R, Sweave and Beamer

Christopher Lee

January 9, 2014

Table of Contents

Downloading packages

- R is found here:
  `http://cran.parentingamerica.com`
- For Windows, find either TeX Live or MiKTeX here:
  `http://www.tug.org/texlive/acquire-netinstall.html`
  `http://miktex.org/download`
- For Mac OSX, find MacTex here:
  `http://tug.org/mactex/`
- Beamer should already be preloaded. If not it may be found here:
  `https://bitbucket.org/rivanvx/beamer/wiki/Home`

Online Tutorials

See the following resources for introductory tutorials and examples for Beamer:

- http://www.math.umbc.edu/~rouben/beamer/
- http://www.informatik.uni-freiburg.de/~frank/ENG/latex-course/latex-course-3/latex-course-3_en.html
- http://users.stat.umn.edu/~sandy/courses/8801/handouts/03.Beamer/beamer.pdf
- http://www.wekaleamstudios.co.uk/supplementary-material/

Section 1

The Ggplot2 package

Ggplot2

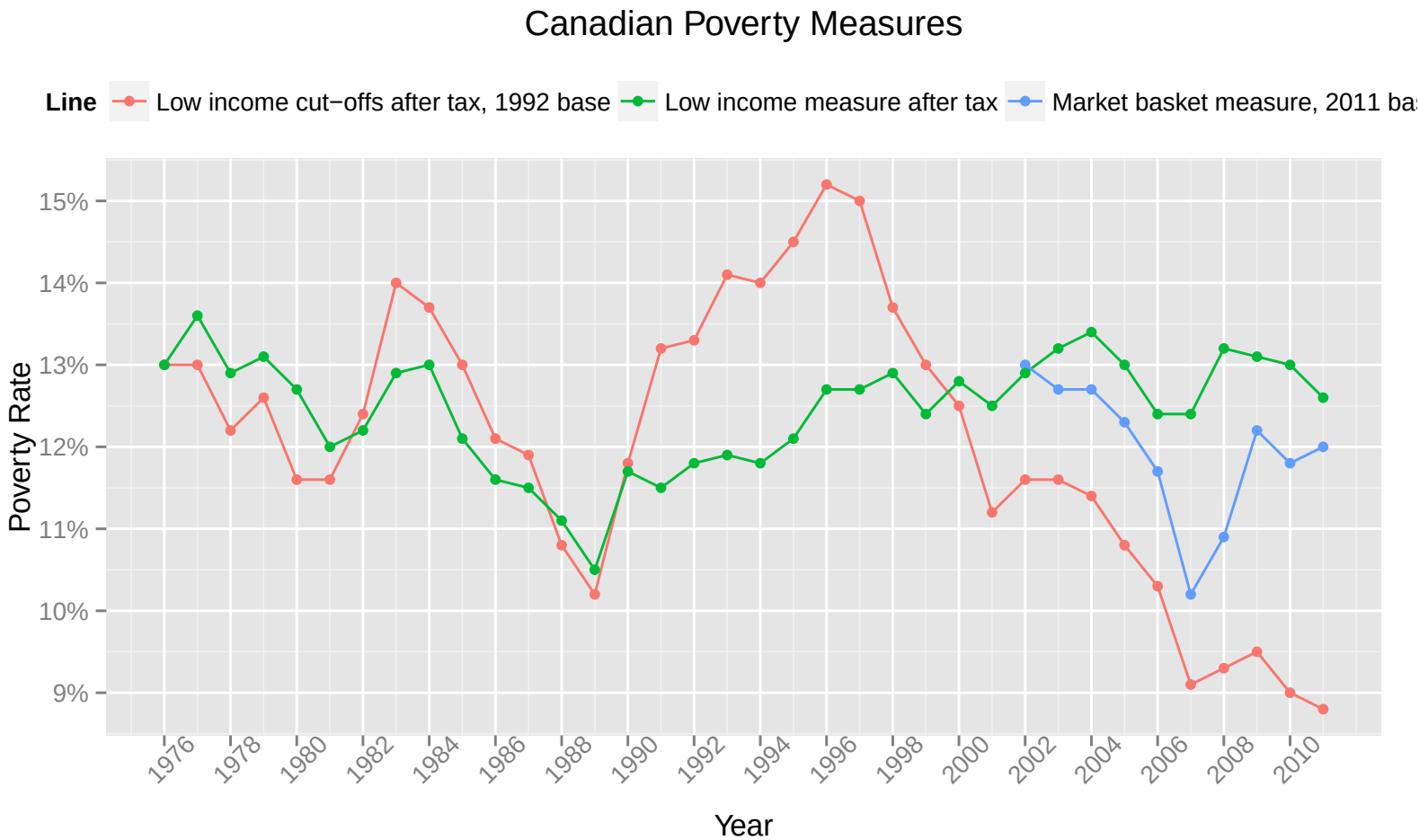
Data

I am using CANSIM 202-0802 data to produce these graphs in R version 3.0.1

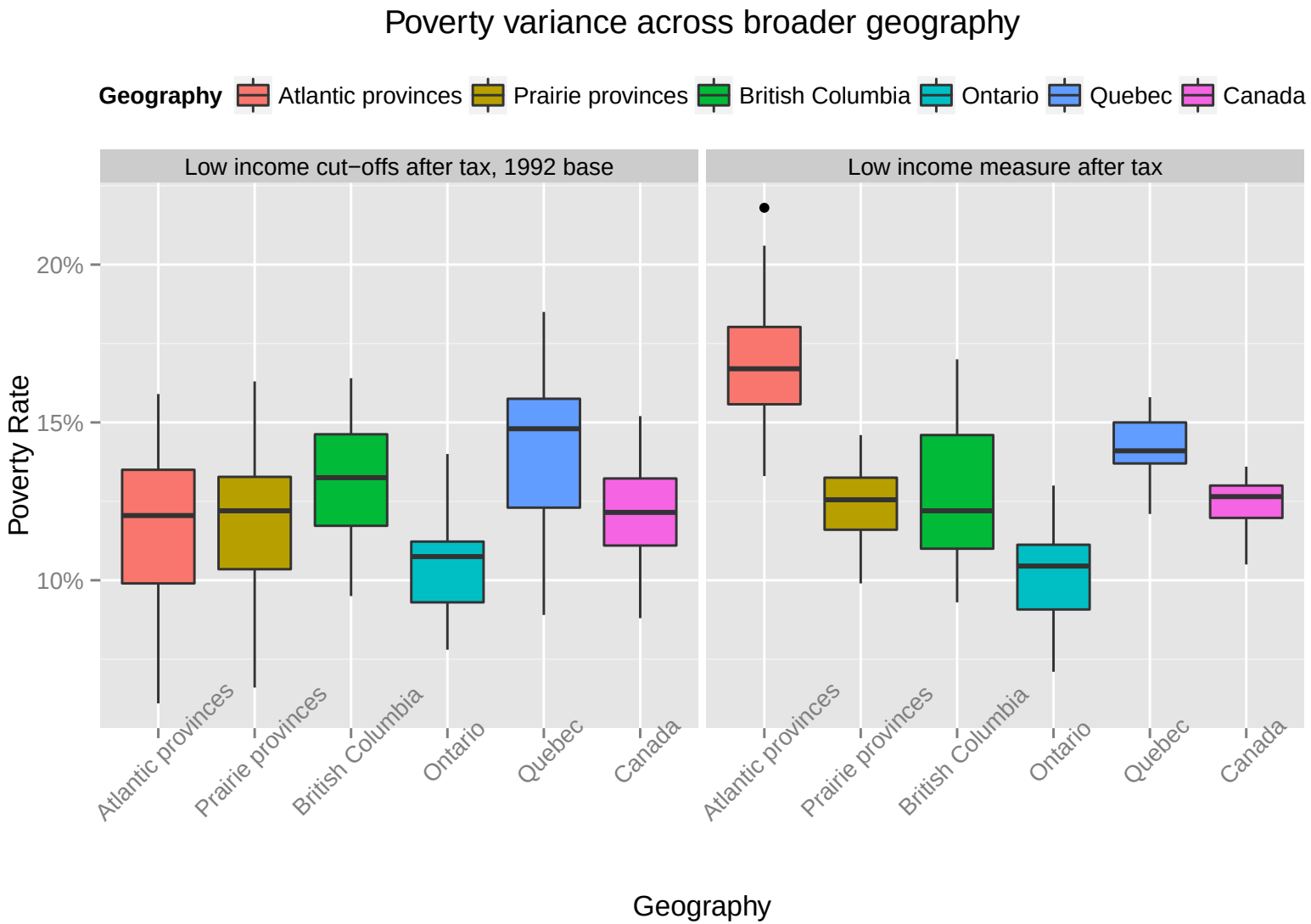
Find the data here:

