
Graphics in R - A Shiny Way to Program

Roche Products Ltd

Carolyn Nixon



Contents



About R

Roche



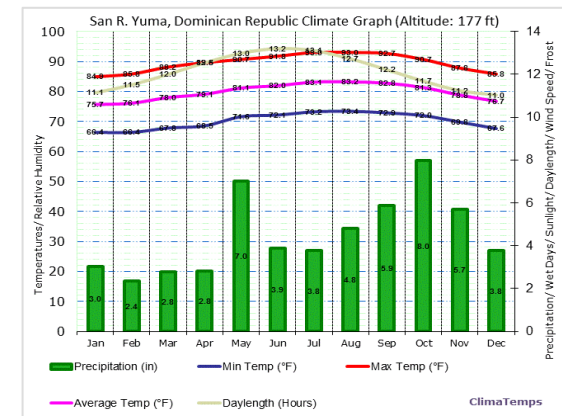
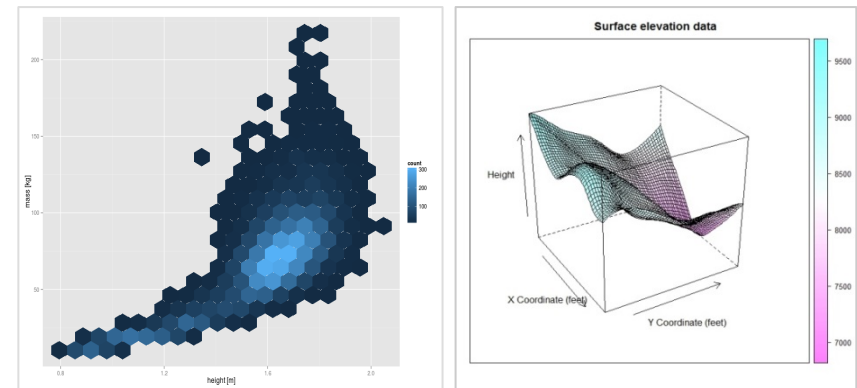
Statistical programming language and software environment.

