する。オプション部分に

```
xelatex, ja=standard
```

と入力する。なお「日付」や「概要」などを英語表記にしたければ、english を書き加える。そうすれば英語論文でも利用可能である。

日本語の入力のためには [文章] → [設定] → [フォント] で非  $\text{T}_\text{E}\text{X}$  フォントの使用にチェックを入れる必要がある。そうすると [文章] → [設定] → [言語] の [文字コード] で、「ユニコード (XeTeX) (UTF8)」が強

制的に選択されるはずである。英語まじじりの文章で、スペルチェックを希望するなら、[言語] を「英語」のままにしておく。もし PDF の栞（簡単な目次）を作りたい場合、[文章] → [設定] → [PDF特性] で [一般] の「Hyperref サポートを使う」をチェックする。xelatex をデフォルトとして用いる場合、図 2 の下側にある [このクラスの規定値を使う] をクリックする。そうすれば新規作成のたびに同じ設定が用いられる。

## 2.4.2 lualatex

[文章] → [設定] → [文章クラス] の [文章クラス] で「Japanese Article (bxjsarticle)」を選択する。オプション部分に

```
lualatex, ja=standard
```

と入力する。なお英語表記にしたければ **english** を書き加える。

[文章] → [設定] → [フォント] で非 $\text{T}_{\text{E}}\text{X}$ フォントを使用にチェックを入れなくても日本語の入力は可能であるが、[文章] → [設定] → [言語] の [文字コード] で、「ユニコード (XeTeX) (UTF8)」を選択する必要がある。なお英語まじじりの文章で、スペルチェックを希望するなら、[言語] を「英語」のままにしておく。もし PDF の栞（簡単な目次）を作りたい場合、[文章] → [設定] → [PDF特性] で [一般] の「Hyperref サポートを使う」をチェックする。

## 2.4.3 uplatex

日本語論文をユニコード対応した uplatex で PDF を作成するなら、まず [文章] → [設定] → [文章クラス] の [文章クラス] を「Japanese Article (bxjsarticle)」を選択する。オプション部分に

```
uplatex, dvipdfmx, ja=standard
```

と入力する。なお英語表記にしたければ, **english** を書き加える。

さらに [文章] → [設定] → [言語] の [文字コード] で「Japanese (pLaTeX) (UTF8)」を選択する。なお英語まじじりの文章で、スペルチェックを希望するなら、[言語] を「英語」のままにしておく。

なお他のエンジンと違い、PDFの葉を作りたい場合でも、[文章] → [設定] → [PDF特性] で [一般] の「Hyperref サポートを使う」をチェックしない。使うためのコマンドを直接プリアンブル部書き加える。

[文章] → [設定] → [Latexプリアンブル] に

```
\usepackage{textcomp}
\usepackage{lmodern}
\usepackage{hyperref}
```

と書き加える。また `uplatex` でなく `platex` をコンパイルエンジンに使うなら、`latex` プリアンブル部に

```
\usepackage{pxjahyper}
```

を付け加える必要がある。

#### 2.4.4 pdflatex

スタイルファイル `bxjsarticle` を使えば、`latex` エンジンが `pdflatex` であっても日本語を扱える。[文章] → [設定] → [文章クラス] の [文章クラス] を「Japanese Article (bxjsarticle)」を選択する。オプション部分に

```
pdflatex, ja=standard
```

と付け加える。

[文章] → [設定] → [言語] の [言語] を英語に、[文字コード] を「ユニコード (utf8)」を選択する。もし PDF の葉 (簡単な目次) を作りたい場合、[文章] → [設定] → [PDF特性] で [一般] の「Hyperref サポートを使う」をチェックする。

幾つかのフォントに対応するために [文章] → [設定] → [L<sup>A</sup>T<sub>E</sub>X プリアンプル] において

```
\usepackage{textcomp}
\usepackage{lmodern}
```

と書き加える。

### 3 使い方

日本語が入った文章は、前節のように正しく設定されていれば、[文章] → [他の形式] → [PDF (xelatex)] などで PDF が作成され表示される。設定によっては他のコンパイルエンジンでも表示される。

ただ PDF が作成されるといっても、テンポラリーファイルとして生成されただけなので、ファイルとして出力するには、[ファイル] → [書き出し] で [PDF (xelatex)] などを選択しなければならない。もし tex のソースコードを出力する必要があるなら、[L<sup>A</sup>T<sub>E</sub>X (xelatex)] などを選択すればよい。

