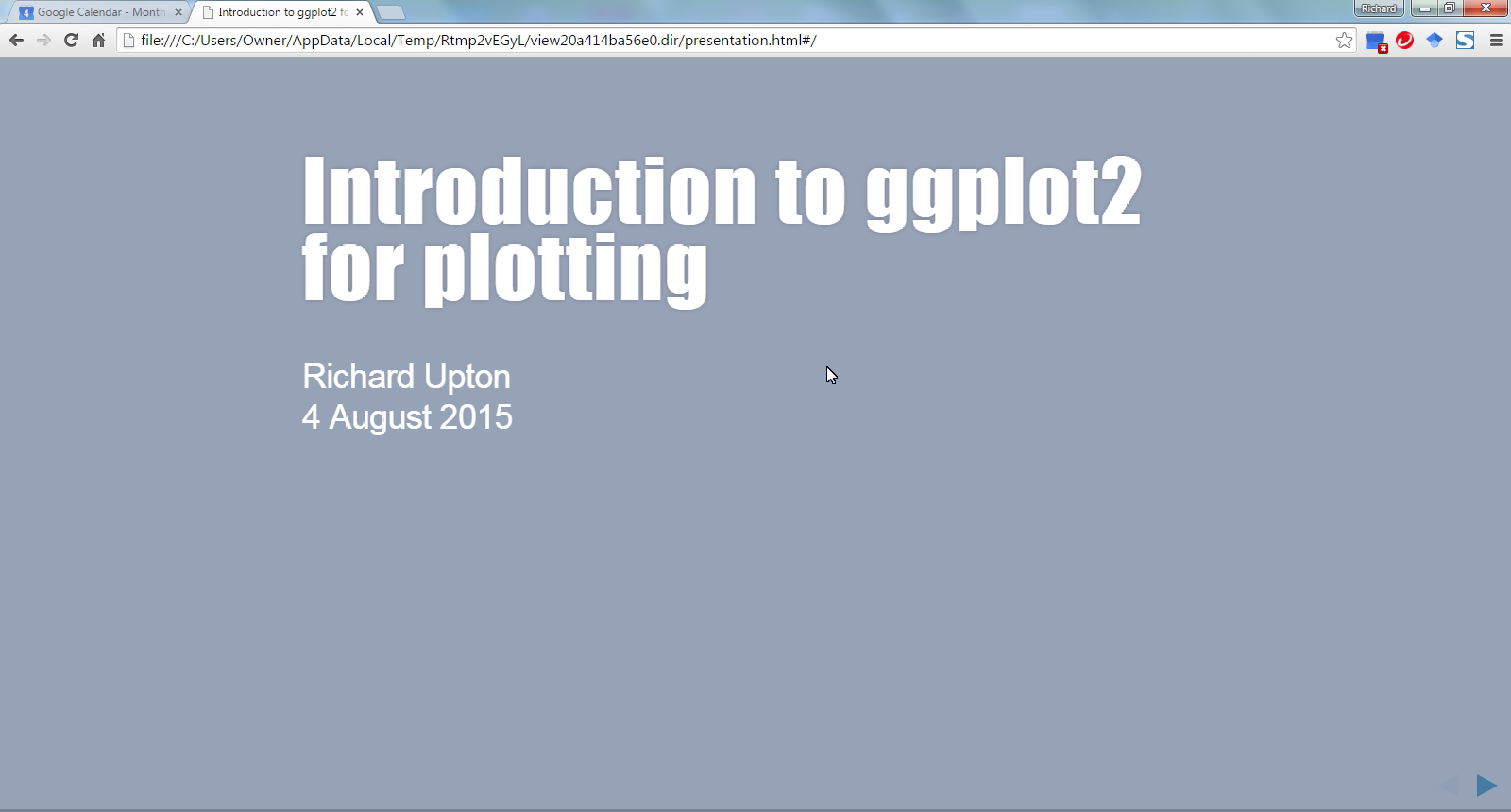
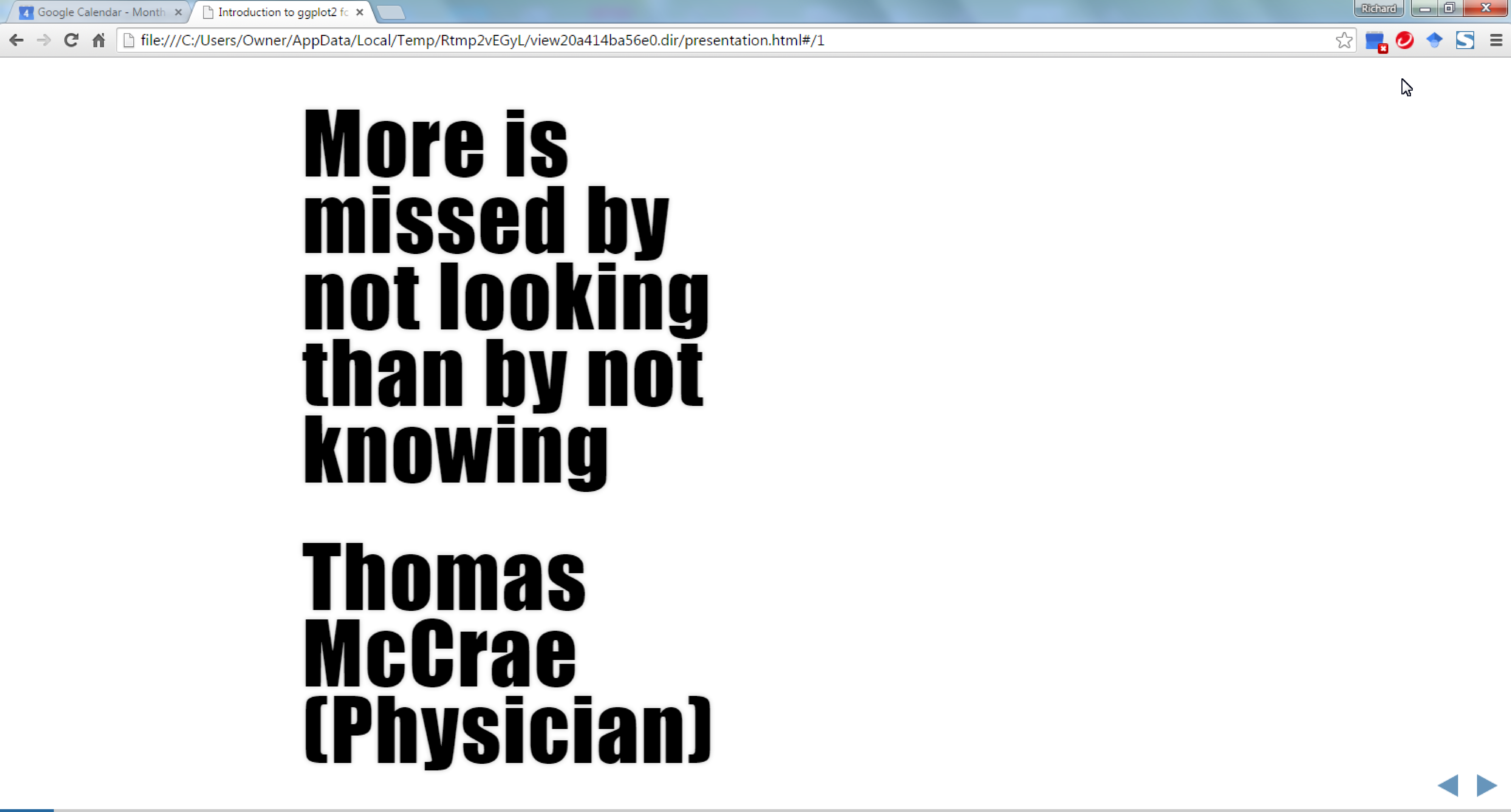

Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/
Richard

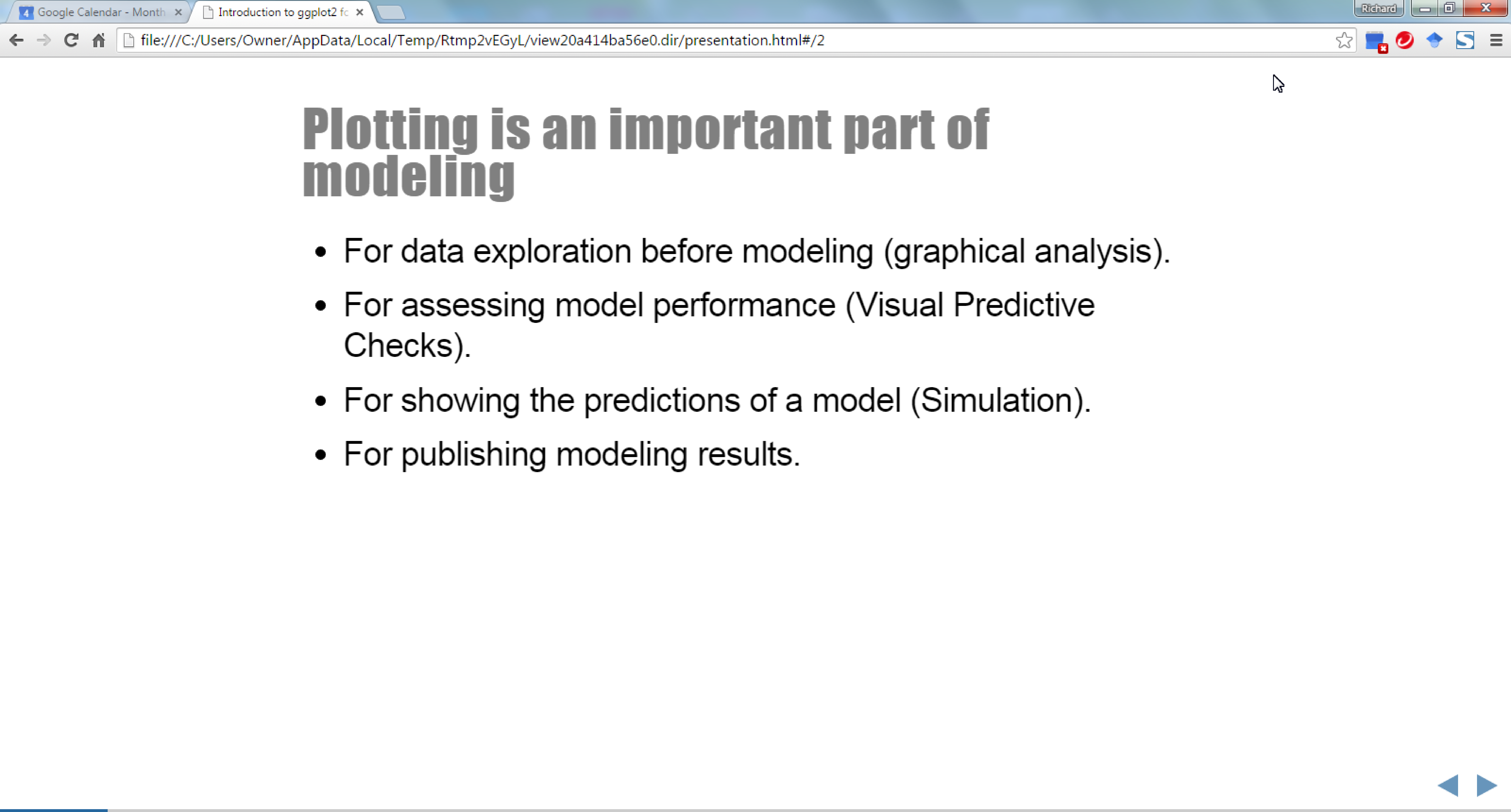
Introduction to ggplot2 for plotting

Richard Upton
4 August 2015


Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/1
Richard

More is missed by not looking than by not knowing

Thomas McCrae (Physician)


Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/2
Richard

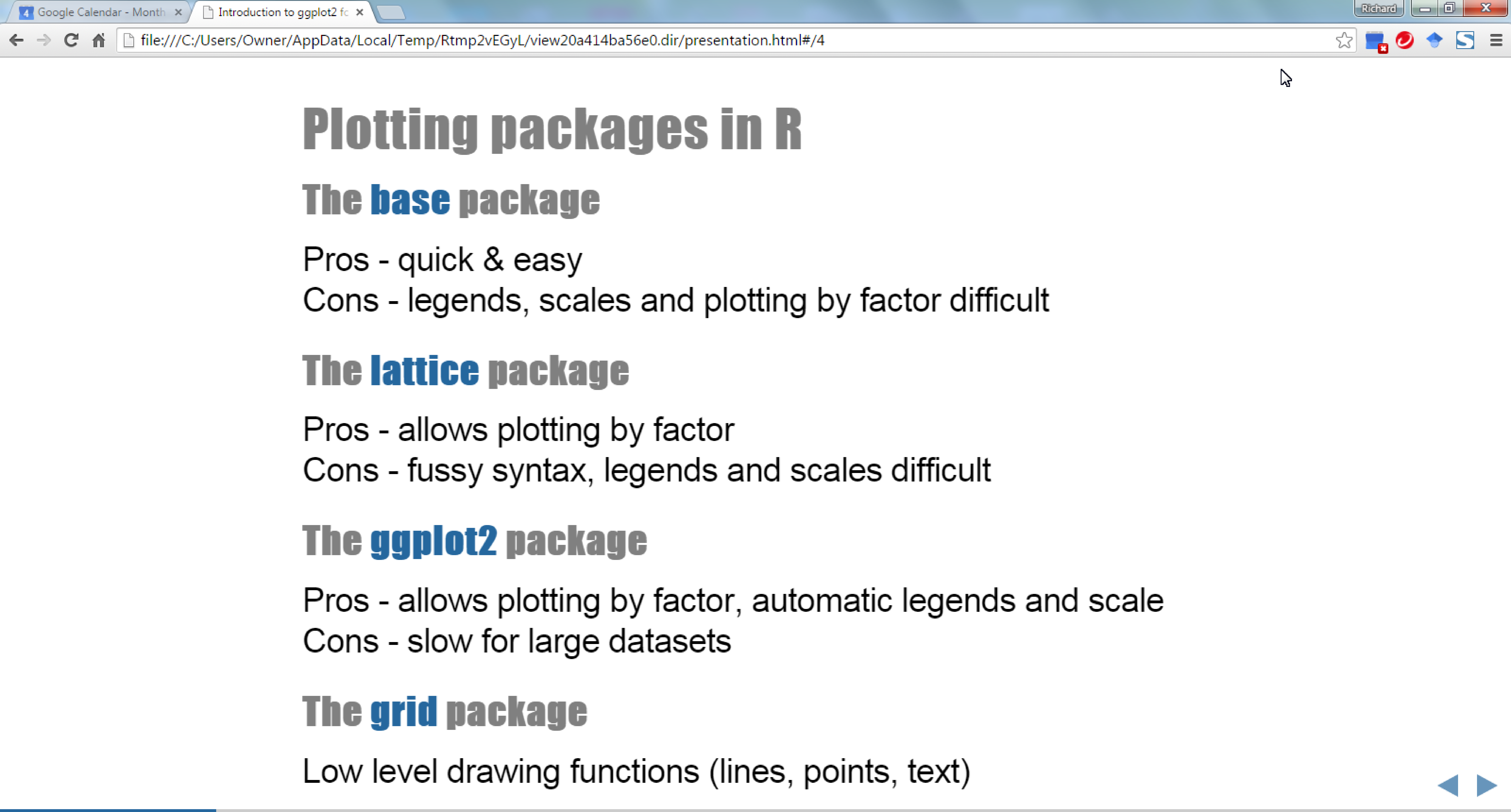
Plotting is an important part of modeling

- For data exploration before modeling (graphical analysis).
- For assessing model performance (Visual Predictive Checks).
- For showing the predictions of a model (Simulation).
- For publishing modeling results.