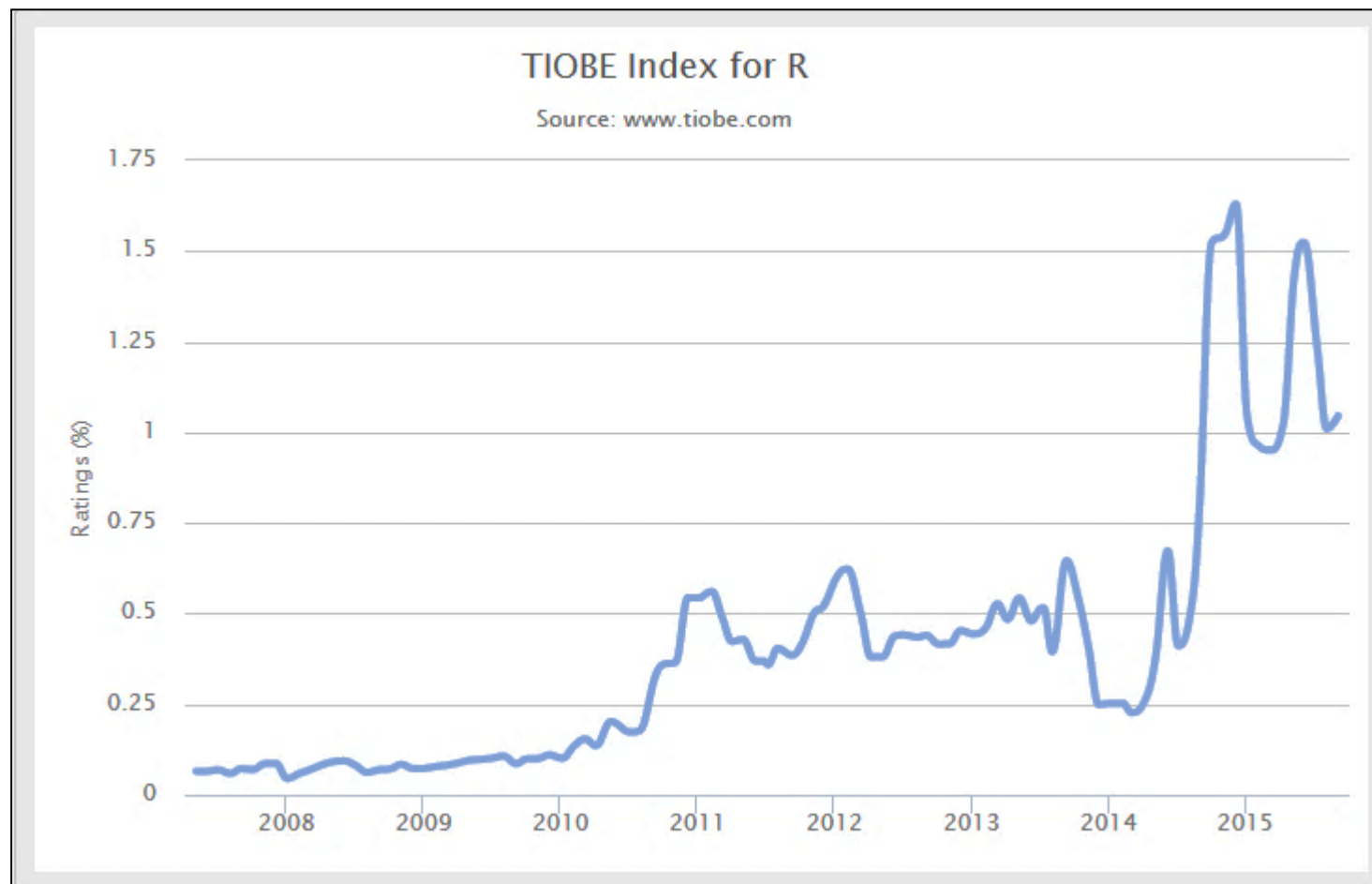
Free to use/Open Source

Wide variety of graphical and statistical techniques

Easy for those without computer programming experience to pick up

Companies such as Google, Pfizer, Merck, Bank of America, the InterContinental Hotels Group and Shell use it for data analysis





Why would I recommend R?


File Edit Code View Plots Session Build Debug Tools Help

Go to file/function Publish

Untitled1* server.R global.R ULR Untitled1* phuse.R Untitled2* phusemarkdown.Rmd df asl

Run App

Environment History

Global Environment

Data

Values

```

1 x <- c(1,2,3,4,5,6,7,8,9,10)
2 y <- c(50,40,57,58,30,45,56,43,42,38)
3 z <- c("Blue","Blue","Brown","Brown","Brown","Brown","Blue","Green","Brown","Brown")
4 zz <- c("Adult","Adult","Child","Adult","Adult","Adult","Adult","Child","Adult")
5
6 df <- data.frame(x,y,z,zz)
7
8 plot(df$x,df$y)
9 plot(asl$armcd,asl$age)
10 plot(x, #x axis
11       y, #y axis
12       xlab="Patient Number",
13       ylab="Age",
14       pch = 21, #plotting symbol
15       mar = c(4, 4, 2, 2), #plot margin size
16       col = "black", #border colour for points
17       bg = "green", #background colour for points
18       ylim = c(25,65)) #y axis limits
19
20 fit <- lm(y ~ x) #fit a linear model between variables y and x
21 abline(fit, #abline is a function that adds a line to the current plot
22        lwd = 3, #line width
23        col = "blue") #colour of line
24
25 title(main = "Title of my plot",
26       ) #title of plot
27
28 text(3, 30, "Label") #Label on graph
29
30 legend("bottomleft", #add a legend to plot
31       legend = "Data", #title of legend
32       pch = 21, #plotting symbol
33       pt.bg = "green", #background colour for points
34       col = "black") #border colour for points
35
36 #Lattice Plot
37 x <- c(1,2,3,4,5,6,7,8,9,10)
38 y <- c(50,40,57,58,30,45,56,43,42,38)
39 z <- c("Blue","Blue","Brown","Brown","Brown","Brown","Blue","Green","Brown","Brown")
40 zz <- c("Adult","Adult","Child","Adult","Adult","Adult","Adult","Child","Adult")

```

9:24 (Top Level)

Console

```

4: In max(x): no non-missing arguments to max; returning -Inf
To access BCE SAS datasets use read.bce like:
dm <- read.bce('/opt/BIOSTAT/sdtm/i28252a/dm.sas7bdat')
or all SAS datasets of a directory into a list of data frames like:
list.df <- read.bce('/opt/BIOSTAT/sdtm/i28252a')

Properly initialised - see http://rochewiki.roche.com/confluence/display/R/read.shogun -
you can read SHOGUN data with read.shogun like:
demog <- read.shogun('/onco/pim/pim4605g/final/rawdata/demog.sas7bdat')
or
list.df <- read.shogun('/onco/pim/pim4605g/final/rawdata')

```

Files Plots Packages Help Viewer

R: R Interface to Google Charts Find in Topic

googleVis-package (googleVis) R Documentation

R Interface to Google Charts

Description

R interface to Google Charts API, allowing users to create interactive charts based on data frames. Charts are displayed locally via the R HTTP help server. A modern browser with Internet connection is required and for some charts Flash. The data remains local and is not uploaded to Google.