なお文章作成中に、背後の tex のソースコードがどうなっているかを知りたいことがある。その場合、[表示] → [ソース面] を選択するとソースコードが別に表示される。またコンパイルしたとき、どのようなメッセージが出ていたのかは、[表示] → [ソース面] を選択すると別に表示される。これによりコンパイル中のエラーが発生したときや、途中で止ったときの原因の究明に役立つ。こうした LyX 画面は図 3 の下部分になる。

図 3 の上部にあるツールバーのボタンを押せば、メニューをクリックすることなく、いろいろな機能が実現できる。どのような機能があるかは、ツールボタンの上にマウスのポインタを当てれば、ショートカットとともに出力される。その他の使い方は本体に付属しているヘルプが充実してかつ日本語に翻訳されているのでそれを熟読していただきたい。以下、論文作

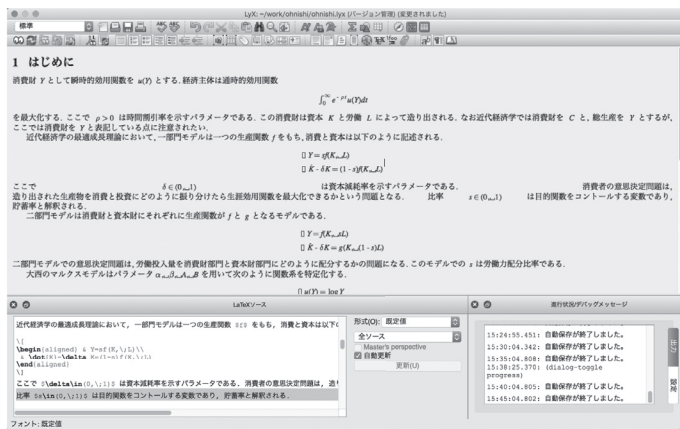


図3 LyX画面

成に特化して基本的な使い方を述べる。

### 3.1 編集

文字修飾や文字の大きさは、該当箇所をハイライトして、[編集] → [文字様式] を選択するかツールバーの該当ボタンで変更する。また右揃えやセンタリングなどは、[編集] → [段落設定] を選択するかツールバーの該当ボタンで変更する。

### 3.2 環境選択ボックス

L<sup>A</sup>T<sub>E</sub>X でのさまざまな環境 (environment) はツールバーの左側に環境ツールボックスをつかうことで導入可能である。標準にもどる作業はよく使うので、キーバインディングを覚えたほうがよい。Macだと [Ctrl] と p を同時に押し s と打つ。Windows と Ubuntu では [Alt] と p を同時に押し、s と打つ。

バージョン2.1だと Mac と Windows の環境選択ボックス内の日本語フォントがずれている。バージョン2.2にすると Mac は修正されているが

Windows はまだずれている。それを修正するためには表示を英語にするとよい。もしくは Ubuntu を使う。以下、環境選択ボックスで表示できるものをいくつか紹介する。

### 3.2.1 文頭辞

表題 (`\title`), 著者 (`\author`), 日付 (`\date`), 概要 (`\begin{abstract}...`), 住所 (`address`) などの書式を入力できる。また LyX 独自の右寄せ住所 (`lyxrightaddress`) というものもある。

### 3.2.2 セクションニング

節 (`\section{...}`) や小節 (`\subsection{...}`) などを指定できる。番号表示がないセクションニング (`\section*{...}`, `\subsection*{...}`) も選べる。

### 3.2.3 箇条書き

いくつかの箇条書きが作れる。記号のみの `itemize`, 番号づけの `enumerate`, 語句つきの `description` 環境が選べる。また LyX 独自の `lyxlist` という表題付きの環境 (ラベリング) もある。箇条書きについてはツールバーに該当ボタンがある。

### 3.2.4 引用

段落の字下げがない引用 (`quote`) だけでなく, 字下げのある引用 (`quotation`) や字上げのある詩句 (`verse`) も選択可能である。

### 3.2.5 コード

このままコード入力の `verbatim` だけでなく, より綺麗に表示できる LyX コード `\begin{lyxcode}...` というものもある。

## 3.3 各種挿入

その他の必要なコマンドは [挿入] → [T<sub>E</sub>X コード] を選択するかツールバーの該当ボタンをクリックして, コマンド入力すればよい。ショートカットは Mac ではコマンドキーと 1 (エル) を同時に押し, それ以外だと [Ctrl] と 1 (エル) を同時に押す。