Google Calendar - Month
Introduction to ggplot2 fc
Richard
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/3
R OpenSci

Why use R for plotting?

- Of the statistical software environments, it's the most versatile.
- It's combination of data manipulation and plotting makes it a “one stop shop”.
- It facilliates Reproducible Research and Open Science.


Google Calendar - Month
Introduction to ggplot2 fc
Richard
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/4

Plotting packages in R

The base package

Pros - quick & easy
Cons - legends, scales and plotting by factor difficult

The lattice package

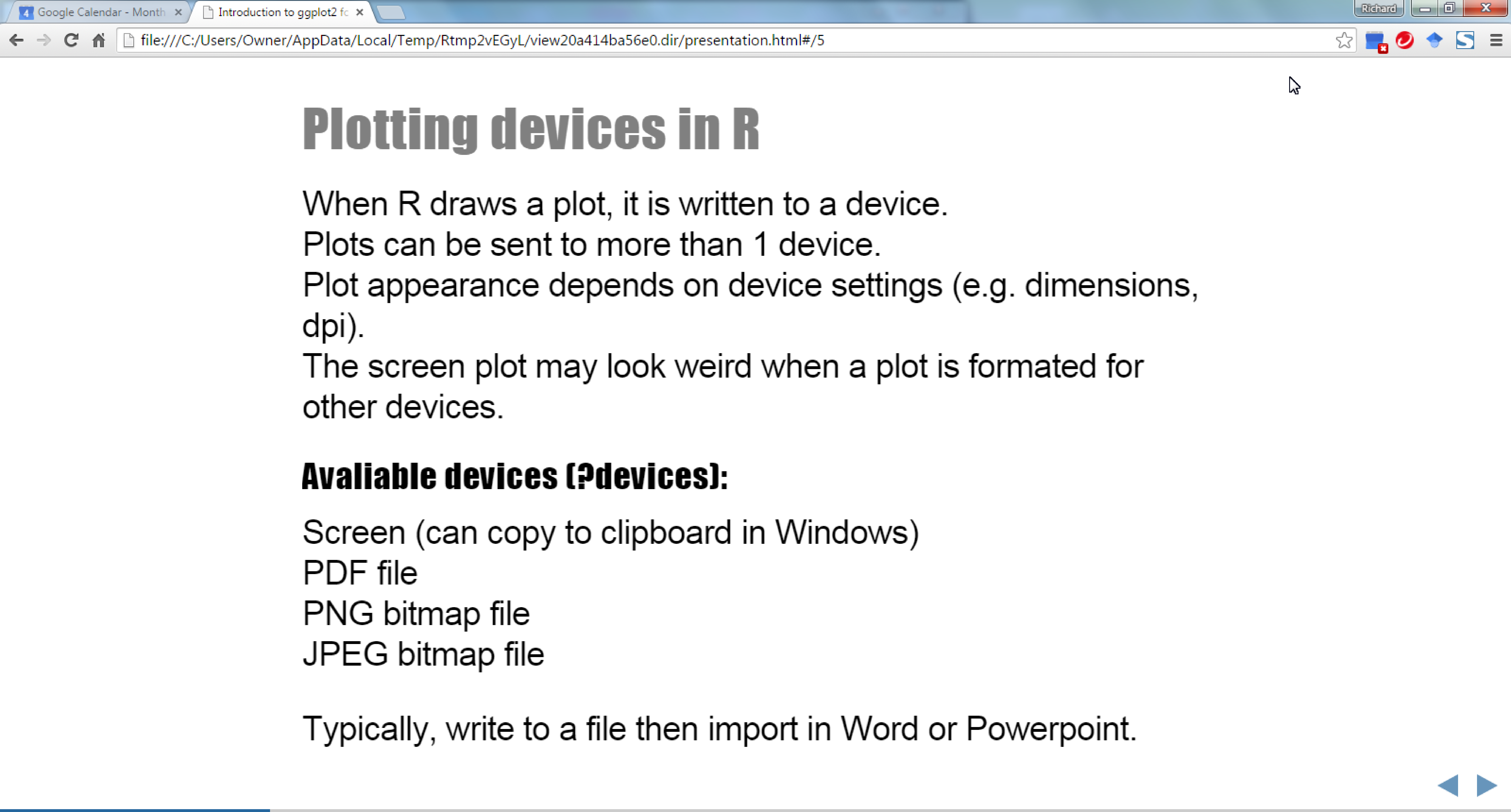
Pros - allows plotting by factor
Cons - fussy syntax, legends and scales difficult

The ggplot2 package

Pros - allows plotting by factor, automatic legends and scale
Cons - slow for large datasets

The grid package

Low level drawing functions (lines, points, text)


Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/5
Richard

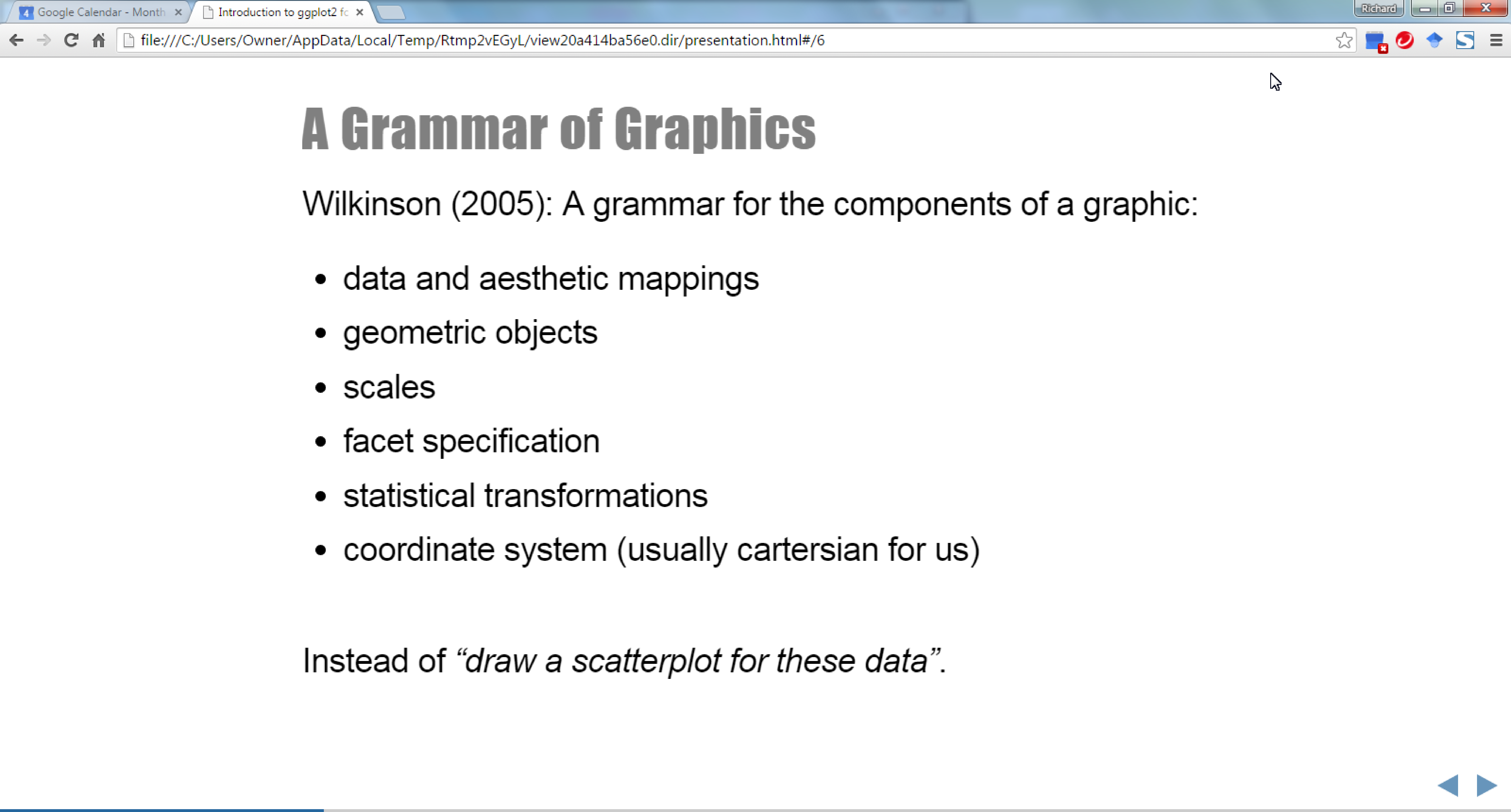
Plotting devices in R

When R draws a plot, it is written to a device.
Plots can be sent to more than 1 device.
Plot appearance depends on device settings (e.g. dimensions, dpi).
The screen plot may look weird when a plot is formated for other devices.

Avaliable devices (?devices):

Screen (can copy to clipboard in Windows)
PDF file
PNG bitmap file
JPEG bitmap file

Typically, write to a file then import in Word or Powerpoint.



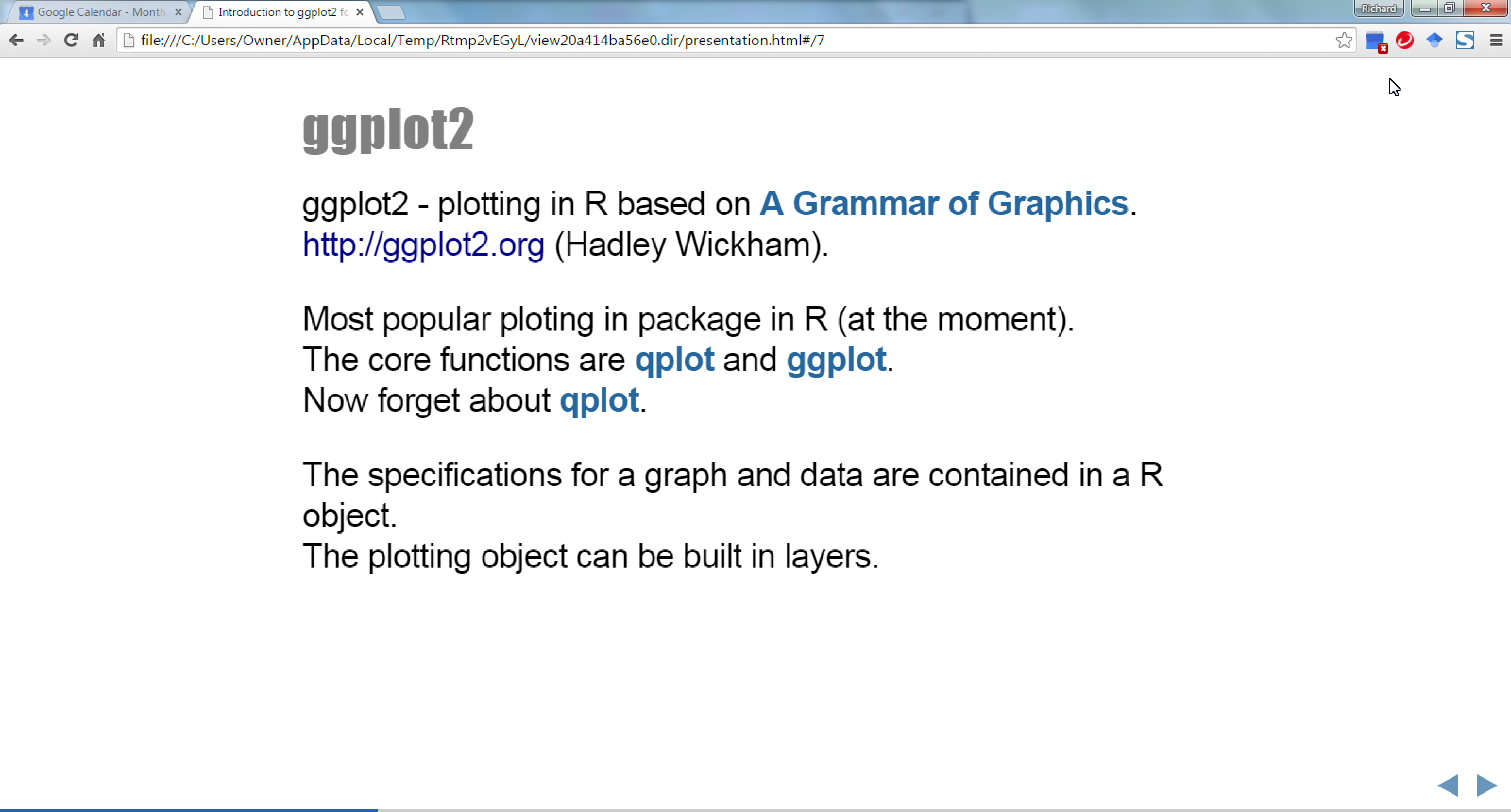
Google Calendar - Month
Introduction to ggplot2 fc
Richard
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/6

A Grammar of Graphics

Wilkinson (2005): A grammar for the components of a graphic:

- data and aesthetic mappings
- geometric objects
- scales
- facet specification
- statistical transformations
- coordinate system (usually cartersian for us)

Instead of *“draw a scatterplot for these data”*.