Please visit the project web site for more information: <http://github.com/mages/googleVis>.

You find further notes on Markus' blog: <http://www.magesblog.com/search/label/googleVis>

Note

See vignette("googleVis") for an introduction to the googleVis package.

Author(s)

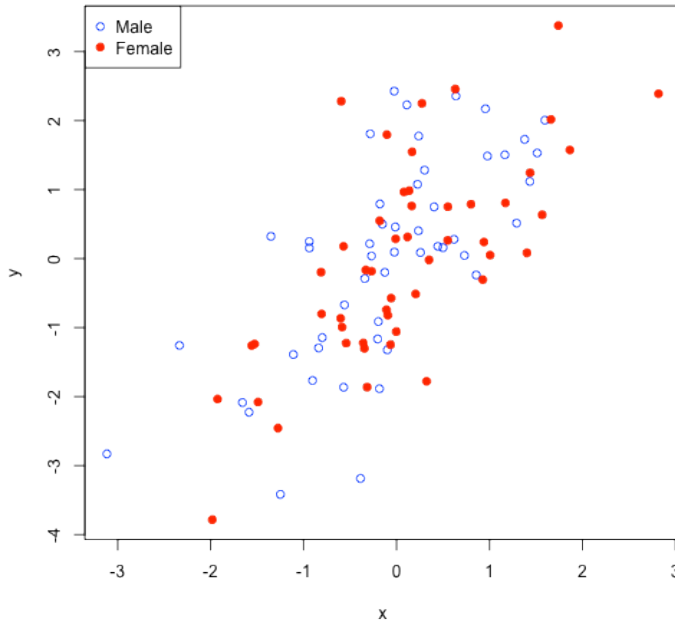
Markus Gesmann, Diego de Castillo

References

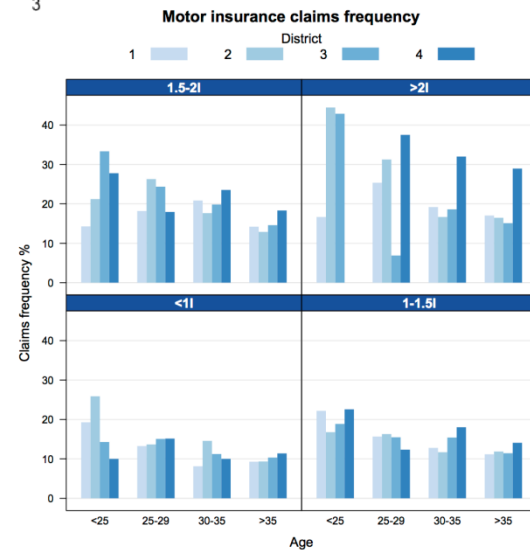
- Google Charts API: <https://developers.google.com/chart/>
- Google Terms of Use: <https://developers.google.com/terms/>
- Google Maps API Terms of Service: <https://developers.google.com/maps/terms>