場合によってはプリアンブルに `usepackage` を指定しなければならない。たとえば論文でよく求められるダブルスペースは `\usepackage{setspace}` をプリアンブル部書き、以降をダブルスペースにする際には `\doublespacing` とコマンド入力する。

なお、LyXはあらかじめいくつかの機能がコマンド入力なしで挿入可能である。そうしたもののいくつかを以下に示す。

### 3.3.1 脚注

[挿入] → [脚注] を選択するかツールバーの該当ボタンをクリックすることで、脚注が挿入される。また傍注も可能である。

### 3.3.2 数式

[挿入] → [数式] で、[行内数式] や [別行立て数式] などが挿入される。[行内数式]についてはツールバーに該当ボタンが存在する。ショートカットは Mac ではコマンドキーと `m` を同時に押し、それ以外だと `[Ctrl]` と `m` を同時に押す。[別行立て数式] は、Mac ではコマンドキーとシフトキーと `m` を同時に押し、それ以外だと `[Ctrl]` とシフトキーと `m` を同時に押す。

数式入力を行えば自動的に数式のツールバーが下部分に登場する。これよりコマンド名を覚えていないものも入力可能である。特に行列の入力は便利である。なお数式は `amslatex` にも対応している。もちろん `latex` のコマンドをそのまま入力してもよい。

### 3.3.3 図

[挿入] → [画像] を選択するかツールバーの該当ボタンをクリックすることで、画像が導入される。png や eps や pdf や jpeg などは気にしなくても自動的に変換される。サイズ変更も可能である。番号付の図にしたければ後述のフロートを作成する必要がある。

### 3.3.4 表

[挿入] → [表] を選択するかツールバーの該当ボタンをクリックすることで、行数と列数を指定して、表が導入される。表組みのレイアウトにつ

いてはどのようにすればよいか直感的に分かり、文字や数値を入力するだけでよい。番号付の表にしたければ後述のフロートを作成する必要がある。

Rなどですでに作成した表を挿入することもできる。そのためには [挿入] → [T<sub>E</sub>Xコード] として

```
\include{tabfile}
```

とすると `tabfile.tex` が読み込まれる。

### 3.3.5 フロート

図もしくは表を選択して、[挿入] → [フロート] で図表にキャプションが挿入可能になり、表示位置が上手く余白を見つけて自動的に移動できる。図表以外にもアルゴリズムや折り返しの図表も選択可能である。図フロートと表フロートはツールバーに該当ボタンが存在する。

なお図について図4のようにして、二重に挿入して複数の図を並べることも可能である。そうすれば `subfig` パッケージが自動的に挿入され、特にプリアンブル部に何も付け加える必要はない。

また表に注釈をつけるパッケージ `threeparttable` を加えるには工夫が必要で図5のようにする。その場合プリアンブル部に `\usepackage{threeparttable}` を付け加えることを忘れてはいけない。

### 3.3.6 URL

[挿入] → [URL] を選択するかツールバーの該当ボタンをクリックすることで、リンク付きの `url` が出力できる。これに伴ってプリアンブル部に自動的に書き込まれるので自分で記入する必要はない。

### 3.3.7 相互参照

脚注や数式や図や表などに番号付けして、相互参照ができる。参照するものに、[挿入] → [ラベル] を選択するかツールバーの該当ボタンをクリックすることで、ラベル付けする。その後、参照したいときに [挿入] → [相互参照] を選択するかツールバーの該当ボタンをクリックすることで、参照するものを選ぶ。



### 3.3.8 プログラムリスト

`listings` の設定が適切になされていれば、プログラムリストの挿入も可能である。設定は [文書] → [設定] で図 2 を登場させ、プログラムリストを選択する。なお (u) `platex` を用いている場合、日本語に対応していないため、プリアンブル部に `\usepackage{jlistings}` が必要である。