`http://www.statcan.gc.ca/tables-tableaux/sum-som/l01/cst01/famil41a-eng.htm`

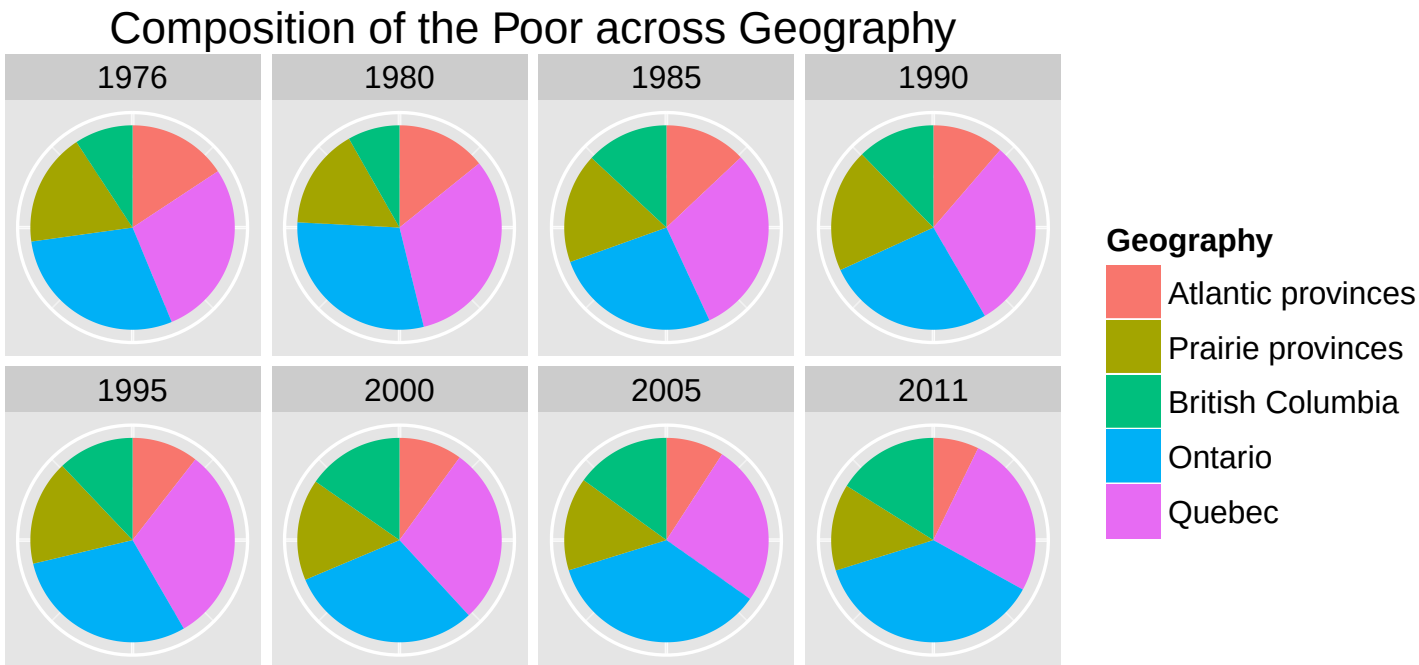
Example 1, Line graph



Example 2, Boxplots



Example 3 Pie Charts



GGplot code

All these plots start with the same code

```
ggplot(data=DATA,aes(x=X,y=Y,fill=FILL,
                                colour=COLOUR)
```

- DATA = self explanatory, our dataframe
- X = What we want to plot on the x-axis
- Y = What we want to plot on the y-axis
- FILL = For our 3rd variable to create sections/bins ontop of the x and y axes.
- COLOUR = Similar to FILL, it is created to map a third variable onto the plot.
- Notice, this alone is not sufficient to create a plot. We have specified the parameters, but now must specify what type of plot we want

Geom code

Geom line, Geom boxplot, Geom point, Geom bar

Here we pick which plots we want to use. Many arguments within the geom functions are very similar to the ones in the ggplot argument. Notice, if you have correctly specified your x, y and the third variable, you can leave `geom_()` blank