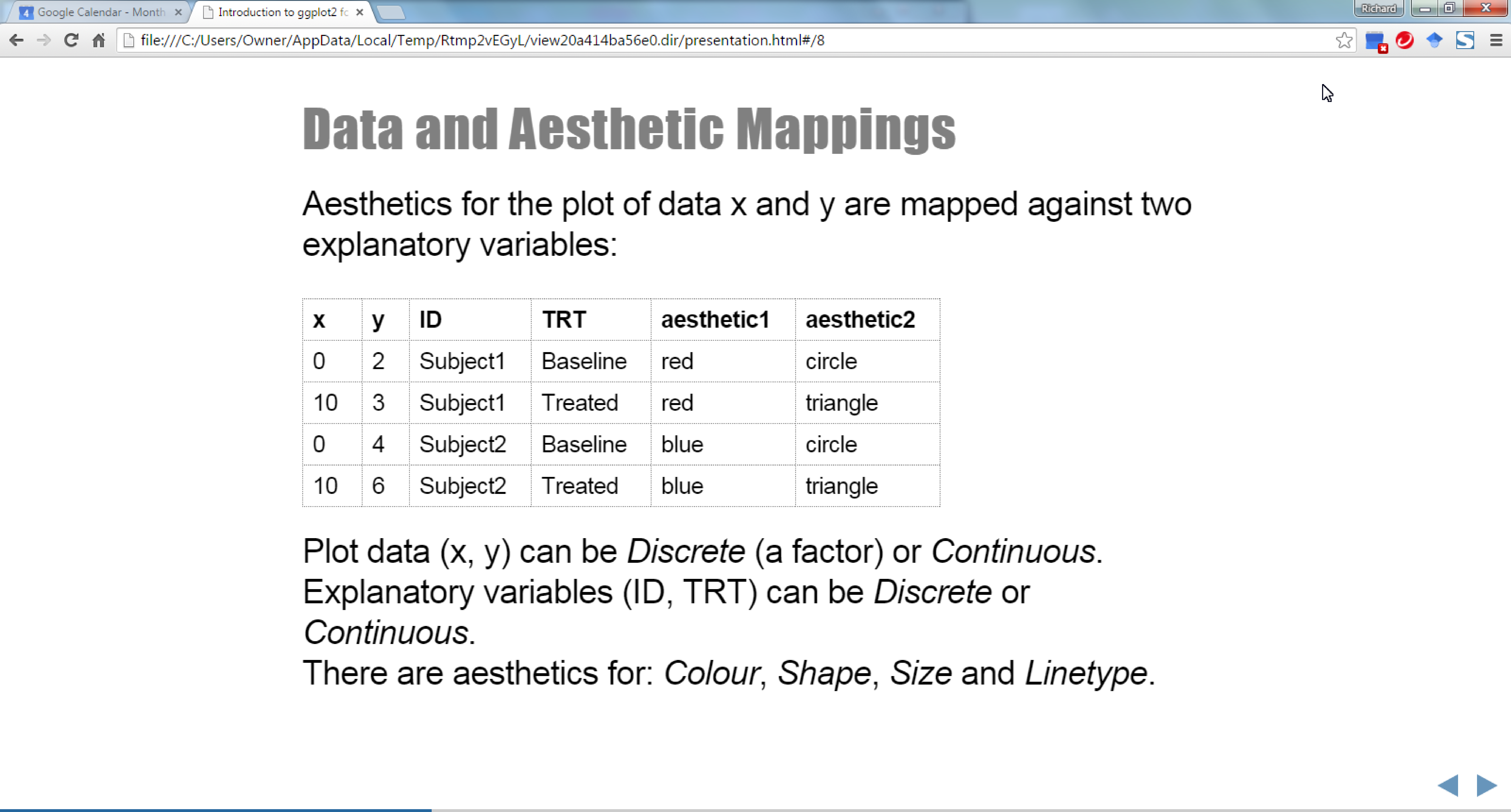

Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/7
Richard

ggplot2

ggplot2 - plotting in R based on **A Grammar of Graphics**.
http://ggplot2.org (Hadley Wickham).

Most popular ploting in package in R (at the moment).
The core functions are **qplot** and **ggplot**.
Now forget about **qplot**.

The specifications for a graph and data are contained in a R object.
The plotting object can be built in layers.

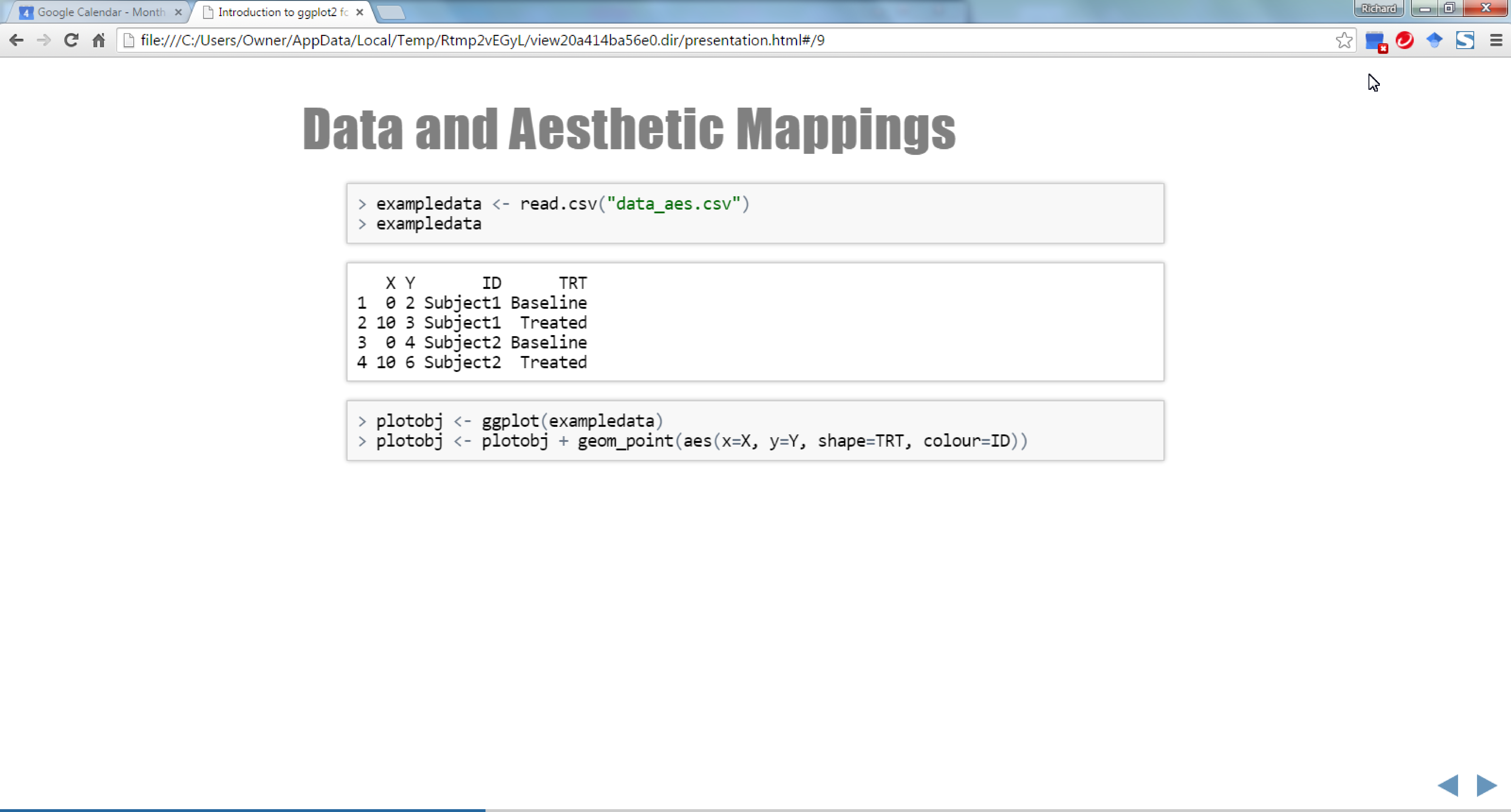

Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/8
Richard

Data and Aesthetic Mappings

Aesthetics for the plot of data x and y are mapped against two explanatory variables:

x	y	ID	TRT	aesthetic1	aesthetic2
0	2	Subject1	Baseline	red	circle
10	3	Subject1	Treated	red	triangle
0	4	Subject2	Baseline	blue	circle
10	6	Subject2	Treated	blue	triangle

Plot data (x, y) can be *Discrete* (a factor) or *Continuous*.
Explanatory variables (ID, TRT) can be *Discrete* or *Continuous*.
There are aesthetics for: *Colour*, *Shape*, *Size* and *Linetype*.



Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/9
Richard

Data and Aesthetic Mappings

```
> exampledata <- read.csv("data_aes.csv")
> exampledata
```

```
   X Y       ID      TRT
1  0 2 Subject1 Baseline
2 10 3 Subject1  Treated
3  0 4 Subject2 Baseline
4 10 6 Subject2  Treated
```

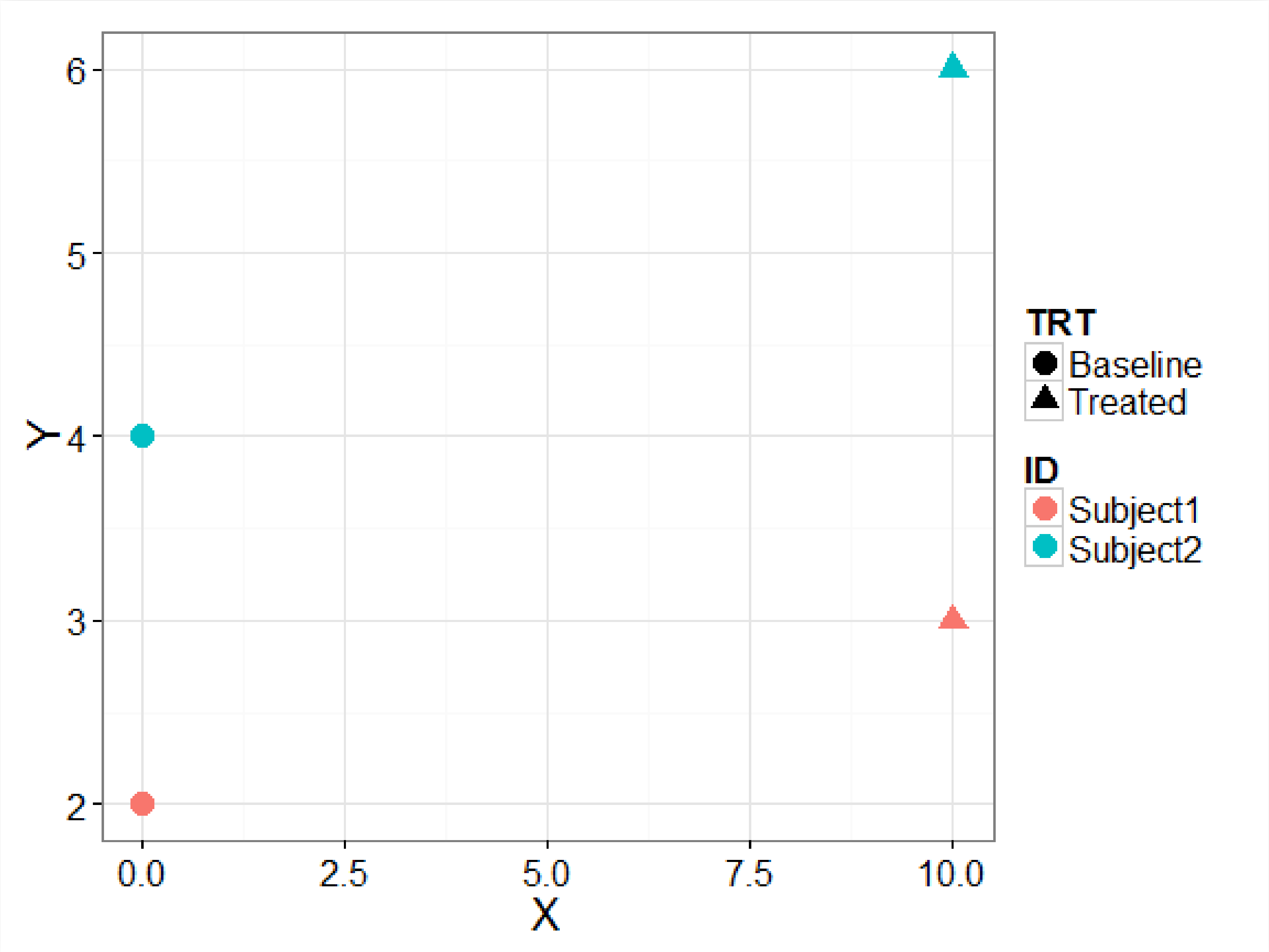
```
> plotobj <- ggplot(exampledata)
> plotobj <- plotobj + geom_point(aes(x=X, y=Y, shape=TRT, colour=ID))
```