## 4 その他の機能

その他、論文作成に必要な機能をいくつかあげる。

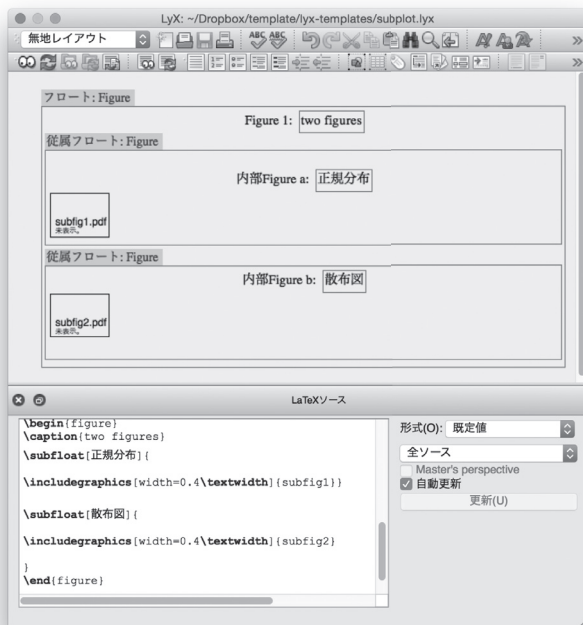


図 4 subfig

## 4.1 スペルチェック

[文書] → [設定] で図2を出現させる。そして [言語] が英語を選択しているとする。そのとき [ツール] → [スペルチェック] を選択するかツールバーの該当ボタンをクリックすることで、スペルチェックが可能である。またその隣のボタンをトグルすることで、wordのようにスペルチェックが必要な単語に波下線がつく。さらに類語を調べることができる。当該単語を選択して、[ツール] → [類語辞典] で類語の候補が挙げられる。

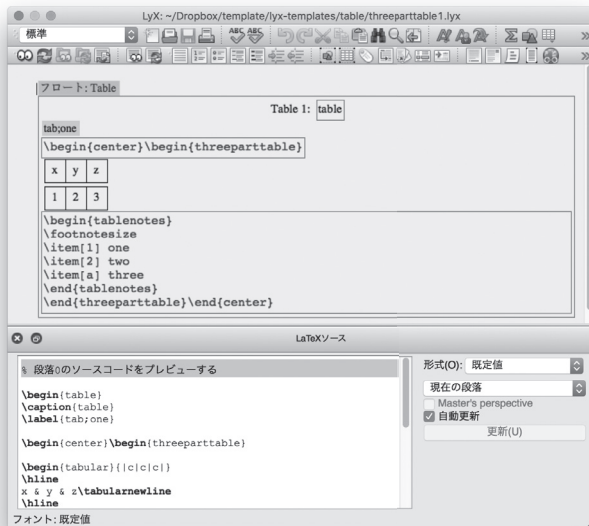


図5 threetable

## 4.2 Beamer

Beamer はプレゼンテーションを作成する latex のスタイルファイルである。事実上, latex で作成するプレゼンテーションファイルの標準となっ

ている。Beamer については宮崎 (2015a) を参照されたい。LyX でも Beamer スタイルを扱える。それぞれの latex エンジンでの設定方法と使い方を述べる。なお古いバージョンの LyX で作成した beamer 対応の文書は最新版では読み込めなくなっているので注意が必要である。

#### 4.2.1 xelatex

xelatex にすれば、uplatex より簡潔に表示できる。[文章] → [設定] → [文章クラス] の [文章クラス] を「Beamer」とする。さらに、[文章] → [設定] → [言語] の [文字コード] で「ユニコード (XeTeX) (UTF8)」を選択する。なお現時点の最新版であるバージョン2.2だと [文章] → [設定] → [フォント] で非 $\text{T}_{\text{E}}\text{X}$ フォントを使用にチェックを入れる必要がある。

[文章] → [設定] → [Latex プリアンプル] に

```
\usepackage{xltextra}
\usepackage{zxjatype}
\usepackage[ipa]{zxjafont}
\usetheme{Warsaw}
\usecolortheme{default}
\usefonttheme{professionalfonts}
```

とする。

デフォルトなら `usetheme` と `usecolortheme` は書き加えなくてもよい。どのようなテーマがあるかについては以下を参照されたい。

<http://www.hartwork.org/beamer-theme-matrix/>

最後の `usefonttheme` は `professionalfonts` を選択することで、綺麗な数式を表示させることができる。

#### 4.2.2 lualatex

luatex についてはプリアンブル部以外は xelatex のときと同じである。

[文章] → [設定] → [Latex プリアンブル] に

```
\usepackage{luatexja}
\renewcommand{\kanjifamilydefault}{\gtdefault}
\usetheme{Warsaw}
\usecolortheme{default}
\usefonttheme{professionalfonts}
```