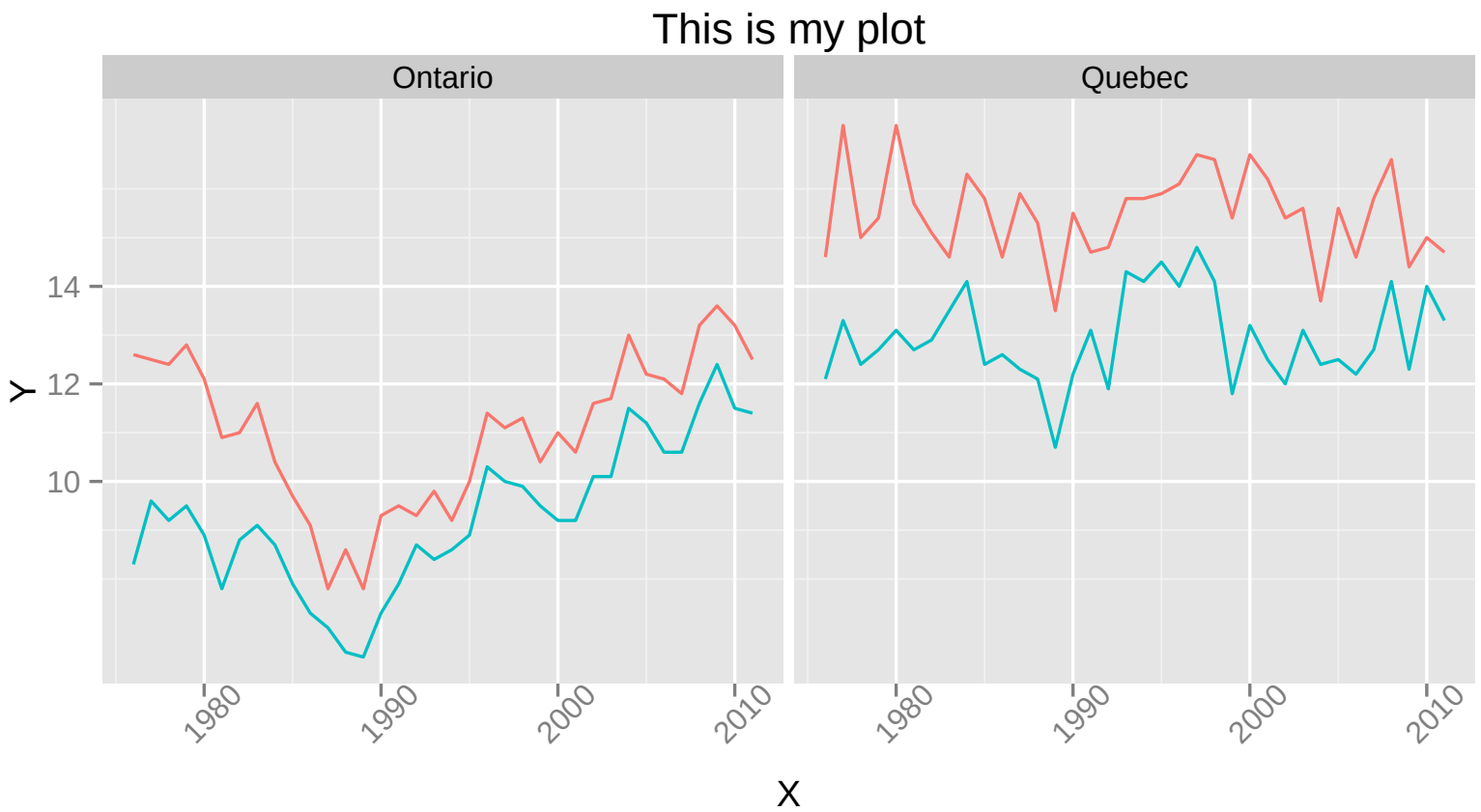
- `geom_line()` : Line graph
- `geom_boxplot()` : Side-by-side boxplots
- `geom_bar(stat='identity)` : Bar charts/ histogram
- `geom_area()` : Area plot

Miscallaneous code

Now with our plots we can tidy up the labels, axes, scales and legends

Imputs	Explanation
`ylab`	Title for y-axis
`xlab`	Title for x-axis
`ggtitle`	Main Title
`guides`	Controls legend
`theme`	Remove or angle ticks
`scale_y_continuous`	Control y-axis breaks and ticks
`scale_x_discrete`	Contrl x-axis breaks and ticks

Example output



Example code explanation

From our dataset DATA, we graph the variable *Year* on the x-axis and the variable *Value* on the y-axis. We assign a seperate color for every factor level of the variable *Population*.

```
ggplot(data=DATA,aes(x=Year,y=Value,colour=Population))+
```

We specify that we want a line graph, and title the plot.

```
geom_line()+gggtitle("This is my plot")
```

We title the y-axis **Y** and the x-axis **X**. We also suppress the legend.

```
ylab("Y")+xlab("X")+guides(fill=FALSE)+
```

We control the y-axis and create ticks on the numbers 10, 12 and 14 of the variable *Value*.

```
scale_y_continuous(breaks=c(10,12,14))+
```

We control the tick labels on the x-axis. We angle the tick labels 45 degrees

```
theme(axis.text.x=element_text(angle=45))+
```

We create several of these plots, each using a different factor level from the variable *Geography*. Essentially, each plot is using different data according to a unique value of *Geography*

```
facet_grid(.~Geography)
```

Result

The graphic is now ready to be placed into a LaTeX environment. No additional wrapper is necessary to bring this graphic from R into LaTeX and eventually to exportable .pdf. We will reuse this code later.

Other resources

For more information see the following links

- `http://www.cookbook-r.com/Graphs/`
- `http://sharpstatistics.co.uk/r/ggplot2-guide/`
- `http://www.ceb-institute.org/bbs/wp-content/uploads/2011/09/handout_ggplot2.pdf`

Exporting Tables to LaTeX environments

Xtable and Hmisc

Now what about tables?

We've seen how to create graphics and plots using ggplot within R. Now lets see how to create tables ready to be exported into LaTeX. There are two packages in R designed for this. To use either, imput the following two lines of code

1. Xtable

```
install.packages("xtable")
library(xtable)
```

2. Hmisc

```
install.packages("Hmisc")
library(Hmisc)
```

Exporting R tables as LaTeX tables

- Tables are only slightly more tricky than plots, in that there are several base R functions to summarize data and create tables in the console. But that output is not immediately exportable to LaTeX
- We look to Xtable and Hmisc, which allows us to wrap R tables within a function that translates it into LaTeX code.
- Exportable objects can range from contingency tables to matrices, regression summaries, ANOVA tables and even raw data

Xtable

List of base tables

R function	Description
`table()`	2-way contingency table
`xtabs()`	3-way contingency table
`prop.table()`	2-way frequency table
`aggregate()`	Custom aggregate data
`anova()`	Analysis of variance wrapper around a regression object

Xtable input

Take an R object of class dataframe, matrix, array, aov, lm... and then enter `xtable(x)` to provide LaTeX output. You are given a series of options for the output.