vast
community
and online
support

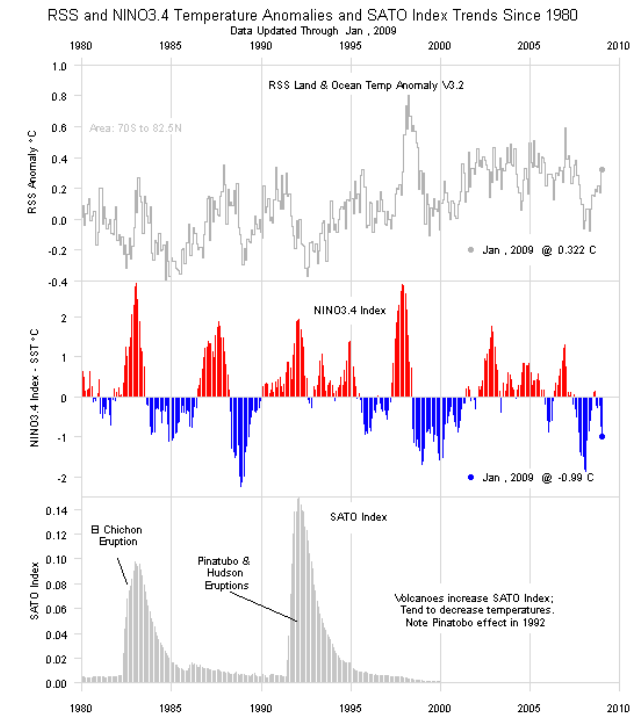
Three Choices



Base



Lattice



ggplot2

Base Graphics

Roche



Base Example

```
x <- c(1,2,3,4,5,6,7,8,9,10)
y <- c(50,49,57,58,30,45,56,43,42,38)
z <- c("Blue", "Blue", "Brown", "Brown", "Brown", "Brown", "Blue",
       "Green", "Brown", "Brown")
zz <- c("Adult", "Adult", "Child", "Adult", "Adult", "Adult", "Adult",
       "Adult", "Child", "Adult")

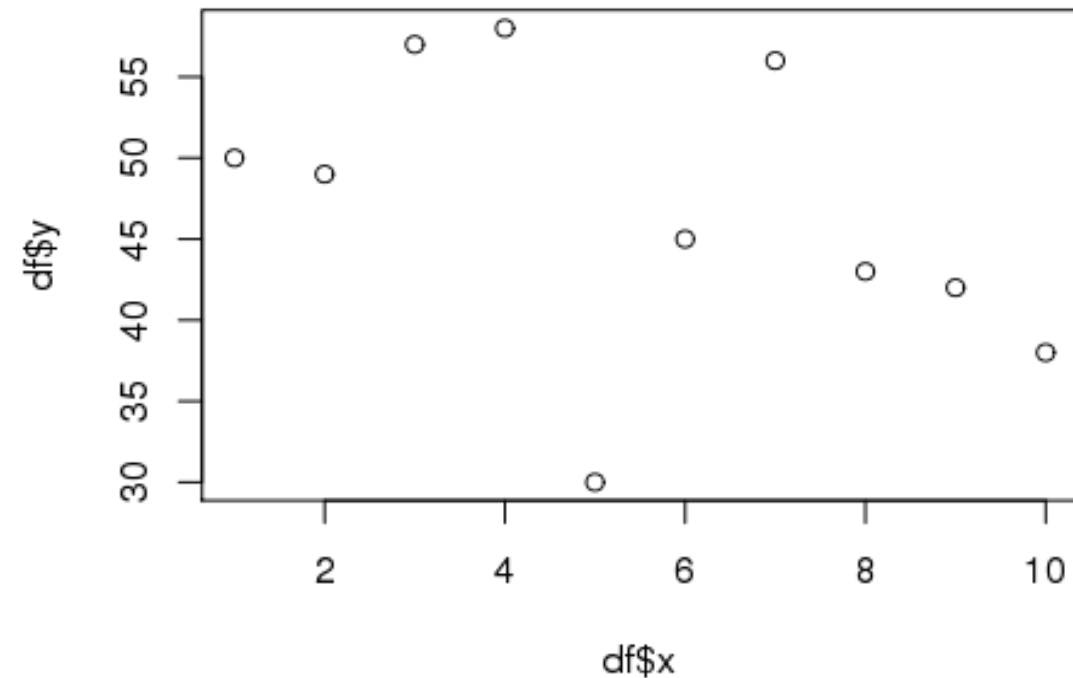
df <- data.frame(x,y,z,zz)
```

Base Example

	x	y	z	zz
1	1	50	Blue	Adult
2	2	49	Blue	Adult
3	3	57	Brown	Child
4	4	58	Brown	Adult
5	5	30	Brown	Adult
6	6	45	Brown	Adult
7	7	56	Blue	Adult
8	8	43	Green	Adult
9	9	42	Brown	Child
10	10	38	Brown	Adult

Base Example

- `plot(df$x, df$y)`



	x	y	z	zz
1	1	50	Blue	Adult
2	2	49	Blue	Adult
3	3	57	Brown	Child
4	4	58	Brown	Adult
5	5	30	Brown	Adult
6	6	45	Brown	Adult
7	7	56	Blue	Adult
8	8	43	Green	Adult
9	9	42	Brown	Child
10	10	38	Brown	Adult

Base Example

```
plot(x, #x axis
     y, #y axis
     xlab="Patient Number",
     ylab="Age",
     pch = 21, #plotting symbol
     mar = c(4, 4, 2, 2), #plot margin size
     col = "black", #border colour for points
     bg = "green", #background colour for points
     ylim = c(25,65)) #y axis limits
```