とする。4 行目は明朝体をゴシック体に変換するコマンドで必要なければ、書き加える必要がない。

### 4.2.3 uplatex

日本語論文を uplatex で PDF 作成するなら、まず、[文章] → [設定] → [文章クラス] の [文章クラス] が「Beamer」になっていて、[クラスオプション] に、dvipdfmx を書き加える。さらに、[文章] → [設定] → [言語] の [文字コード] で「Japanese (pLaTeX) (UTF8)」を選択する。

[文章] → [設定] → [Latex プリアンブル] に

```
\usepackage{textcomp}
\usepackage{lmodern}
\usepackage{pxjahyper}
\renewcommand{\kanjifamilydefault}{\gtdefault}
\usetheme{Warsaw}
\usecolortheme{default}
\usefonttheme{professionalfonts}
\setbeamertemplate{navigation symbols}{{}}
```

と付け加える。4 行目は明朝体をゴシック体に変換するコマンドで必要なければ、書き加える必要がない。8 行目はナビゲーションは dvipdfmx では使えないので、使わないように最後に設定している。

### 4.2.4 pdflatex

[文章] → [設定] → [文章クラス] の [文章クラス] を「Beamer」とする。さらに, [文章] → [設定] → [言語] の [文字コード] で「ユニコード (utf8)」を選択する。

[文章] → [設定] → [Latexプリアンブル] に

```
\usepackage{textcomp}
\usepackage{lmodern}
\usetheme{Warsaw}
\usecolortheme{default}
\usefonttheme{professionalfonts}
```

を書き加える。なおこの設定では日本文書は出力できない。

#### 4.2.5 使い方

スタイルファイルを Beamer にすると, 環境ツールバーが増えていることがわかる。フレームを作り, そのなかに箇条書きをつけるには次の手順をとる。

1. レイアウトを [フレーム] に, フレーム見出しを入力,
2. [→], [RET], [TAB] と打ち, レイアウトを [箇条書き] にして, 箇条書きを入力,
3. [RET] を 3 回うち, レイアウトを [分離線] にする。<sup>\*1</sup>

列挙だけでなく, 環境ツールバーにはフレームや箇条書きだけでなく, 複数の列を表記する設定や, アニメーション表記する重ね書きや, 強調して表示するブロックなどある。

フレームの箇所で右クリックすればフレームオプションを入力することができる。また文字を選択して右クリックし, 文字様式を選択すれば, 色を分けることができる。デフォルトのテーマで赤字になる `\alert{...}` は

<sup>\*1</sup> LyX2.1だと分離線をレイアウトから入力する必要がある。

強調ブロックを、青字になる `\structure{...}` は構造を選択する。

レイアウトをいちいち移動するのは面倒なので、LyX の設定ファイル `user.bind` に以下を付け加える。

```
\bind "M-p f" "layout Frame"  
\bind "M-s M-a" "flex-insert Alert"  
\bind "M-s M-s" "flex-insert Structure"
```

設定ファイル `user.bind` は Mac だとフォルダ `~/Library/Application Support/LyX-2.2/bind/` にある。linux なら `~/lyx/bind` である。Windows なら `%APPDATA%\LyX2.2\bind\` である。こうすると、ショートカットで入力することができる。たとえば Mac だと `[Ctrl]` と `p` を同時に押し、`f` と打てばレイアウトがフレームになる。Windows と Ubuntu では `[Alt]` と `p` を同時に押し、`f` と打つ。

また LyX において、Beamer-Article というスタイルファイルを用いることができる。Beamer で用いたファイルスタイルを変更することで通常の文章ファイルとして出力される。一つのファイルスタイルを変えることで表示用と配布資料に切替えられる。

### 4.3 参考文献作成

LyX は `bibtex/upbibtex` を組み合わせて使うことができる。`bibtex` は参考文献を半自動的に作成するツールである。`upbibtex` はユニコードおよび日本語に対応している。どちらも `bib` ファイルを読み込むことで論文中に参考文献リストを付け加えることができ、簡単に引用スタイルを変更したり、他の論文への再利用も可能になる。`bib` ファイルおよび `bibtex/upbibtex` について宮崎 (2015b) を参照されたい。