- Digits
- Line divisions
- Floating environment
- Captions
- Alignment of columns
- others

Wrapping xtable in print

Wrapping xtable in the print function is essential because it allows us to specify the LaTeX object to not float the table. A floated table is within a table environment and is sandwiched between `\begin{table}` and `\end{table}`

Once a table is floated there is very little we can do with it. The biggest problem is always **Overfull Hbox**. Tables are often too long to fit to the page width. Since Beamer is in a presentation format, its a bad idea to play around too much with the page size.

Sometimes we have Overfull Vbox where the table is too long, but this is easily corrected by specifying xtable to deliver output in ”longtable” format and splitting the table across multiple frames. This cannot be done with wide tables.

Wrapping xtable in print Cont...

We can shrink this "un-floated" table by first loading the **graphicx** package into LaTeX and then use the **resize** argument. It should look something like this.

```
\documentclass{beamer}
\usepackage{graphicx}
\begin{document}
\resizebox{\linewidth}{!}{%
<<echo=FALSE,warning=FALSE>> =
print(xtable(x),floating=FALSE)
@
}
```

Xtable example part 1

First lets create the table in R. It doesn't look too nice.

```
table<-with(subset(data2,data2$Year>=2000&data2$Geography=="Canada"&
data2$Statistic=="Percentage of persons in low income"&
as.character(data2$Line)=="Low income cut-offs after tax, 1992 base"),
xtabs(Value~Population+Year))
```

```
##                                    Year
## Population                          2000 2001 2002 2003 2004 2005 2006 2007
##   Females                           13.6 12.1 12.4 12.2 11.9 11.1 10.7  9.3
##   Males                             11.4 10.3 10.7 11.0 10.8 10.5 10.0  9.0
##   Persons under 18 years            13.9 12.2 12.4 12.7 13.0 11.7 11.1  9.5
##   Persons 18 to 64 years            12.9 11.7 12.0 12.2 11.9 11.4 11.1  9.9
##   Persons 65 years and over          7.6  6.7  7.6  6.8  5.6  6.2  5.3  4.8
##   Persons in economic families       9.3  8.1  8.6  8.7  8.2  7.5  7.1  6.0
##   Child in single mother families 40.1 37.4 43.0 41.4 40.4 32.9 31.2 26.7
##   Child in two-parent families       9.5  8.3  7.4  7.9  8.4  7.8  7.5  6.5
##   Unattached individuals            32.9 30.8 29.5 29.7 30.1 30.5 29.4 27.6
##   All persons                       12.5 11.2 11.6 11.6 11.4 10.8 10.3  9.1
##                                    Year
## Population                          2008 2009 2010 2011
##   Females                           9.8  9.5  9.3  8.9
##   Males                             8.9  9.5  8.7  8.7
##   Persons under 18 years            9.0  9.4  8.2  8.5
##   Persons 18 to 64 years           10.1 10.4 10.1  9.7
##   Persons 65 years and over         5.8  5.1  5.3  5.2
##   Persons in economic families      6.2  6.5  5.9  5.5
##   Child in single mother families 23.3 21.5 21.8 23.0
##   Child in two-parent families      6.4  7.3  5.7  5.9
##   Unattached individuals           27.3 26.9 26.9 27.7
##   All persons                       9.3  9.5  9.0  8.8
```

Xtable example part 2

Now lets wrap our xtab() table within xtable. Again this looks very ugly. We have forgotten to change some options in the chunk code.

```
table2<-
xtable(table,digits=1,align="l|rrrrrrrrrrrr")
```

```
## % latex table generated in R 3.0.1 by xtable 1.7
## % Thu Jan  9 00:09:20 2014
## \begin{table}[ht]
## \centering
## \begin{tabular}{l|rrrrrrrrrrrr}
##   \hline
##  & 2000 & 2001 & 2002 & 2003 & 2004 & 2005 & 200
##   \hline
## Females & 13.6 & 12.1 & 12.4 & 12.2 & 11.9 & 11.
##   Males & 11.4 & 10.3 & 10.7 & 11.0 & 10.8 & 10.
##   Persons under 18 years & 13.9 & 12.2 & 12.4 &
##   Persons 18 to 64 years & 12.9 & 11.7 & 12.0 &
```

Example part 3

We have to modify our R chunk code for LaTeX to properly read xtable. The table looks right now, except that it is far too large for the frame

```
<<echo=FALSE,warning=FALSE, results='asis'>> =
table2<-
xtable(table,digits=1,align="l|rrrrrrrrrrr")
@
```

	2000	2001	2002	2003	200
Females	13.6	12.1	12.4	12.2	11.
Males	11.4	10.3	10.7	11.0	10.
Persons under 18 years	13.9	12.2	12.4	12.7	13.
Persons 18 to 64 years	12.9	11.7	12.0	12.2	11.
Persons 65 years and over	7.6	6.7	7.6	6.8	5.
Persons in economic families	9.3	8.1	8.6	8.7	8.
Child in single mother families	40.1	37.4	43.0	41.4	40.
Child in two-parent families	9.5	8.3	7.4	7.9	8.
Unattached individuals	32.9	30.8	29.5	29.7	30

Example part 4

This shows our Overfull Hbox problem. To remedy this we wrap the xtable within print, specify floating=FALSE and resize the table

```
\resizebox{\linewidth}{!}{%
<<echo=FALSE,warning=FALSE,results='asis'>> =
print(table2,floating=FALSE)
@
}
```

	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
Females	13.6	12.1	12.4	12.2	11.9	11.1	10.7	9.3	9.8	9.5	9.3	8.9
Males	11.4	10.3	10.7	11.0	10.8	10.5	10.0	9.0	8.9	9.5	8.7	8.7
Persons under 18 years	13.9	12.2	12.4	12.7	13.0	11.7	11.1	9.5	9.0	9.4	8.2	8.5
Persons 18 to 64 years	12.9	11.7	12.0	12.2	11.9	11.4	11.1	9.9	10.1	10.4	10.1	9.7
Persons 65 years and over	7.6	6.7	7.6	6.8	5.6	6.2	5.3	4.8	5.8	5.1	5.3	5.2
Persons in economic families	9.3	8.1	8.6	8.7	8.2	7.5	7.1	6.0	6.2	6.5	5.9	5.5
Child in single mother families	40.1	37.4	43.0	41.4	40.4	32.9	31.2	26.7	23.3	21.5	21.8	23.0
Child in two-parent families	9.5	8.3	7.4	7.9	8.4	7.8	7.5	6.5	6.4	7.3	5.7	5.9
Unattached individuals	32.9	30.8	29.5	29.7	30.1	30.5	29.4	27.6	27.3	26.9	26.9	27.7
All persons	12.5	11.2	11.6	11.6	11.4	10.8	10.3	9.1	9.3	9.5	9.0	8.8

Floating

Remember, the **resize** command will only work on a ”non-floating” object. Therefore if you have an Overfull hbox, you must first wrap your table in print() and specify floating=FALSE.