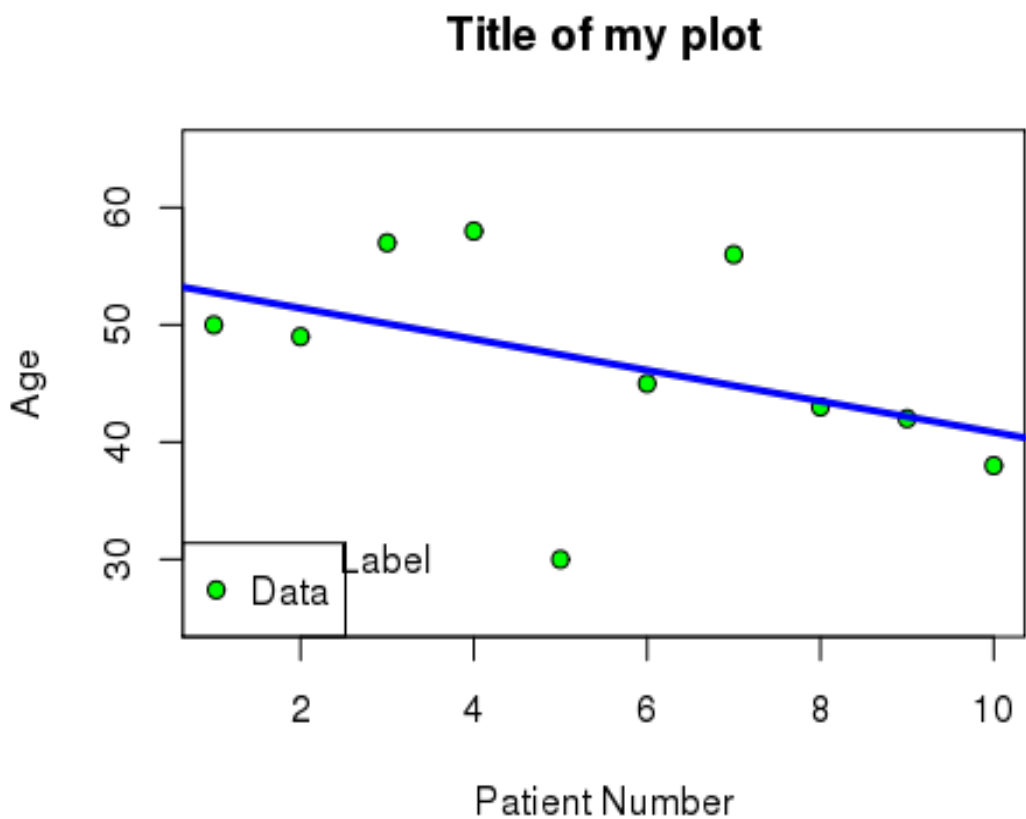
```
fit <- lm(y ~ x) #fit a linear model between variables y and x
abline(fit, #abline is a function that adds a line to the current plot
       lwd = 3, #line width
       col = "blue") #colour of line
```

```
title(main = "Title of my plot") #title of plot
```

```
text(3, 30, "Label") #Label on graph
```

```
legend("bottomleft", #add a legend to plot
      legend = "Data", #title of legend
      pch = 21, #plotting symbol
      pt.bg = "green", #background colour for points
      col = "black") #border colour for points
```