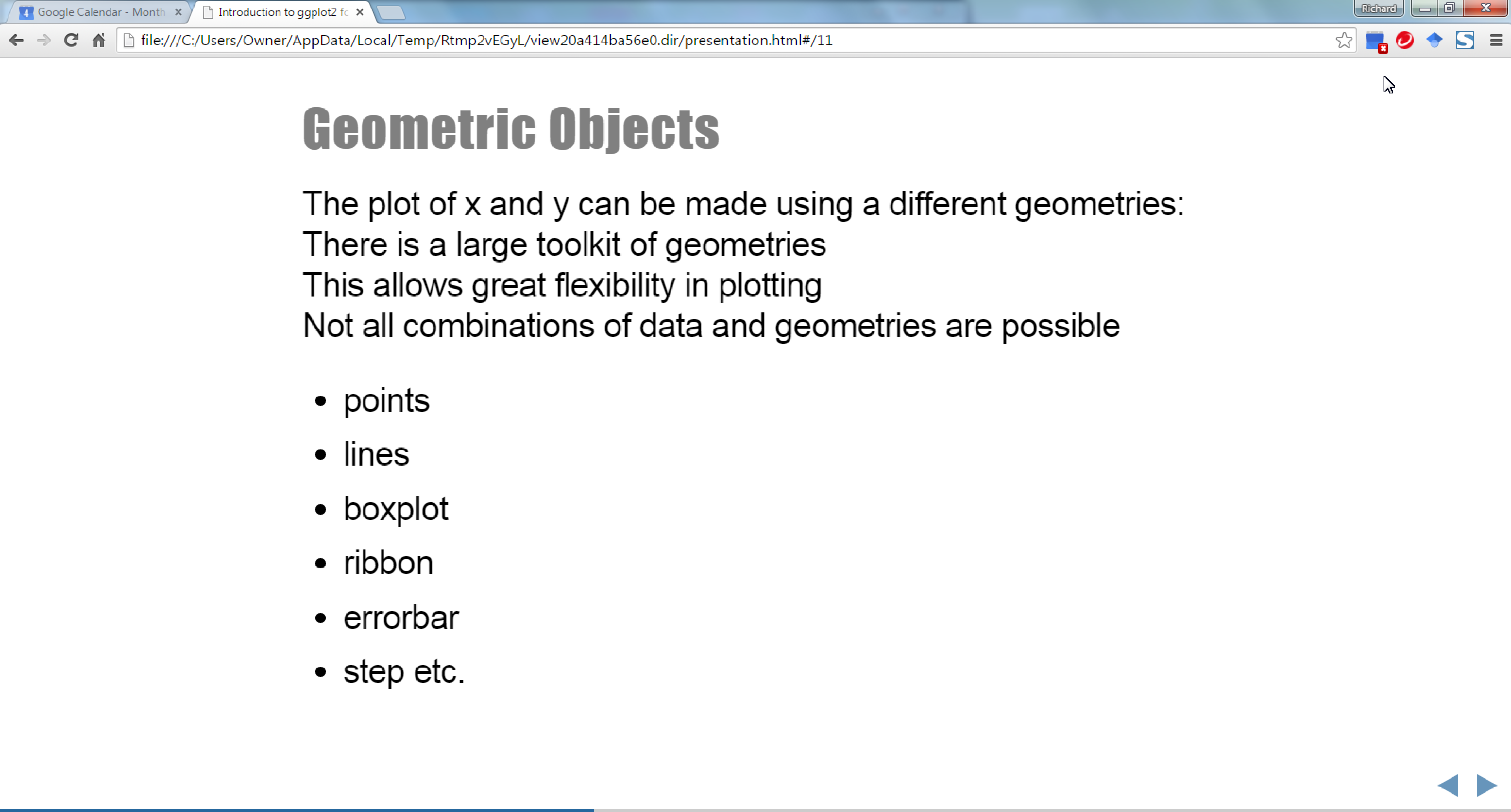
Google Calendar - Month | Introduction to ggplot2 fc

Richard

file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/10

Data and Aesthetic Mappings

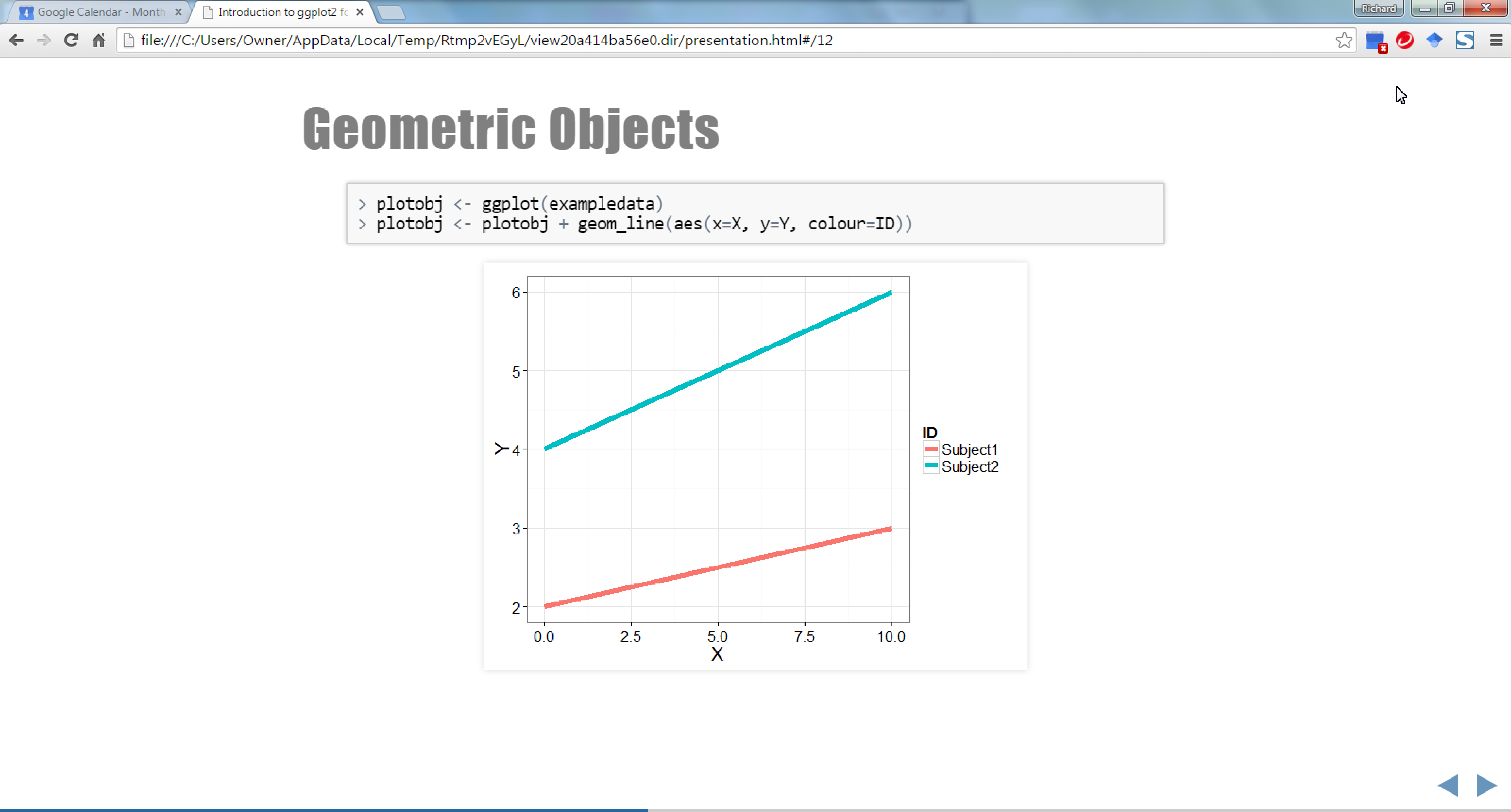



Google Calendar - Month
Introduction to ggplot2 fc
Richard
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/11

Geometric Objects

The plot of x and y can be made using a different geometries:
There is a large toolkit of geometries
This allows great flexibility in plotting
Not all combinations of data and geometries are possible

- points
- lines
- boxplot
- ribbon
- errorbar
- step etc.


Google Calendar - Month
Introduction to ggplot2 fc
Richard
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/12

Geometric Objects

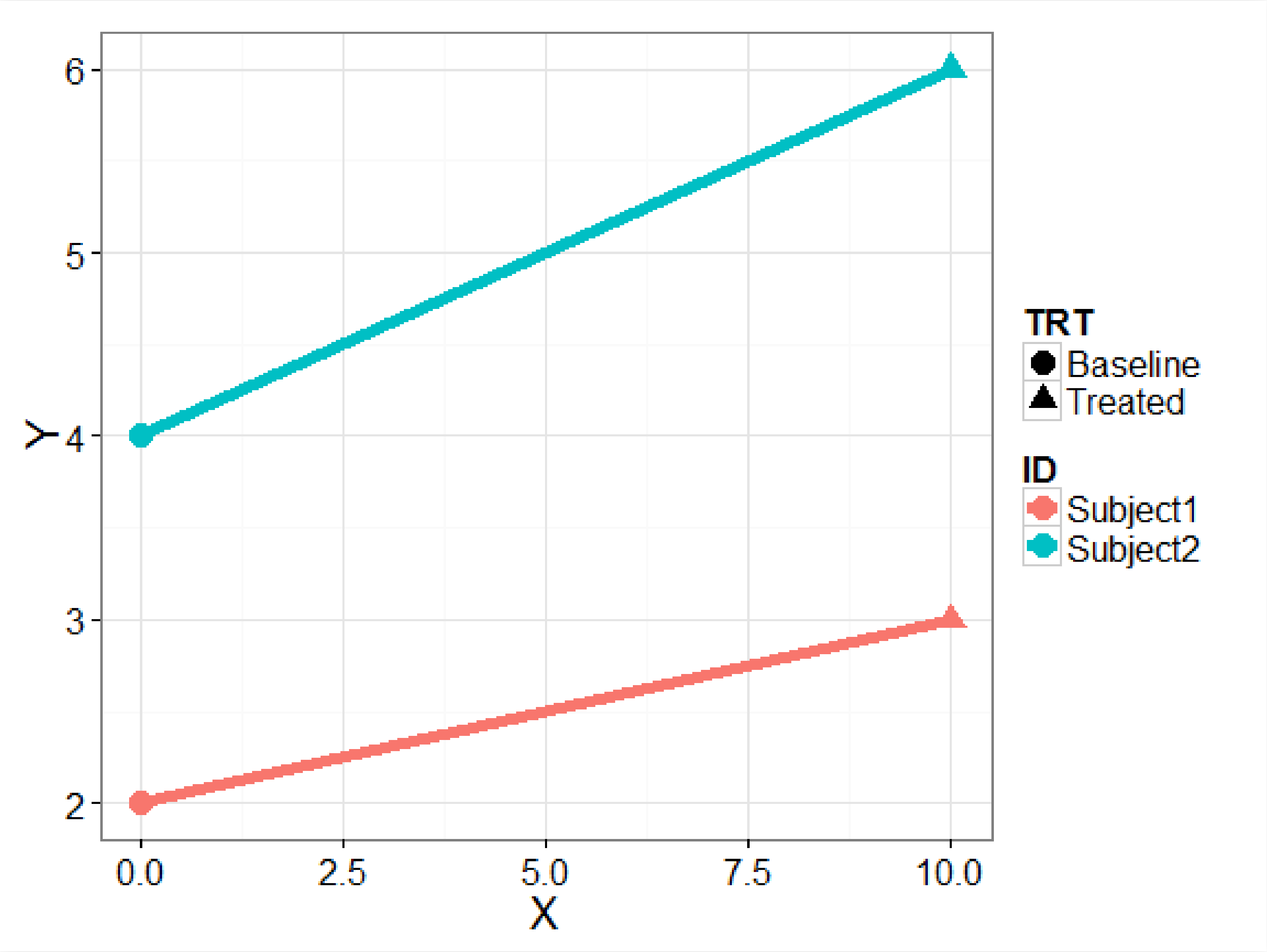
```
> plotobj <- ggplot(exampledata)
> plotobj <- plotobj + geom_line(aes(x=X, y=Y, colour=ID))
```

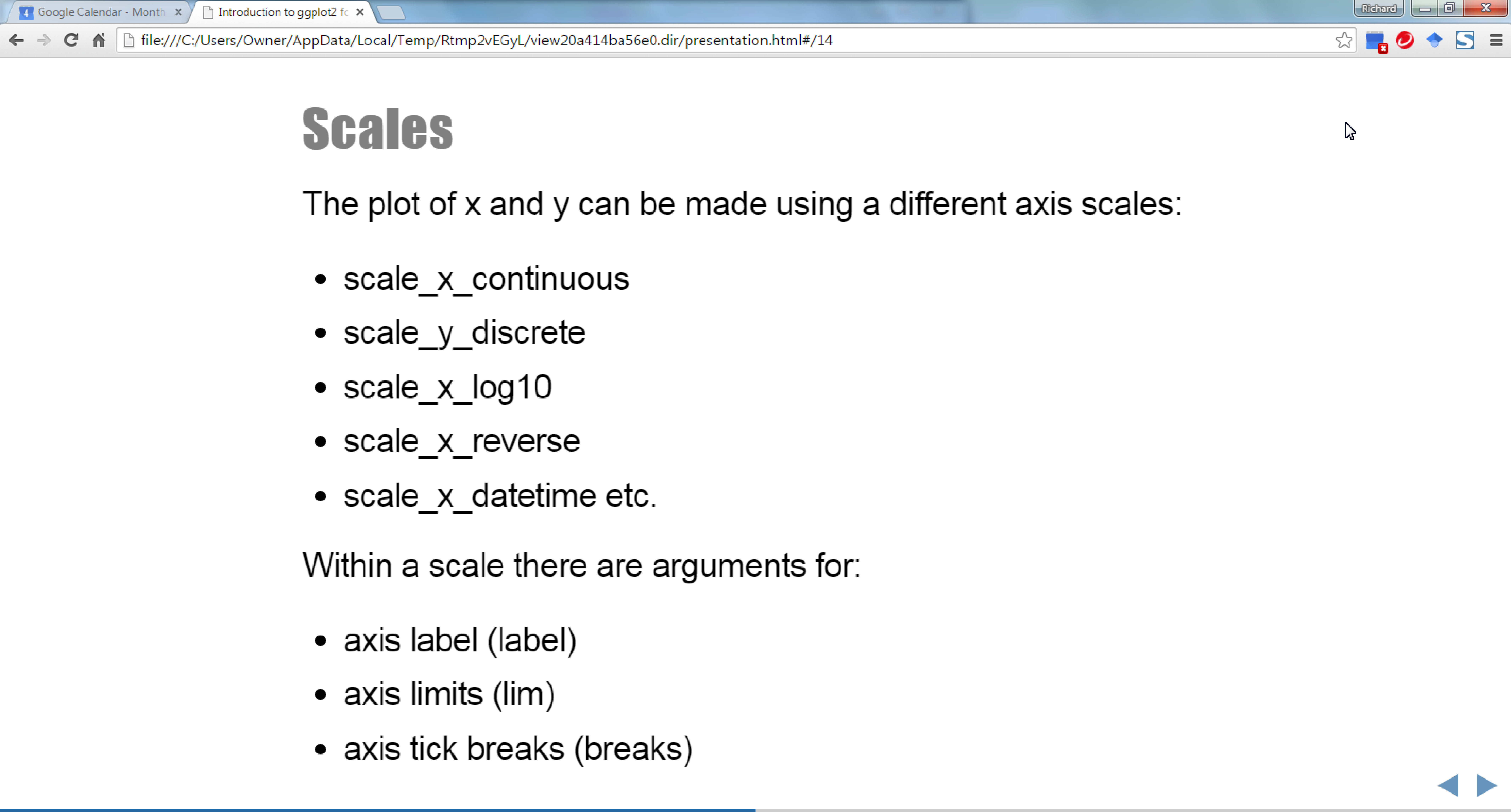
6
5
4
3
2
Y
0.0
2.5
5.0
7.5
10.0
X
ID
Subject1
Subject2