I’ll go over the R chunk code and Sweave code in the next section.

A simpler solution: Xtable font size

Sometimes the resize command is unecessary and a simpler solution is available. Xtable allows for us to change the font size from normal size to very small to very large

xtable(...,size=" ")

By changing the font size of the table, we can easily remedy minor Overfull Hbox and Vbox. There are 10 sizes.

- tiny
- scriptsize
- footnotesize
- small
- normal
- large
- Large
- LARGE
- huge
- Huge

Dressing up a table

Lets say we have a properly sized vanilla table

	LICO	LIM	MBM
1995	14.50	12.10	
1996	15.20	12.70	
1997	15.00	12.70	
1998	13.70	12.90	
1999	13.00	12.40	
2000	12.50	12.80	
2001	11.20	12.50	
2002	11.60	12.90	13.00
2003	11.60	13.20	12.70
2004	11.40	13.40	12.70
2005	10.80	13.00	12.30
2011	8.80	12.60	12.00

We will try to add a few features to the table:

- Percentage signs
- Captions
- Line divisions
- A little colour

To use the following code add `\usepackage{colortbl}` to the preamble and `library(stringr)` to the chunk code

Dressing up a table: Result

	LICO	LIM	MBM
1995	14.5%	12.1%	
1996	15.2%	12.7%	
1997	15.0%	12.7%	
1998	13.7%	12.9%	
1999	13.0%	12.4%	
2000	12.5%	12.8%	
2001	11.2%	12.5%	
2002	11.6%	12.9%	13.0%
2003	11.6%	13.2%	12.7%
2004	11.4%	13.4%	12.7%
2005	10.8%	13.0%	12.3%
2011	8.8%	12.6%	12.0%

Table : Poverty Lines

```
tab3<-apply(tab2, 2, func)
func<-function(u){
  ifelse(!is.na(u),
  sprintf("%.1f%%",u),u)}
pos<-as.list(seq(1,
 nrow(tab3),by=2))
com<-rep(
  "\\rowcolor[gray]{.9}",
 length(pos))
print(xtable(tab3,
    caption="Poverty Lines",
    align="l||c|c|c",
    size="small"),
   hline.after=
    c(-1,0,11,nrow(tab2)),
   add.to.row=list(
    pos=c(list(-1,8),pos),
    command=c(
     "\\rowcolor[rgb]{1,.8,.8}"
     "\\rowcolor[rgb]{.8,1,0}",
     com)))
```

Dressing up a table: Explanation

First we define a function that will paste a % on every non-NA element

```
tab3<-apply(tab2, 2, function(u)
  ifelse(!is.na(u),sprintf( "%.1f%%", u),u))
```

Then we need to specify the rows in the table which are alternating

```
pos<-as.list(seq(1,nrow(tab3),by=2))
```

Then we pick the color for the alternating rows defined by `pos`

```
com<-rep("\\rowcolor[gray]{.9}",length(pos))
```

Now call xtable and use the caption option to define your caption

```
print(xtable(tab3,caption="Poverty Lines",
```

Create vertical lines between certain columns

```
align="l||c|c|c",
```

Also specify a different font size for the table and close xtable options

```
size="small"),
```

Dressing up a table: Explanation cont.

Now in the print options, create horizontal lines to divide key rows

```
hline.after=c(-1,0,11,nrow(tab2)),
```

Select the header, another row and our alternating rows

```
add.to.row=list(pos=c(list(-1,8),pos),
```

Now lets select colors for our header, another row and alternating rows

```
command=c("\\rowcolor[rgb]{1,.8,.8}",
  "\\rowcolor[rgb]{.8,1,0}",
  com)))
```

Now we are finished.

Hmisc

The Hmisc equivalent of the `xtable()` wrapper is the `latex()` function. It operates in a similar manner to xtable. There are many advanced features in Hmisc that are unavailable to xtable such as multi-line headers, very simple cell selection plus all the functionality of xtable. Hmisc output generally looks cleaner than xtable output. With our last example table: `tab2`, we can write a basic latex table with Hmisc using the following code.

```
latex(tab2, file= '')
```

`tab2` is the object we want wrapped, and `file= ''` stops Hmisc from creating a separate output postcript file. In other words, it means we will get standard latex output.

Hmisc and Floating

Like with Xtable, we may run into scenarios where we have Overfull Hbox. In that case we can resize our R table only if our table is non-floating. it In order to specify the output of `latex()` to be a non-floating object, we write:

```
latex(...,table.env=FALSE,center=FALSE)
```

So instead of your last 3 lines of code looking like

```
\end{tabular}
\end{center}
\end{table}
```

You are simply left with

```
\end{tabular}
```

Now we can squeeze in a resize for this tabular environment, then center and table it later if we want. Like with xtable, these two options are necessary if you need to tinker or resize the table.

Hmisc (latex) inputs: Useful options

- file: *See slides above*
- table.env: *See slides above*
- center: *See slides above*
- size: *Same as xtable*
- vbar: *Vertical line divisions*
- rowname: *Supress row names*
- extracolheads: *Add sub-headers*
- cellTexCmds *Format table cells*
- booktabs: *Different table style*
- ctable: *Different table style*

Hmisc (latex) inputs: Group commands

Column Groups

- cgroup: *Name your column groups*
- n.cgroup: *Select your column groups*
- cgroupTexCmd: *Format your column groups*
- cgroup.just: *Align your column groups*

Row Groups

- rgroup: *Name your row groups*
- n.rgroup: *Select your row groups*
- rgroupTexCmd: *Format your row groups*
- rowlabel: *Simple header for row names*