Base Example



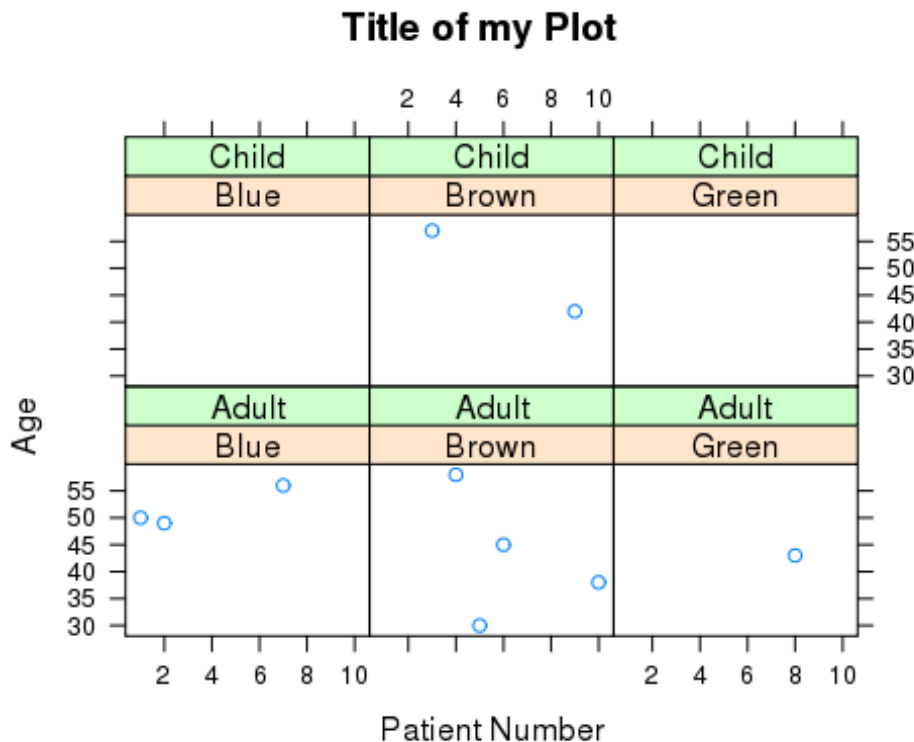
	x	y	z	zz
1	1	50	Blue	Adult
2	2	49	Blue	Adult
3	3	57	Brown	Child
4	4	58	Brown	Adult
5	5	30	Brown	Adult
6	6	45	Brown	Adult
7	7	56	Blue	Adult
8	8	43	Green	Adult
9	9	42	Brown	Child
10	10	38	Brown	Adult

Lattice



Lattice Example

- `xyplot( y~x|z*zz, #plot variables x by y given z`
`data = df, #data`
`main="Title of my Plot", #title`
`ylab="Age", #y axis label`
`xlab="Patient Number") #x axis label`



	x	y	z	zz
1	1	50	Blue	Adult
2	2	49	Blue	Adult
3	3	57	Brown	Child
4	4	58	Brown	Adult
5	5	30	Brown	Adult
6	6	45	Brown	Adult
7	7	56	Blue	Adult
8	8	43	Green	Adult
9	9	42	Brown	Child
10	10	38	Brown	Adult