Plots can be built in layers

Two different geometries used in the same plot:

```
> plotobj <- ggplot(exampledata)
> plotobj <- plotobj + geom_point(aes(x=X, y=Y, shape=TRT, colour=ID))
> plotobj <- plotobj + geom_line(aes(x=X, y=Y, colour=ID))
```




Google Calendar - Month
Introduction to ggplot2 fc
Richard
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/14

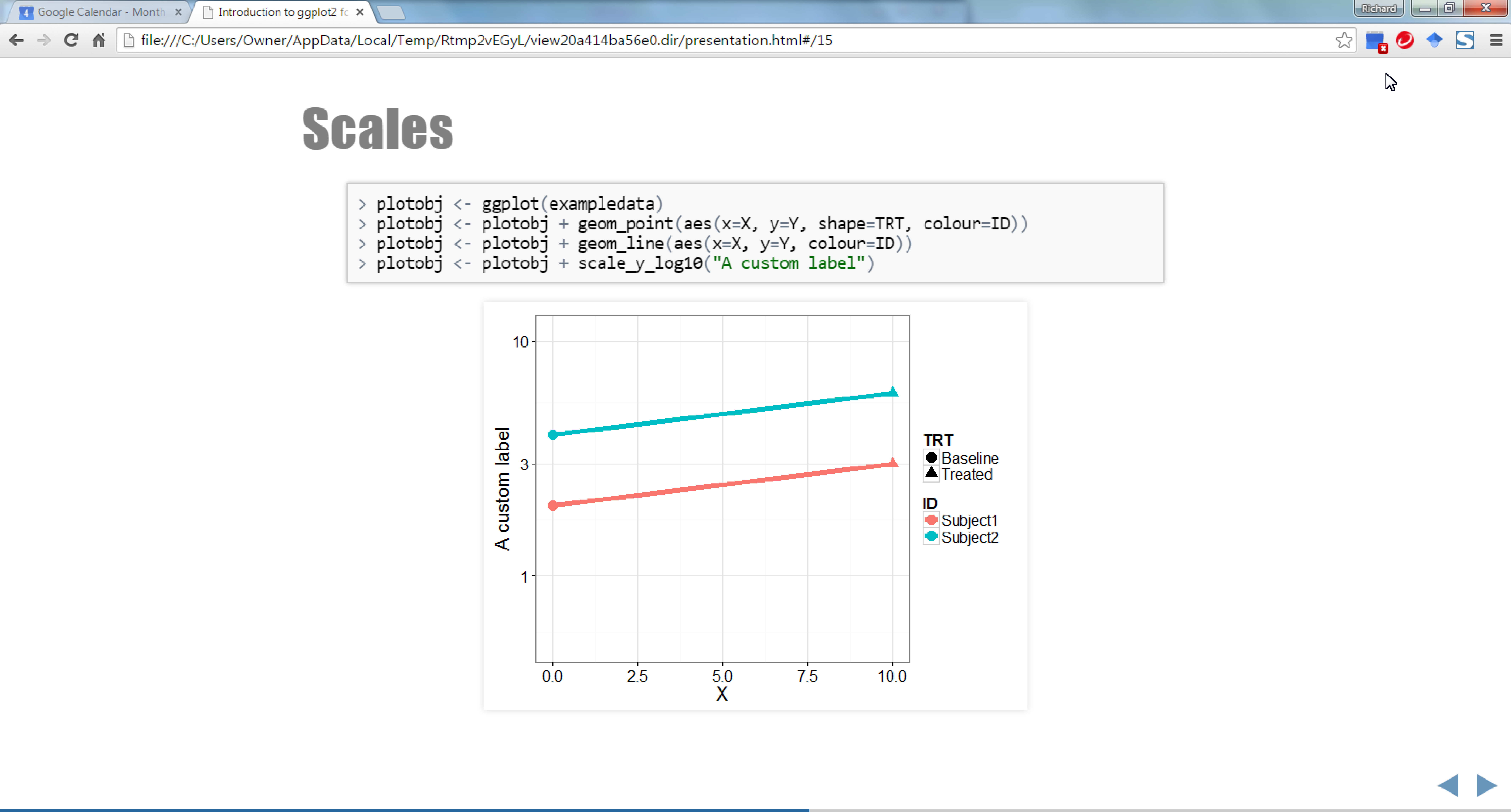
Scales

The plot of x and y can be made using a different axis scales:

- scale_x_continuous
- scale_y_discrete
- scale_x_log10
- scale_x_reverse
- scale_x_datetime etc.

Within a scale there are arguments for:

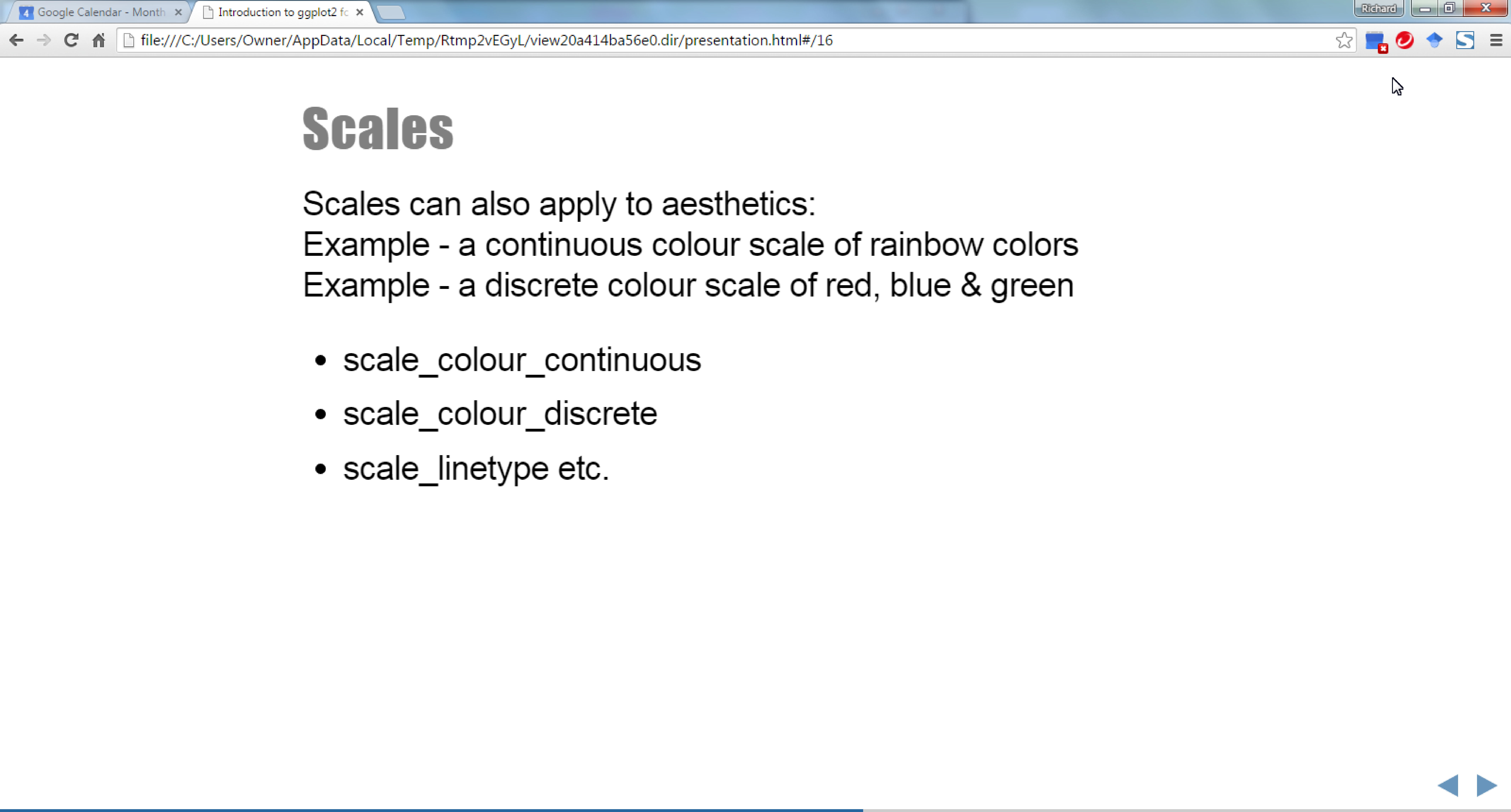
- axis label (label)
- axis limits (lim)
- axis tick breaks (breaks)


Google Calendar - Month
Introduction to ggplot2 fc
Richard
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/15

Scales

```
> plotobj <- ggplot(exampledata)
> plotobj <- plotobj + geom_point(aes(x=X, y=Y, shape=TRT, colour=ID))
> plotobj <- plotobj + geom_line(aes(x=X, y=Y, colour=ID))
> plotobj <- plotobj + scale_y_log10("A custom label")
```

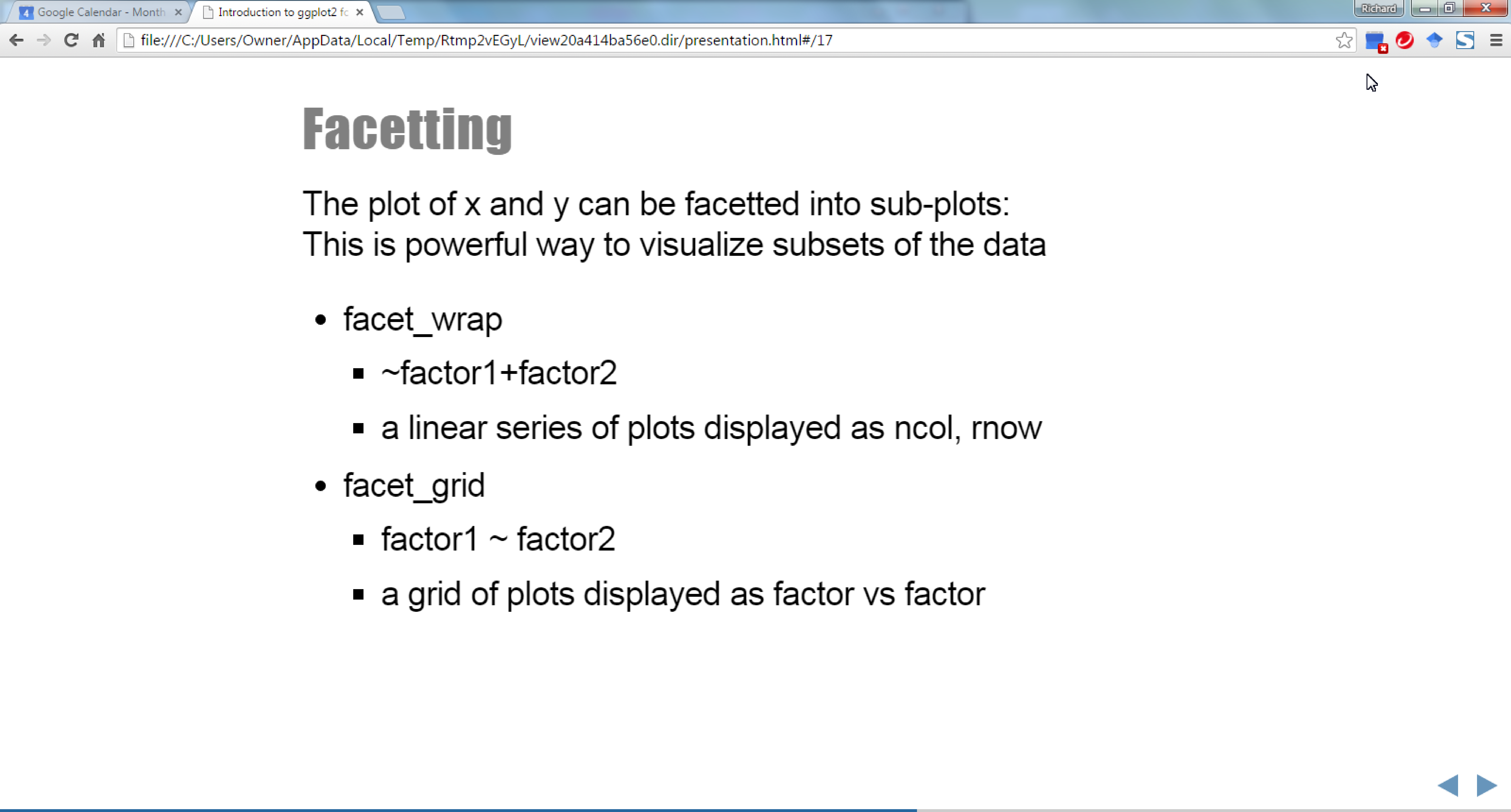
10
3
1
A custom label
0.0
2.5
5.0
7.5
10.0
X
TRT
● Baseline
▲ Treated
ID
● Subject1
● Subject2


Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/16
Richard

Scales

Scales can also apply to aesthetics:
Example - a continuous colour scale of rainbow colors
Example - a discrete colour scale of red, blue & green