Hmisc example part 1

Lets start with some subsetted data. Here are the first couple rows.

```
##   Year Geography                                        Line
## 1 2005    Quebec Low income cut-offs after tax, 1992 base
## 2 2006    Quebec Low income cut-offs after tax, 1992 base
## 3 2007    Quebec Low income cut-offs after tax, 1992 base
## 4 2008    Quebec Low income cut-offs after tax, 1992 base
## 5 2009    Quebec Low income cut-offs after tax, 1992 base
## 6 2010    Quebec Low income cut-offs after tax, 1992 base
##                                  Statistic  Population Value
## 1 Number of persons in low income (x 1,000) All persons   870
## 2 Number of persons in low income (x 1,000) All persons   828
## 3 Number of persons in low income (x 1,000) All persons   784
## 4 Number of persons in low income (x 1,000) All persons   828
## 5 Number of persons in low income (x 1,000) All persons   686
## 6 Number of persons in low income (x 1,000) All persons   778
```

Hmisc example part 2

The data is in long format so we will call *cast* from the *reshape* package in order to display the data how we want

```
##    Geography Year LICO Persons in low income LICO Pov. rate
## 1    Ontario 2005                       1276           10.3
## 2    Ontario 2006                       1283           10.3
## 3    Ontario 2007                       1111            8.8
## 4    Ontario 2008                       1187            9.3
## 5    Ontario 2009                       1306           10.1
## 6    Ontario 2010                       1153            8.8
## 7    Ontario 2011                       1182            9.0
## 8     Quebec 2005                        870           11.7
## 9     Quebec 2006                        828           11.1
## 10    Quebec 2007                        784           10.4
## 11    Quebec 2008                        828           10.9
## 12    Quebec 2009                        686            8.9
## 13    Quebec 2010                        778           10.0
## 14    Quebec 2011                        745            9.5
##    LICO Pov. rate LIM Persons in low income LIM Pov. rate
## 1            10.3                      1452          11.7
## 2            10.3                      1418          11.3
## 3             8.8                      1418          11.2
## 4             9.3                      1588          12.4
## 5            10.1                      1681          13.0
## 6             8.8                      1608          12.3
## 7             9.0                      1576          12.0
## 8            11.7                      1048          14.1
## 9            11.1                      1006          13.4
## 10           10.4                      1076          14.3
## 11           10.9                      1172          15.4
## 12            8.9                      1030          13.4
## 13           10.0                      1129          14.5
## 14            9.5                      1095          14.0
```

Hmisc example part 3

For the sake of comparison, lets first create the table in Xtable

Geography	Year	LICO Persons in low income	LICO Pov. rate	LIM Persons in low income	LIM Pov. rate
Ontario	2005	1276.0	10.3	1452.0	11.7
Ontario	2006	1283.0	10.3	1418.0	11.3
Ontario	2007	1111.0	8.8	1418.0	11.2
Ontario	2008	1187.0	9.3	1588.0	12.4
Ontario	2009	1306.0	10.1	1681.0	13.0
Ontario	2010	1153.0	8.8	1608.0	12.3
Ontario	2011	1182.0	9.0	1576.0	12.0
Quebec	2005	870.0	11.7	1048.0	14.1
Quebec	2006	828.0	11.1	1006.0	13.4
Quebec	2007	784.0	10.4	1076.0	14.3
Quebec	2008	828.0	10.9	1172.0	15.4
Quebec	2009	686.0	8.9	1030.0	13.4
Quebec	2010	778.0	10.0	1129.0	14.5
Quebec	2011	745.0	9.5	1095.0	14.0

Table : Using Xtable

This looks adequate, but lets see if Hmisc can do better...

Hmisc example part 4

	Year	LICO		LIM	
		Number of persons in low income (x 1,000)	Poverty rate %	Number of persons in low income (x 1,000)	Poverty rate %
Ontario					
1	2005	1276	10.3	1452	11.7
2	2006	1283	10.3	1418	11.3
3	2007	1111	8.8	1418	11.2
4	2008	1187	9.3	1588	12.4
5	2009	1306	10.1	1681	13.0
6	2010	1153	8.8	1608	12.3
7	2011	1182	9.0	1576	12.0
Quebec					
8	2005	870	11.7	1048	14.1
9	2006	828	11.1	1006	13.4
10	2007	784	10.4	1076	14.3
11	2008	828	10.9	1172	15.4
12	2009	686	8.9	1030	13.4
13	2010	778	10.0	1129	14.5
14	2011	745	9.5	1095	14.0

Table : Using Hmisc

This looks a bit nicer I think. Lets go over the code in detail

Hmisc example: code

First, lets cast the data, drop *Geography* and rename headers.

```
tab6<-cast(tab4,Geography+Year~Line+Statistic,
  value="Value")
tab6<-tab6[,-1]
colnames(tab6)<-
c("Year","Number of persons in low income",
  "Poverty rate","Number of persons in low income",
  "Poverty rate")
```

Now lets wrap our *tab6* in Hmisc's latex function and avoid any unecessary file creation. Lets also set a font size.

```
latex(tab6,file='',size ='tiny',
```

Add extra headers beneath our new column names.

```
extracolheads=c('','(x 1,000)','\\%',
'(x 1,000)','\\%'),
```

Hmisc example: code cont...

Now lets look at multi-line headers. We want the first 7 rows in one group and the next 7 in another. Then name the respective groups. We want the first row, then the next 2 and the next 2 in groups so name those as well

```
n.rgroup=c(7,7),rgroup=c('Ontario','Quebec'),
n.cgroup=c(2,2),cgroup=c('','LICO','LIM'),
```

Remove any header for the rownames and center all the columns

```
rowlabel='',col.just=c('c','c','c','c','c'),
```

Finally, and most importantly, remove the float (ie. remove `\begin{table} \begin{center}` and the ends as well) so we can resize it later.

```
table.env=FALSE,center='none')
```

Section 3

Using Sweave and Knitr

Sweave and Knitr

Now knowing how to create plots and tables in a LaTeX format, how exacty do we get our R code into LaTeX in the first place? The generic R script file will not work. To get started

- Open R
- Instead of a script file, open up an R Sweave file with extension .Rnw
- Go to Tools $\Rightarrow$ Options $\Rightarrow$ Sweave
- Select ”Weave Rnw files using” $\Rightarrow$ knitr

You must have LaTeX in order to run this type of file. We will use **knitr** to move R code into LaTeX code, rather than the default **Sweave** which has a few annoying problems

Sweave layout

In a .Rnw file, you must use LaTeX code. To use R code instead, you must create a ”Chunk”.

- To begin the chunk, type:

  <<>>=

- To end the chunk, type:

  @

On your screen it should look like this:

```
\documentclass{article}
\begin{document}

LATEX CODE GOES HERE

<<>>=

R CODE GOES HERE

@

\end{document}
```

R to Sweave

- Within the chunk, the R code is exactly the same as if you were writing R code in a script file.
- It is better to create a chunk for every table or plot, rather than putting all your R code and figures/tables in the same chunk.
- Plots are almost never sized correctly to the page. You will have to tweak with the chunk code in order to display plots with your desired dimensions.