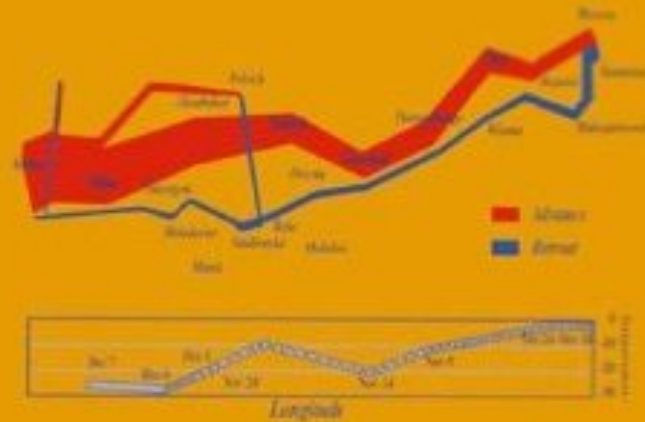
ggplot2

Roche

Statistics and Computing

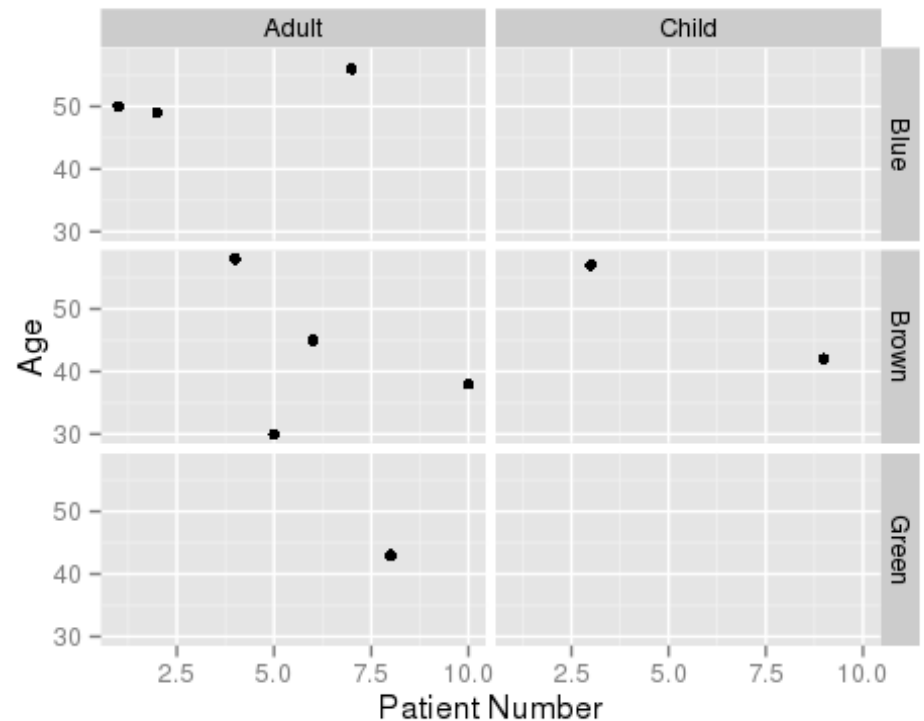
Leland Wilkinson

The Grammar of Graphics



ggplot Example

	x	y	z	zz
1	1	50	Blue	Adult
2	2	49	Blue	Adult
3	3	57	Brown	Child
4	4	58	Brown	Adult
5	5	30	Brown	Adult
6	6	45	Brown	Adult
7	7	56	Blue	Adult
8	8	43	Green	Adult
9	9	42	Brown	Child
10	10	38	Brown	Adult



```
library("ggplot2")
```

```
#qplot (quick plot)
```

```
qplot(x, #xaxis  
      y, #y axis  
      data=df, #data  
      facets=z~zz, #grouping/facets  
      ylab="Age", #y axis label  
      xlab="Patient Number") #x axis label
```

ggplot Example

```
p <- ggplot(df, aes(x=x, y=y, group = z, color = z, shape = zz)) #  
very basic scatterplot
```

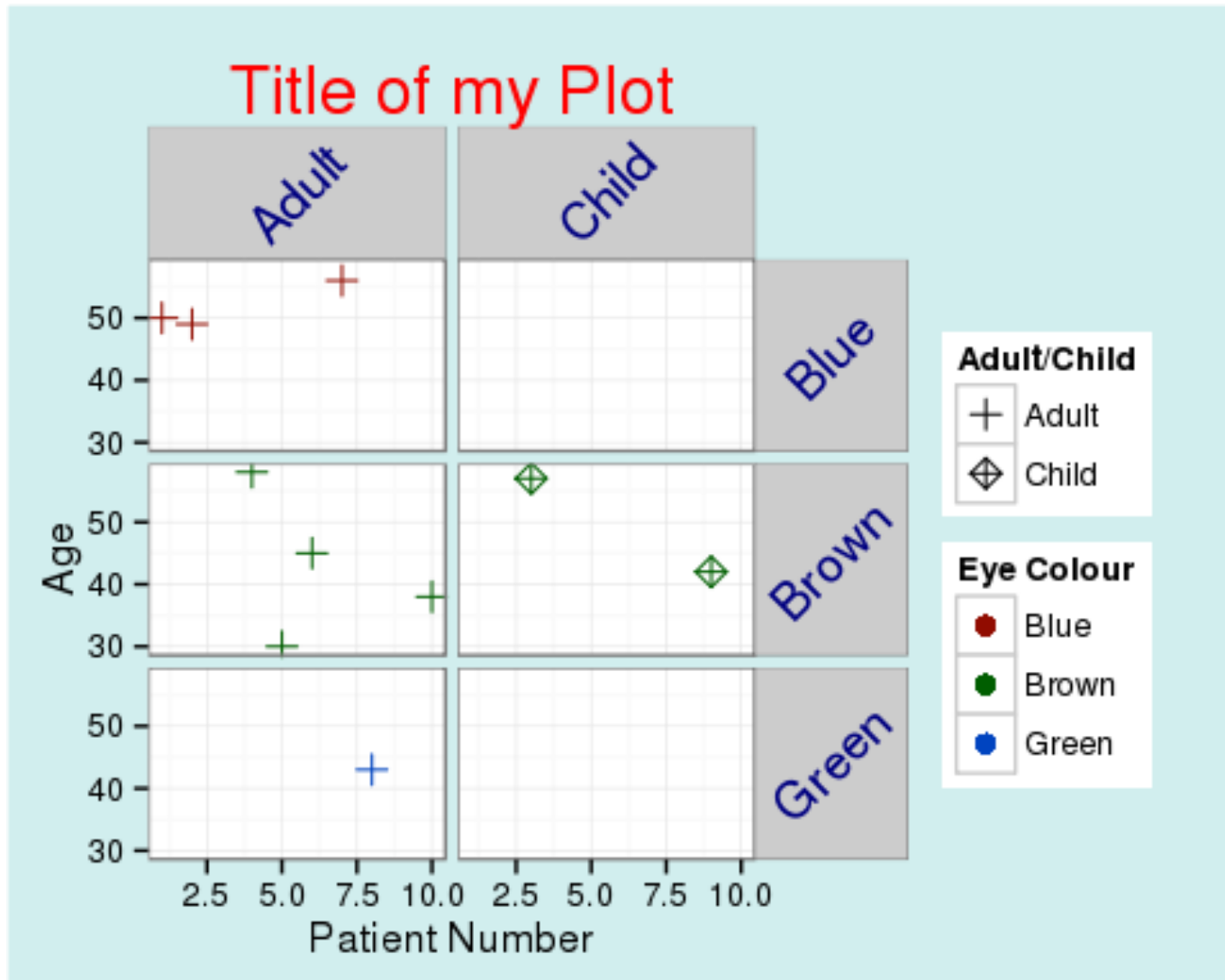
```
p <- p + geom_point(size = 3) #add points  
p <- p + theme_bw() #remove background grid  
p <- p + xlab("Patient Number") #x axis label  
p <- p + ylab("Age") #y axis label  
p <- p + ggtitle("Title of my Plot") #plot title
```

```
p <- p + facet_grid(z ~ zz) #groups/facets  
p <- p + scale_colour_hue(name="Eye Colour", l=30) #Name of first  
Legend and colours to use
```

```
p <- p + scale_shape_manual(name="Adult/Child", values= c(3,9)) #n  
ame of second legend and symbols to use  
p <- p + theme(strip.text.x = element_text(size = 15, colour = "na  
vy", angle = 45)) #alter text aesthetics for x axis facets  
p <- p + theme(strip.text.y = element_text(size = 15, colour = "na  
vy", angle = 45)) # alter text aesthetics for u axis facets  
p <- p + theme(plot.title = element_text(colour = "red", size = 20  
)) #alter title aesthetics
```

```
p <- p + theme(plot.background = element_rect(fill = "lightcyan2")  
 ) #alter plot background colour
```