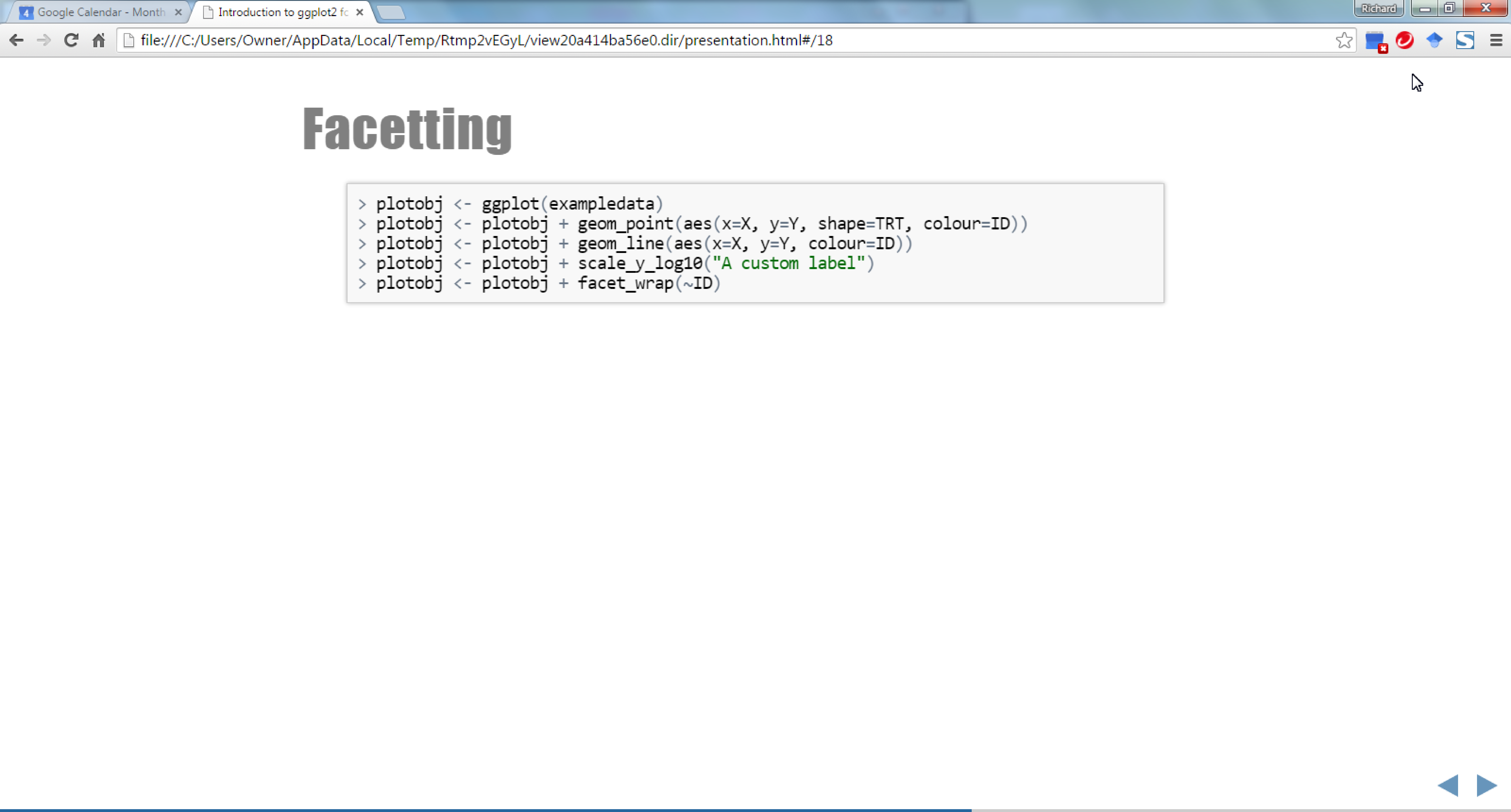
- scale_colour_continuous
- scale_colour_discrete
- scale_linetype etc.


Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/17
Richard

Facetting

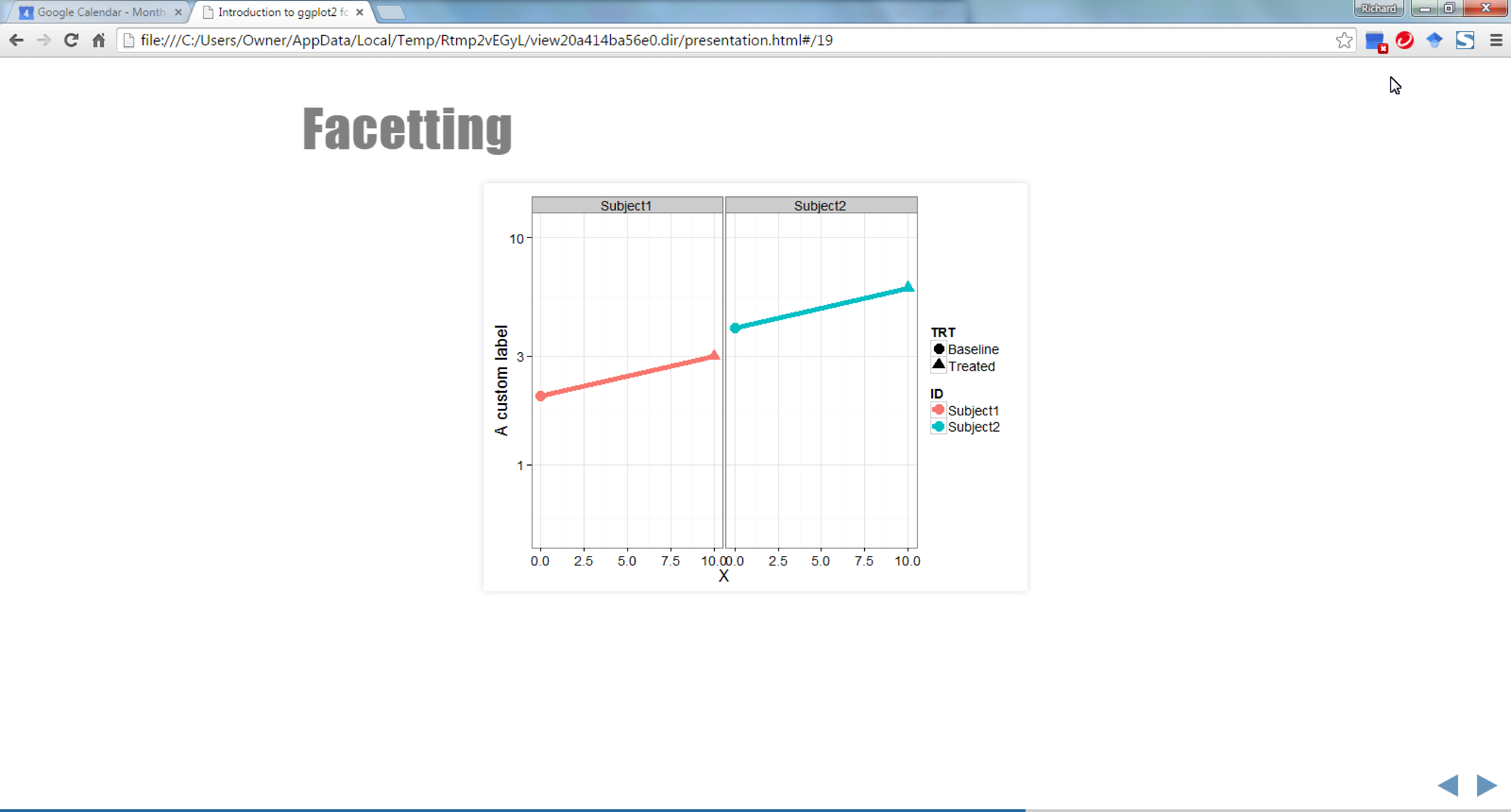
The plot of x and y can be facetted into sub-plots:
This is powerful way to visualize subsets of the data

- facet_wrap
  - ~factor1+factor2
  - a linear series of plots displayed as ncol, rnow
- facet_grid
  - factor1 ~ factor2
  - a grid of plots displayed as factor vs factor


Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/18
Richard

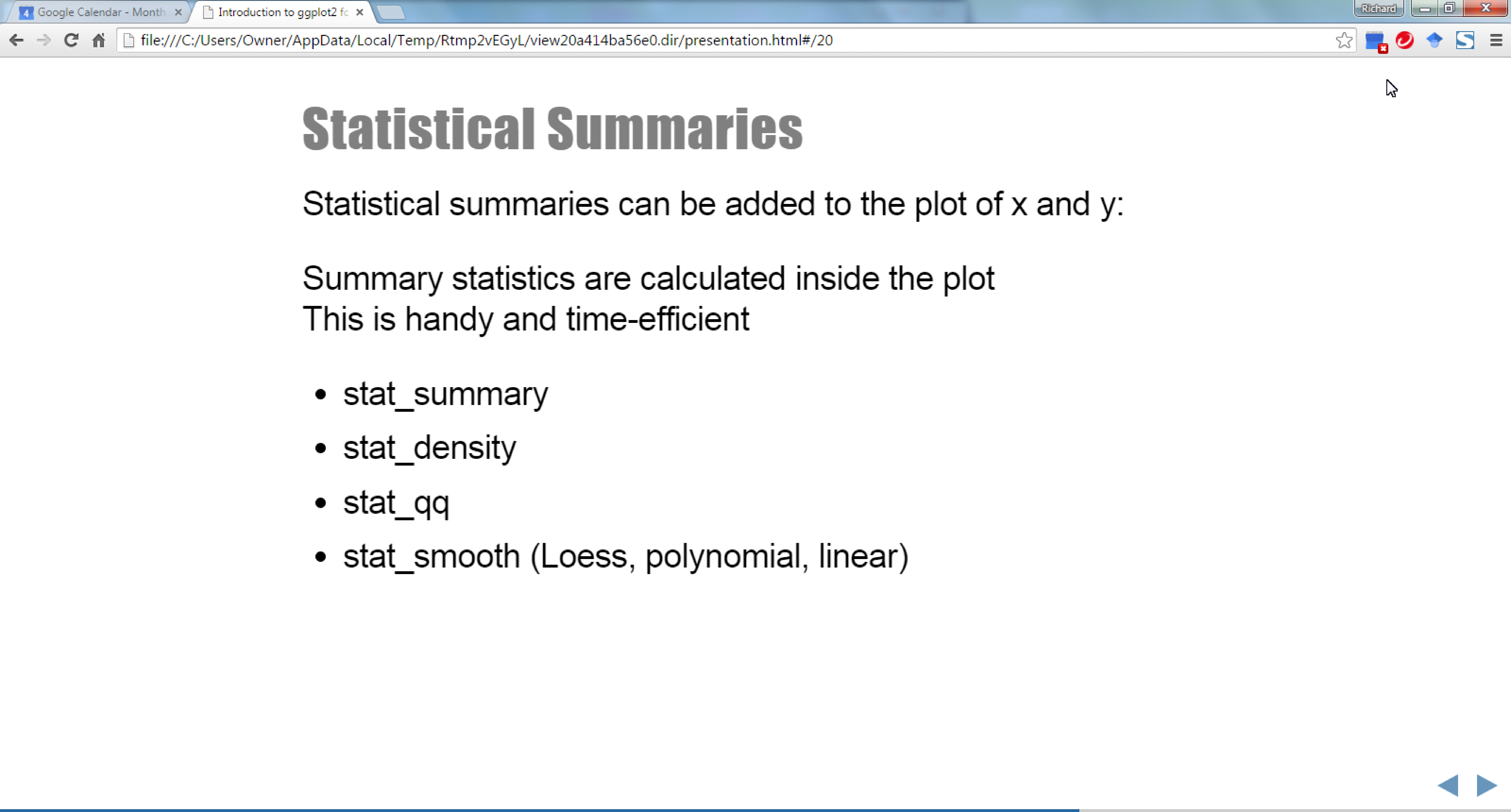
Facetting

```
> plotobj <- ggplot(exampledata)
> plotobj <- plotobj + geom_point(aes(x=X, y=Y, shape=TRT, colour=ID))
> plotobj <- plotobj + geom_line(aes(x=X, y=Y, colour=ID))
> plotobj <- plotobj + scale_y_log10("A custom label")
> plotobj <- plotobj + facet_wrap(~ID)
```