Chunk Code

You can control the output of the R code within your chunk.

```
<< options go here >>=
```

A couple options:

- echo — Prints the R code in the output
- warnings — Prints the warnings in the output
- results — Whether to display the chunk output
- include — How to display the chunk output
- size — Font size of chunk output
- message — Prints messages in the output
- fig.height — Change height of figure
- fig.width — Change width of figure

Chunk code example


```
<<echo=FALSE,warning=FALSE,message=FALSE,
                    fig.height=4,size='tiny'>>=
```

- The actual R chunk code is not displayed
- Warnings from the R chunk code are not displayed
- Messages from the R chunk code are not displayed
- The figure in the chunk has a height of 4 inches
- The font size for the chunk output is tiny

Finally, compiling the .pdf

- After you have set up your R chunk with LaTeX friendly plots and tables, the very last step is to click **Compile PDF**
- This will create several files, a .log file, a .tex file, several auxilary files and the .pdf file.
- Any plot output from your R chunks will also be saved in a folder (if unspecified) called **figure** in whichever directory you are working from.

Section 3

Beamer and LaTeX

LaTeX Preamble

Here's a very basic intro to LaTeX for the purposes of Beamer.

1. Define a document class. We want beamer
2. Run any packages you require for LaTeX
3. Choose your desired beamer color and theme
4. Create a title
5. Create an author
6. Run any other beamer options

LaTeX Preamble example

1. `\documentclass{beamer}`
2. `\usepackage{graphicx}`
3. `\usetheme{PaloAlto}`
4. `\title{This is my title}`
5. `\author{This is my name}`
6. `\setbeamertemplate{itemize items}[ball]`

Title and ToC

Now that the preamble is finished, lets get onto the code:

1. `\begin{document}`
2. `\frame{\titlepage}`
3. `\tableofcontents`

- No. 2 provides us a title page which displays some of the things in our preamble
- No. 3 provides us a table of contents which will list the sections and subsections of the document.

The frame

In beamer, you will create your presentation frame by frame.

1. Start a frame environment with `\begin{frame}`
2. Create a frame title with `\frametitle{My frame title}`
3. Now you are within the frame. You can then write anything and it will display within the frame.
4. Finally, close the frame environment with `\end{frame}`

The frame Cont...

We can dress up our frame a bit.

1. Start a block with `\begin{block}{My block title}`
2. Write anything in the block.
3. Maybe you want bulletpoints with `\begin{itemize}`
4. Now within the itemize environment, write items
   ```
   \item This is my 1st bulletpoint
   \item This is my 2nd bulletpoint
   ```
5. Now close the itemize environment with `\end{itemize}`
6. Now close the block environment with `\end{block}`

The following frame is our end result

My frame title

Now you are within the frame. You can then write anything and it will display within the frame.

My block title

Write anything in the block.

- This is my 1st bulletpoint
- This is my 2nd bulletpoint

More frames

What if we want to section off the frame into several parts?

1. First type `\begin{columns}[T]` to vertically align the sections
2. Now specify the first section dimensions with `\begin{column}{width}` ie. `\begin{columns}{.5\textwidth}`
3. With this section, you can write, itemize, do anything
4. Now close this section with `\end{column}`
5. Now specify the second section with `\begin{column}{width}`. ie. `\begin{columns}{.5\textwidth}`
6. Try writing here as well
7. Now close this section with `\end{column}`
8. Then close the environment with `\end{columns}`

More frames example

With this section, you can write, itemize, do anything

Try writing here as well

Pausing in Frames

You can use `\pause` in order to show the contents of your whole frame in several seperate frames.

1. Write something
2. Now write `\pause`
3. Now write something else
4. Now write `\pause` again
5. Write something here too

Pausing example part 1

- Write something

Pausing example part 2

- Write something
- Now write something else

Pausing example part 3

- Write something
- Now write something else
- Now write something here too

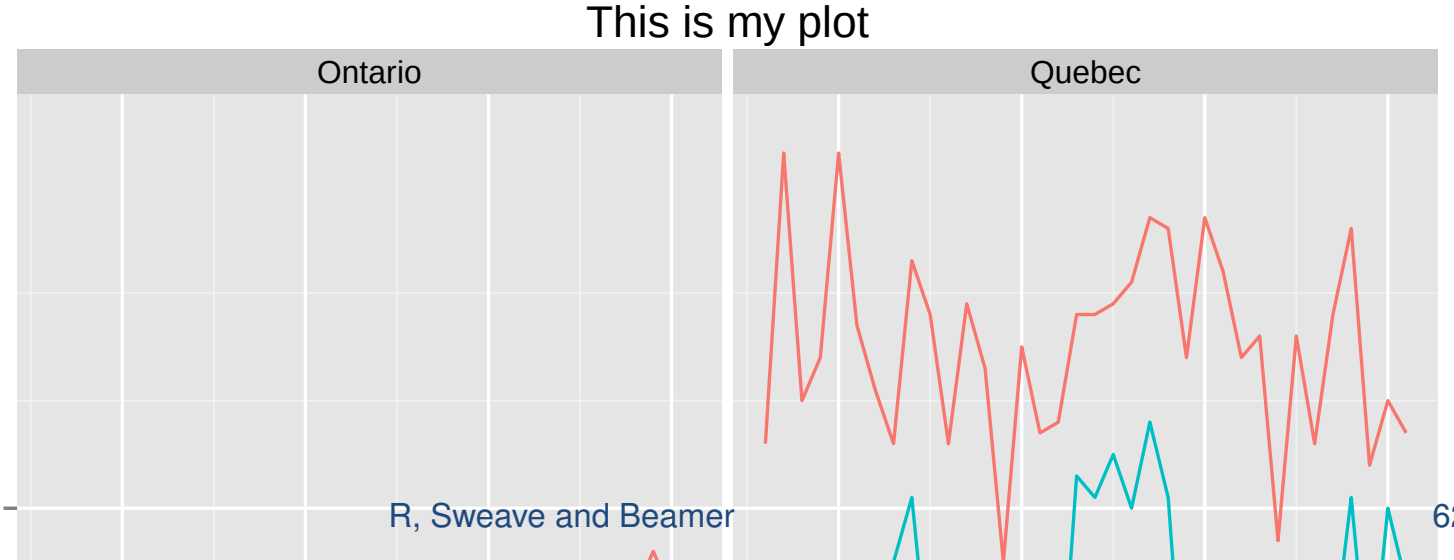
Displaying R output in frame

Within the frame environment, lets put in a plot that we previously made.

```
\begin{frame}
\frametitle{R plot example}
<<echo=FALSE,warning=FALSE>>
INSERT GGPLOT CODE
@
\end{frame}
```