#### 4.3.1 natlib

経済学で著者 (年) 形式の引用を導入するための `natlib` スタイルをどの

ように導入するかを述べる。以前説明したように図 1 の環境設定画面から [出力] → [Latex] として BibTeX がユニコードで動くよう upbibtex に変更する。その上で [文書] → [設定] で図 2 を出現させる。そして [書誌情報] で [引用様式] の **Natbib** をチェックし、Natbib 様式を著者一年にする。

開いたファイルの本文内で [挿入] → [一覧] → [BiBTeX書誌情報] とし、データベースで使いたい bib ファイルと、様式で使いたい bst ファイルを選択する。ただし英語論文を **bxjsarticle** 以外のスタイルファイルで **pdflatex** で実行する際には、**jecon.bst** を選択してはうまく動作しない。

そのあと、参照したい箇所を選択して、[挿入] → [文献引用] で引用したい文献を選ぶ。選んだあと [文章] → [他の形式] → [PDF (Xelatex)] などで PDF を作成すれば参考文献が引用された文章が作成される。なおリーディングリストを作るには、[挿入] → [一覧] → [BiBTeX書誌情報] とし、[内容] を「全ての書誌情報」とすればよい。

宮崎 (2015b) で述べたように私は Zotero で文献管理し、bib ファイルを論文ごとに出力させる。私は利用したことがないが、LyX と Zotero を直接連携させる Firefox のアドインが以下のサイトに存在する。

<https://addons.mozilla.org/ja/firefox/addon/lyz/>

<https://github.com/willsALMANJ/lyz>

他にも bib ファイル管理は JabRef (<http://www.jabref.org/>) というソフトが有名である。LyX では JabRef との連携も可能である。それについて以下のサイトを参照されたい。

<http://www.int.otaru-uc.ac.jp/lyx-howto%5D%5D/writing/bib>

#### 4.4 knitr

LyX は Sweave および knitr も対応可能である。どちらも tex に R コー

ドを埋め込んで、文章を作成する R のパッケージである。Sweave はインストールしたらデフォルトで装備されているが、knitr は latex だけでなく html や docx など他の形式での出力も可能であり、現在、再現可能文章を作成するには標準ツールになりつつある。ここでは knitr に絞って説明する。この knitr については宮崎 (2016) も参照されたい。

なお LyX で knitr を利用して PDF 出力することができるのは、latex のエンジンが pdflatex と lualatex と xelatex のときのみである。uplatex のときは Rnw ファイルもしくは tex ファイルとしてしか出力できない。RStudio で tex ファイルに変換したり、Texmaker で pdf に変換する必要がある。

#### 4.4.1 設定

[文書] → [設定] で図 2 を出現させる。そして [モジュール] で Rnw (knitr) を選択し、[追加] を押せばよい。なお Mac の場合、ホームディレクトリの R の設定ファイル ~/.Rprofile をなければ作り、そこに

```
Sys.setlocale(locale = "ja_JP.UTF-8")
```

を追加しないと、日本語のまじったチャックを読みとらない。

#### 4.4.2 使い方

[挿入] → [T<sub>E</sub>Xコード] で

```
<<>>=  
x <-rnorm(10)  
x  
@
```

など打ちこんで、[文章] → [表示 (他形式)] → [PDF (xelatex)] を選択すればよい。

ただ RStudio とちがって、チャックコードを便利に編集する機能はない。複雑なチャックを実行する場合、Rnw 型式で出力して、Rnw で立ち上げて、いろいろ動作確認して、修正したものを再度 LyX に戻って修正する



羽目になってしまうかもしれない。

簡単なチャンク以外は外部ファイルとして読み込ませるほうがミスも少ない。たとえば `hoge.R` という名前の以下の外部ファイルがあるとする。

```
## ----test-a ----
1 + 1
x = rnorm(10)
## ----test-b ----
mean(x)
sd(x)
```

ここで、ラベルとして、`## ----ラベル名 ----` となっていなければならない。

これを読み込ませるには、まず、

```
<<external-code, cache=FALSE, echo=FALSE>>=
library(knitr)
read_chunk('hoge.R')
@
```