file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/19

Facetting

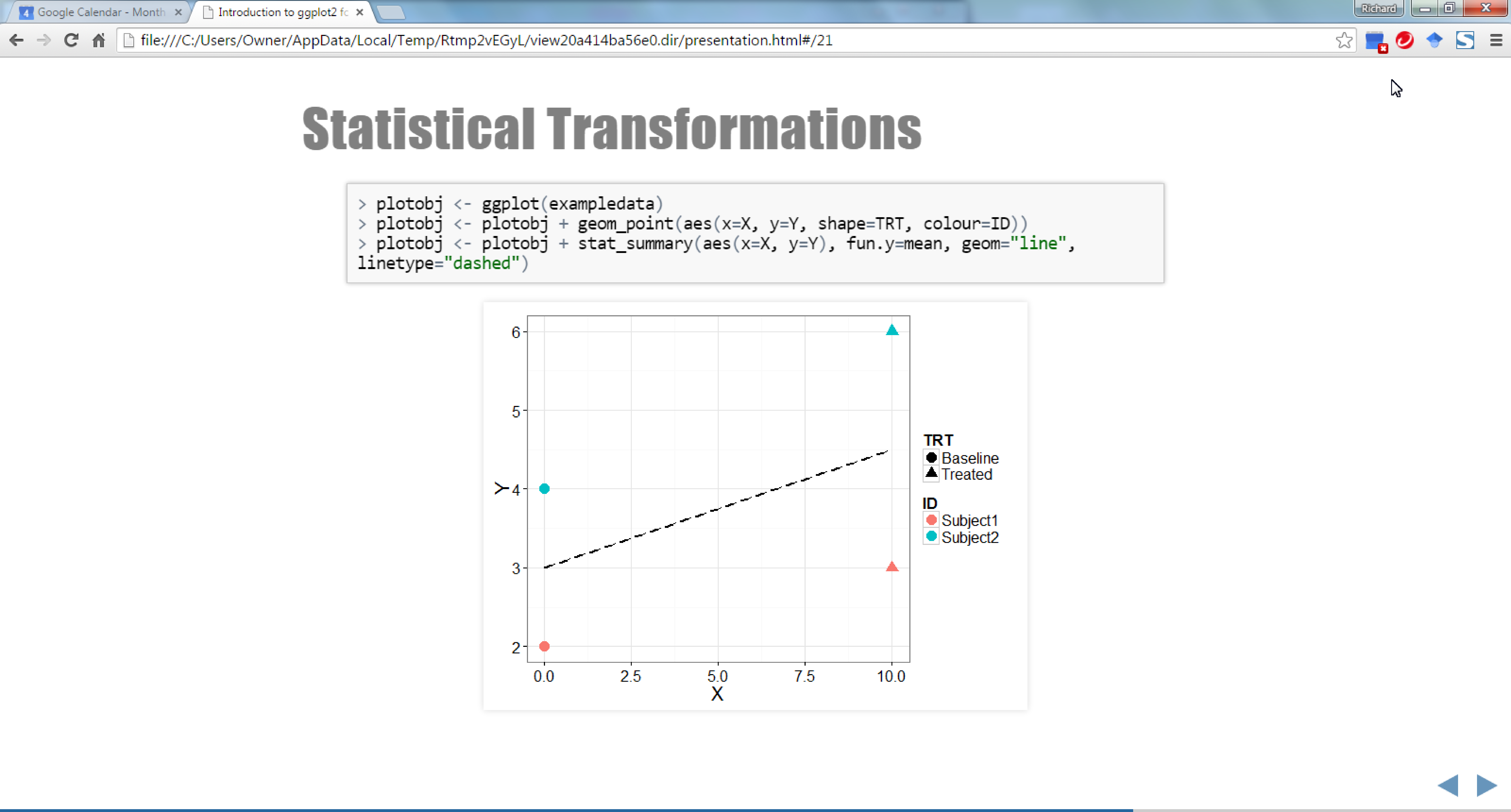

Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/20
Richard

Statistical Summaries

Statistical summaries can be added to the plot of x and y:

Summary statistics are calculated inside the plot
This is handy and time-efficient

- stat_summary
- stat_density
- stat_qq
- stat_smooth (Loess, polynomial, linear)

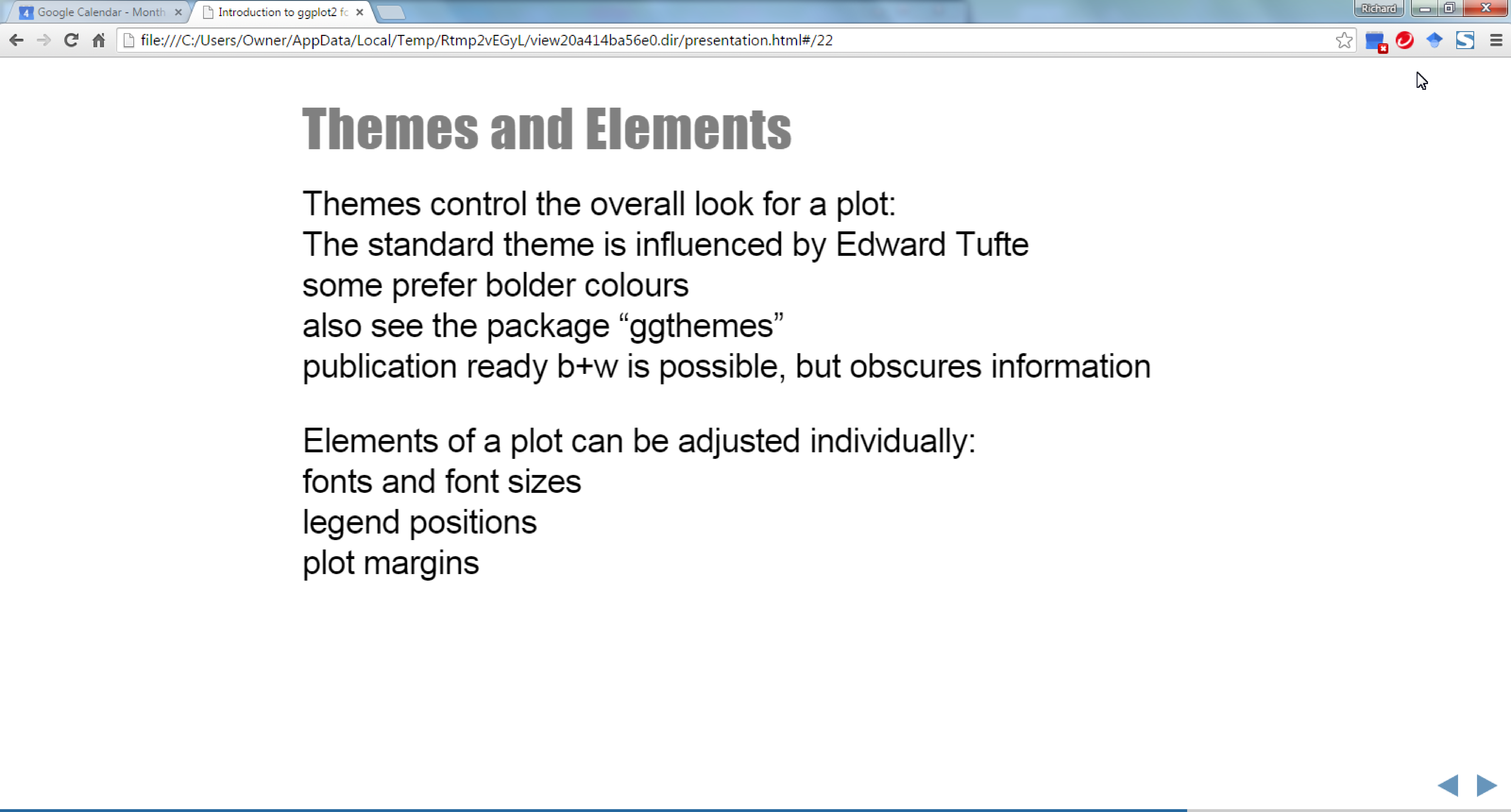


Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/21
Richard

Statistical Transformations

```
> plotobj <- ggplot(exampledata)
> plotobj <- plotobj + geom_point(aes(x=X, y=Y, shape=TRT, colour=ID))
> plotobj <- plotobj + stat_summary(aes(x=X, y=Y), fun.y=mean, geom="line",
linetype="dashed")
```

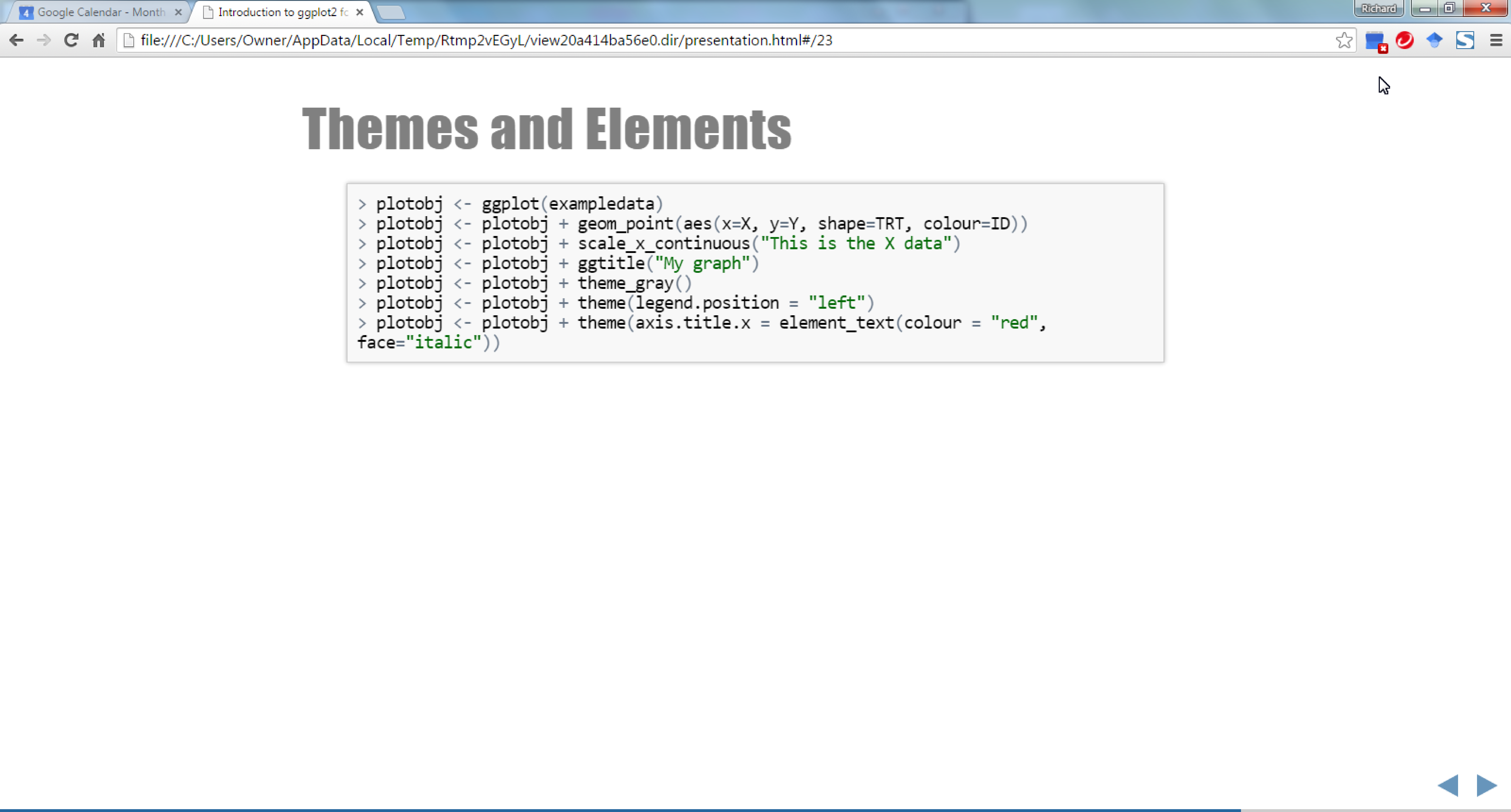
6
5
4
3
2
Y
0.0
2.5
5.0
7.5
10.0
X
TRT
● Baseline
▲ Treated
ID
● Subject1
● Subject2


Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/22
Richard

Themes and Elements

Themes control the overall look for a plot:
The standard theme is influenced by Edward Tufte
some prefer bolder colours
also see the package “ggthemes”
publication ready b+w is possible, but obscures information

Elements of a plot can be adjusted individually:
fonts and font sizes
legend positions
plot margins


Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/23
Richard

Themes and Elements

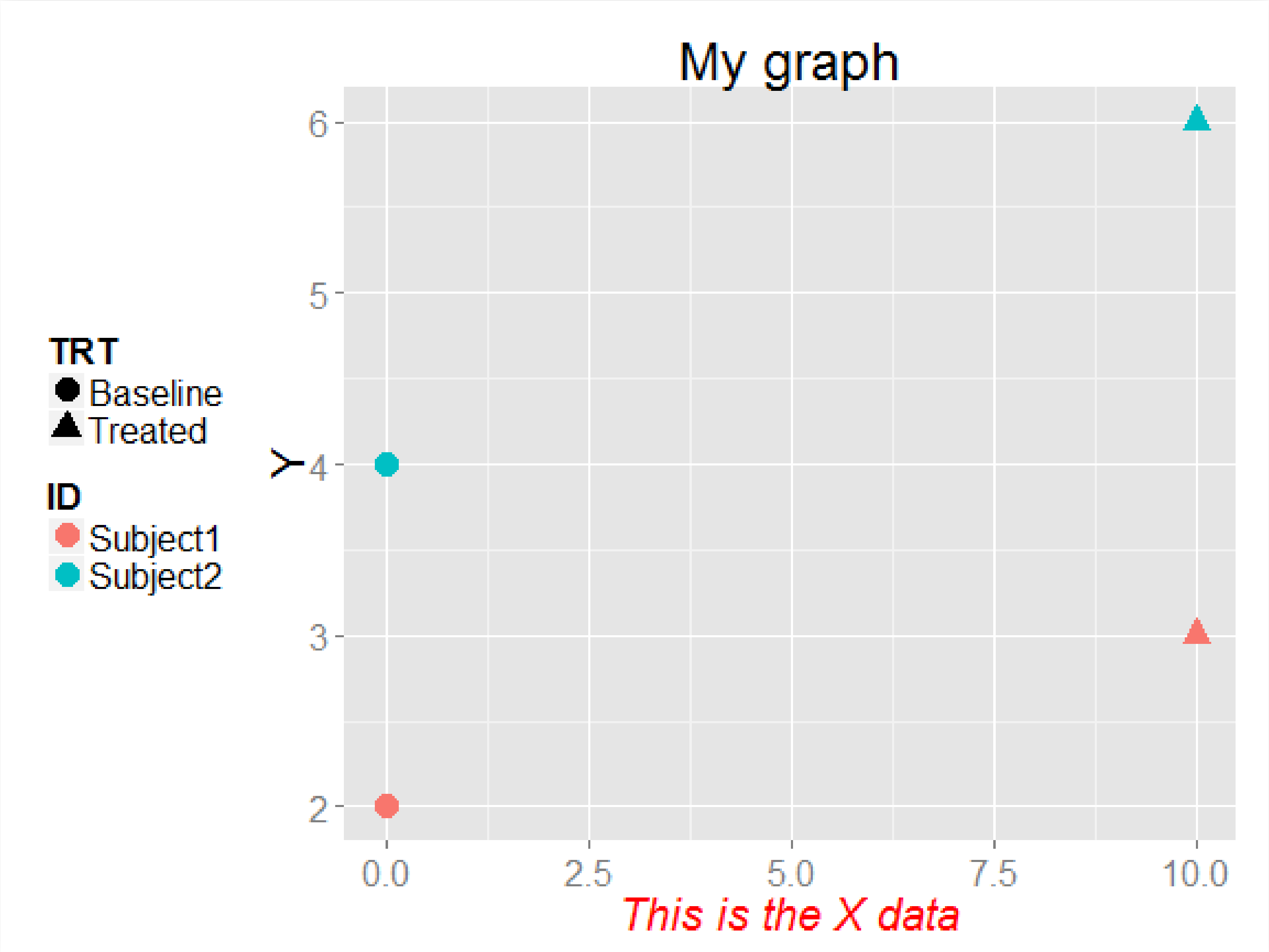
```
> plotobj <- ggplot(exampledata)
> plotobj <- plotobj + geom_point(aes(x=X, y=Y, shape=TRT, colour=ID))
> plotobj <- plotobj + scale_x_continuous("This is the X data")
> plotobj <- plotobj + ggtitle("My graph")
> plotobj <- plotobj + theme_gray()
> plotobj <- plotobj + theme(legend.position = "left")
> plotobj <- plotobj + theme(axis.title.x = element_text(colour = "red",
face="italic"))
```