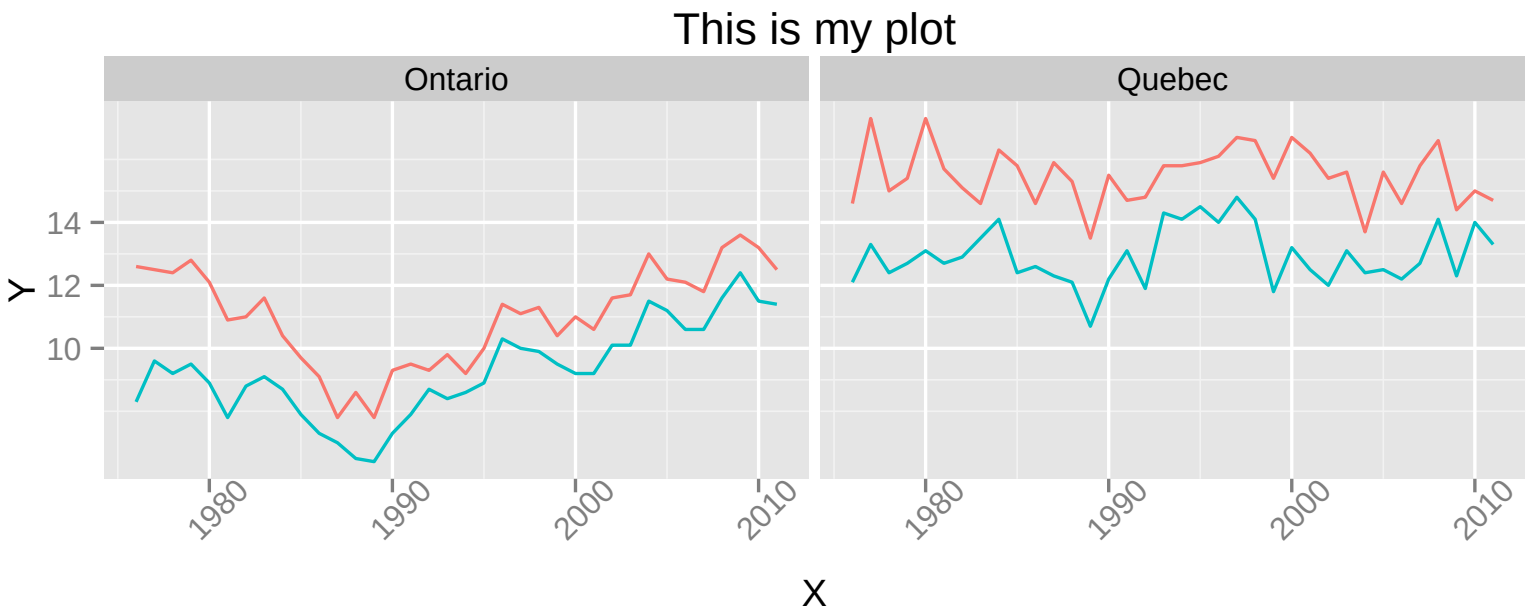
R plot example


```
ggplot(subset(data2, (Geography == "Ontario" |
    "Low income measure after tax" & Statistic
    (Population == "Males" | Population == "Fem
    colour = Population)) + geom_line() + ggtit
    xlab("X") + guides(fill = FALSE) + scale_y_
    14)) + theme(legend.position = "none", axis
    facet_grid(. ~ Geography)
```



R plot example Cont...

The last plot is not sized correctly. Playing with the fig.width and fig.height in the chunk options, we can get...



Inserting images

- It is very easy to insert images into your frames. To do this, you must load the **graphicx** package in the preamble.
- Within a frame environment, use
  `\includegraphics[options here]{file here}`
- Ie. `\includegraphics[width=4in]{Picture.pdf}`
  This gives us the file "Picture.pdf" with a width of 4 inches
- Alternatively you can use **scalebox** to shrink a picture to fit the page with
  `\scalebox{x}{\includegraphics{Picture.pdf}}`
- Ie. `\scalebox{.5}{\includegraphics{Picture.pdf}}`
  This gives us the file "Picture.pdf" but shrunk by half
- The graphic can be in many formats but it is recommended to use .png, .pdf, .eps or .jpg files

Small tips: Font size and verbatim


```
\tiny, \scriptsize, \footnotesize, \small,
\normalsize, \large, \Large, \LARGE,
\huge & \Huge
```

These are used to control the font size of the frame. Alternatively you can wrap these commands in `{ }` if you only want to resize a specific line. Ie. `{\small this is small text}`

Sometimes you want to make sure your text isn't misinterpreted as runnable code. To do this, use the verbatim environment `\begin{verbatim}` and `\end{verbatim}`. **However**, when using verbatim, you must specify the frame as *fragile*:

```
\begin{frame}[fragile]
```

Small tips: supressed captions

This pertains specifically for article class and not beamer class...Your table captions will be preceded by **Table 1:** or **Table 2:** or whichever table number you are on. You can suppress this by manually writing your caption in latex.

Following a non-floated table (make sure you don't see a `\begin{table}` or `\end{table}` anywhere!), add `\caption{My caption title}` after the `\end{tabular}` commmand. To suppress the numbering, write `\usepackage{caption}` in the preamble and instead of `\caption{My caption title}`, write `\caption*{My caption title}`

Why are things going wrong?

- Sometimes your .Rnw will fail to compile. The Log is sometimes not very helpful or specific. So we should go over a few mistakes
- Sometimes you forget to close an environment. Every line of code: `\begin{}` **must** be closed by `\end{}` and in correct order.
- If you have `\begin{document}`, `\begin{frame}`, `\begin{block}`, `\begin{itemize}`, the code must end in the corresponding order" `\end{itemize}`, `\end{block}`, `\end{frame}`, `\end{document}`
- I recommend testing each frame after completion

Anything Else?

I am not an expert in either LaTeX or Beamer. There are many more features to Ggplot, Sweave and Beamer. Bibtex? Framezoom? Animated graphics? More detail into Hmisc?

Additional resources

- `http://en.wikibooks.org/wiki/LaTeX/Presentations#Columns`
- `http://www.nyu.edu/projects/politicsdatalab/latex/beamer_nyu.pdf`