とすればよい。チャンクオプションに `cache=FALSE` と指定しなければならない。`echo=FALSE` とすればインプットを隠すことができる。

そうして、本文中に、

```
<<test-a>>=
@
```

とすればよい。

#### 4.4.3 Beamer

knitr を組み込んだ Beamer よる PDF プレゼンファイルが作成可能であ

る。そのためにフレームオプションに **fragile** を付け加える必要がある。また出力が一つのスライドに収まらない場合、フレームオプションに **allowframebreaks** を付け加える必要がある。さら文書の最初のほうで以下の設定を付け加える。

```
<<setup, include=FALSE>>=
library(knitr)
# the kframe environment does not work with allowframebreaks,
so remove it
knit_hooks$set(document = function(x) {
  gsub('\\\\(begin|end)\\{kframe\\}', '', x)
})
@
```

**default** 以外のスライドテーマの場合、出力を分割するために **\frametitle** を加えないようにしないとイケない。

## 5 まとめ

以上、LyX の使い方を論文作成および報告スライド作成に特化して説明した。本の作成やそのための索引作成方法、ファイルを分割して挿入する方法は北浦 (2009) を参照されたい。LyX について数式が多いと動作が重くなることがあるのでファイル分割はしばしば有効である。

最後にまとめとして私の使い方を紹介する。ひとつのプロジェクトはひとつのフォルダに入れておく。たとえばフォルダ名を **project** とする。論文で用いる式展開を漏れなく記した計算ノートをも **project.lyx** として作成する。論文に必要な参考文献は Zotero で管理して **project.bib** として当該フォルダに出力する。なお **jecon.bst** もコピーして移しておく。

データ分析のためのファイルは **project.Rmd** に全ての計算を記録する。

ほかにも原データファイルがあり、データ整形が必要なときがある。そのときは `data.Rmd` を作成し、その出力を `project.csv` とする。大規模な計算結果の図表は pdf 形式や tex 形式で出力しておく。もしくはその計算を knitr をつかって `project.R` を出力する。

英語論文は `article.lyx` に日本語論文は `jarticle.lyx` とする。参考文献のために `project.bib` を、作図や作表のため `project.R` を読み込む。プレゼンファイルは英語の場合 `beamer.lyx` とし、日本語の場合 `jbeamer.lyx` とする。これらは LyX から pdf や tex ファイルとして出力することができる。

まとめると、`project` フォルダ内に `project.bib` と `jecon.bst` およびデータファイルを用意し、LyX で `project.lyx` を、RStudio で `project.Rmd` を研究ノートとして作成する。論文やプレゼンファイルは英語の場合 `article.lyx`, `beamer.lyx` で作成し、日本語の場合は `article.lyx`, `jbeamer.lyx` で作成する。

論文の基本は研究ノート `project.lyx` と `project.Rmd` であり、このファイルは常に最新の結果を反映しておく。それをもとに論文やプレゼンファイルを knitr を用いて出来る限り自動化する。このようにしておけば、コピーのミスもほとんどなく、再現可能論文が作成可能になるであろう。

論文の作成過程において過去の方法が望ましかったときがしばしばあり、またプレゼンファイルは報告時間によってスライドの数が変わるため複数のファイルを用意しようとしてどれが最新が混乱することもある。また共同で論文を作成するときには互いにどの箇所を修正したのかのやりとりが必要になる。こうしたことはソフトウェア開発で常識となりつつあるバージョン管理を使えば解決できる。次回ではこのバージョン管理について現在主流となっている Git について紹介する。

### 参考文献

- 北浦訓行 (2009) 『LyX入門』, 技術評論社, 東京。
- 小浪吉史 (2002) 『ドキュメントプロセッサ LYX入門』, テクノプレス, 東京。
- 藤田眞作 (2010) 『LATEX2 $\epsilon$  論文作法』, ピアソン桐原, 東京。
- 宮崎憲治 (2015a) 「学術研究のためのオープンソース・ソフトウェア (1) : XeLaTeX」, 『経済志林』, 第82巻, 第4号, 285-321頁, 3月。
- (2015b) 「学術研究のためのオープンソース・ソフトウェア (2) : BiBTeX と Zotero」, 『経済志林』, 第83巻, 第2号, 119-149頁, 11月。
- (2016) 「学術研究のためのオープンソース・ソフトウェア (4) : knitr」, 『経済志林』, 第83巻, 第4号, 137-168頁, 3月。