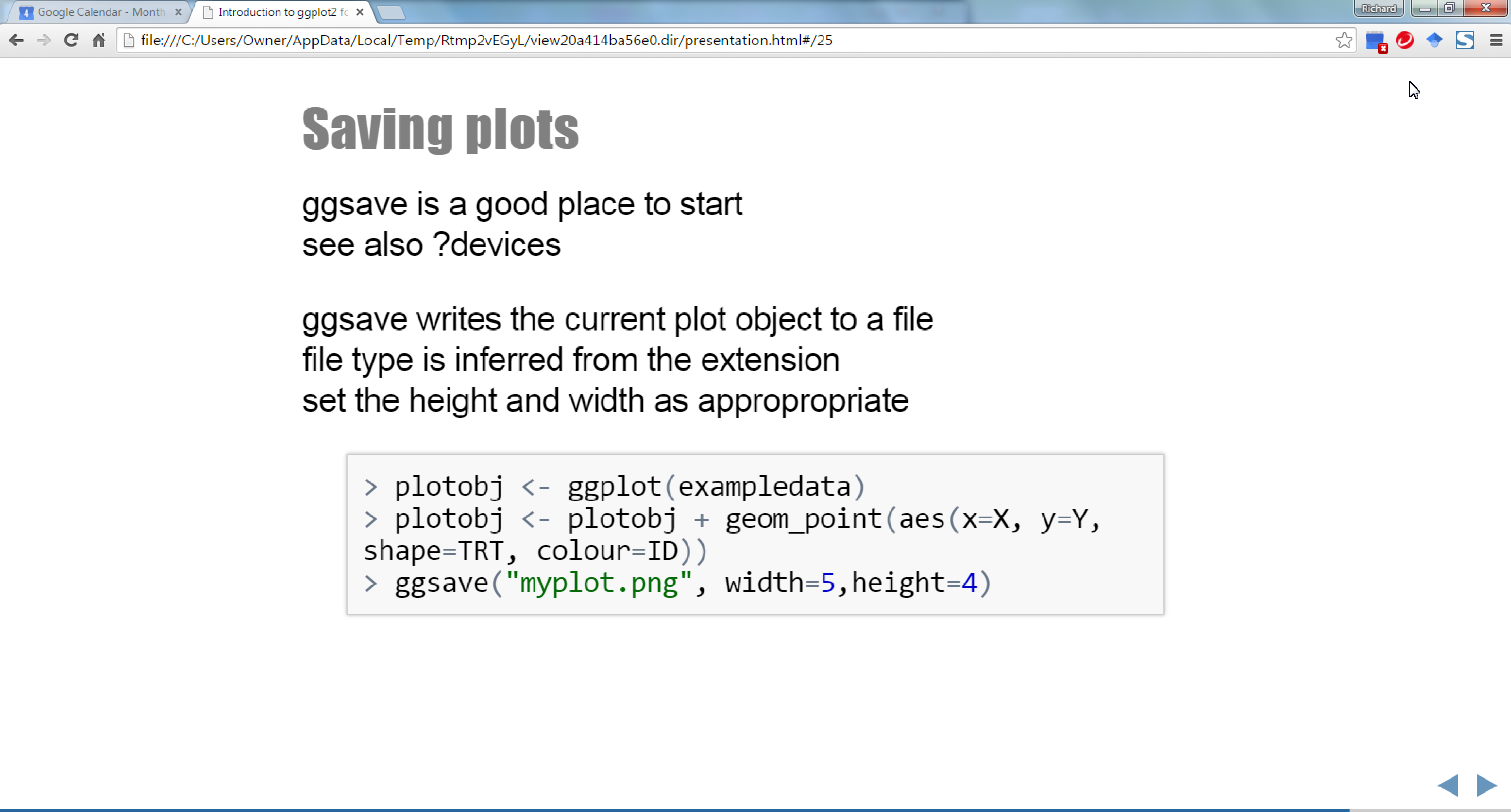
Google Calendar - Month

Introduction to ggplot2 fc

file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/24

Themes and Elements



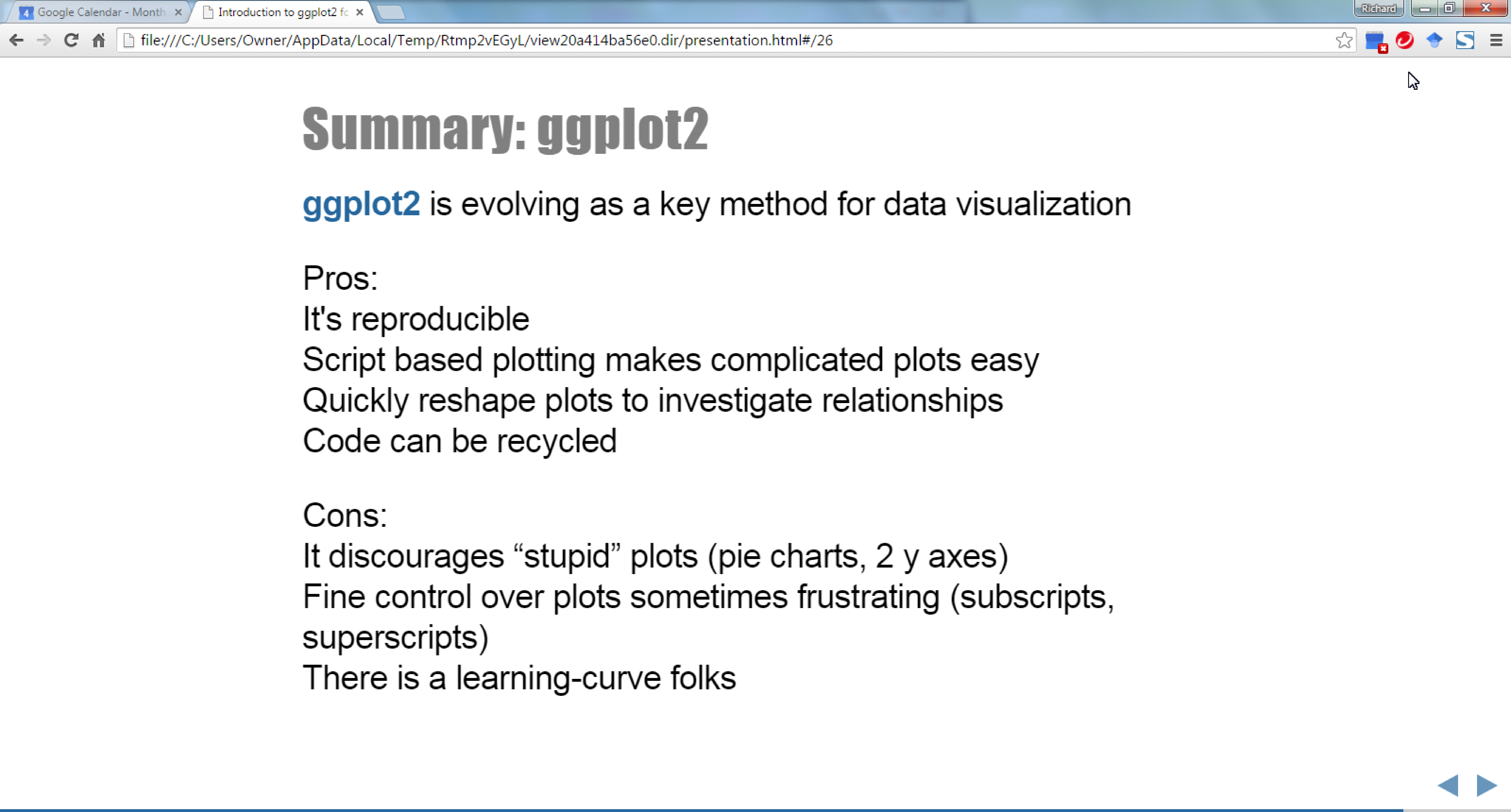

Google Calendar - Month
Introduction to ggplot2 fc
Richard
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/25

Saving plots

ggsave is a good place to start
see also ?devices

ggsave writes the current plot object to a file
file type is inferred from the extension
set the height and width as appropropriate

```
> plotobj <- ggplot(exampledata)
> plotobj <- plotobj + geom_point(aes(x=X, y=Y,
shape=TRT, colour=ID))
> ggsave("myplot.png", width=5,height=4)
```


Google Calendar - Month
Introduction to ggplot2 fc
Richard
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/26

Summary: ggplot2

ggplot2 is evolving as a key method for data visualization

Pros:
It's reproducible
Script based plotting makes complicated plots easy
Quickly reshape plots to investigate relationships
Code can be recycled

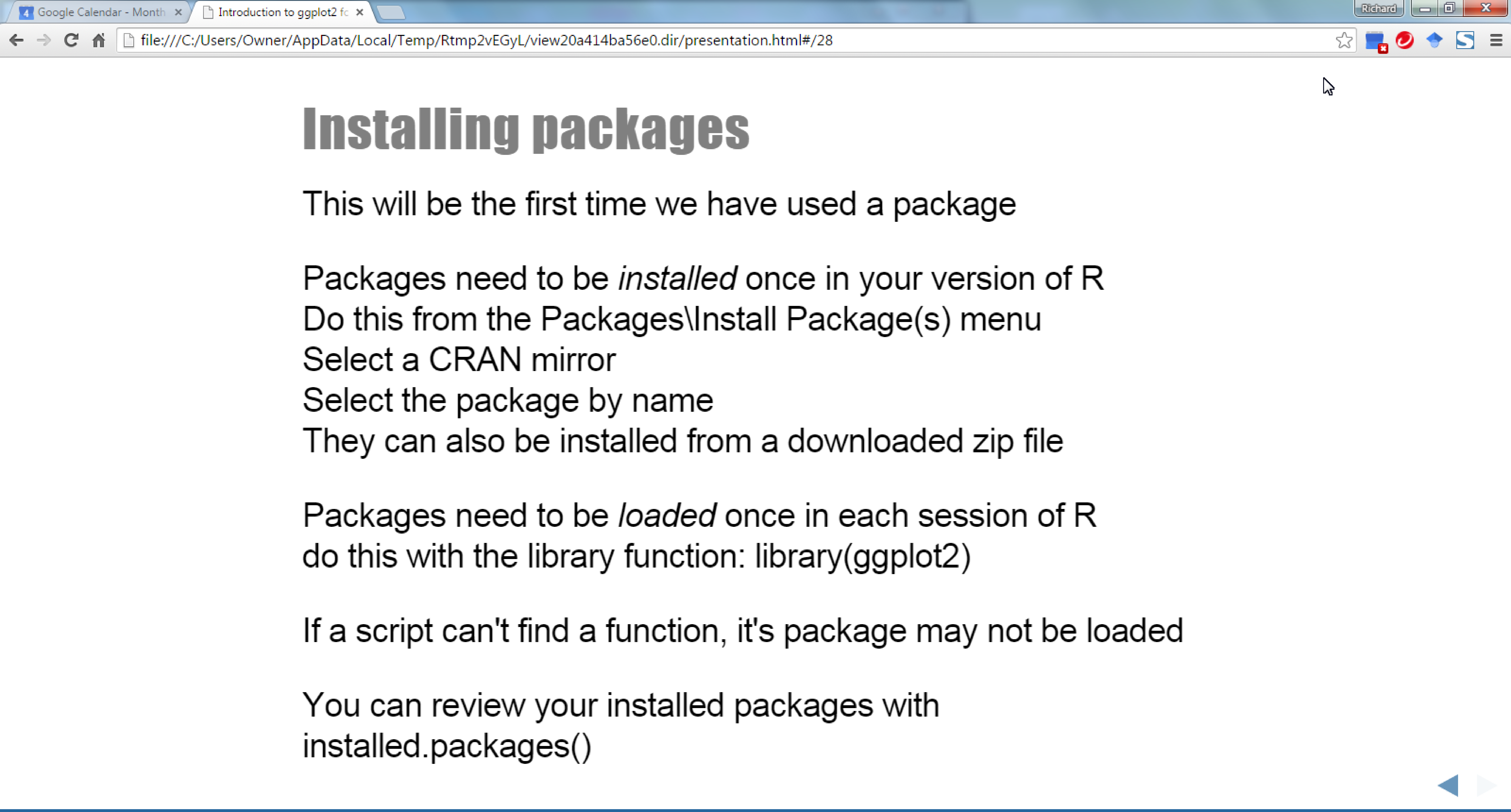
Cons:
It discourages “stupid” plots (pie charts, 2 y axes)
Fine control over plots sometimes frustrating (subscripts, superscripts)
There is a learning-curve folks

Google Calendar - Month

Introduction to ggplot2 fc

file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/27

Richard


Google Calendar - Month
Introduction to ggplot2 fc
file:///C:/Users/Owner/AppData/Local/Temp/Rtmp2vEGyL/view20a414ba56e0.dir/presentation.html#/28
Richard

Installing packages

This will be the first time we have used a package

Packages need to be *installed* once in your version of R
Do this from the Packages\Install Package(s) menu
Select a CRAN mirror
Select the package by name
They can also be installed from a downloaded zip file

Packages need to be *loaded* once in each session of R
do this with the library function: library(ggplot2)

If a script can't find a function, it's package may not be loaded

You can review your installed packages with
installed.packages()