ggplot Example



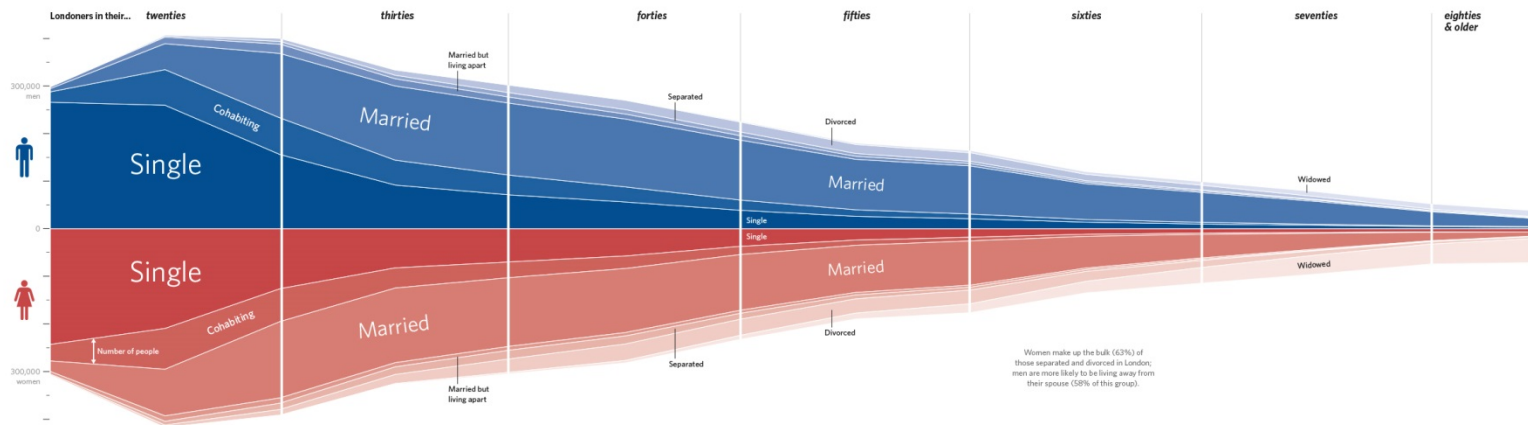
London Cycle Hire Journeys

Relationship Status

We age, we marry, we migrate.

Twenty-five and single? In London, you're anything but alone. According to the 2011 Census, more than half of twentysomethings go solo. Then come your thirties and that deep river of available singles your age narrows to a shallow stream. By age fifty, it's little more than a trickle.

Most will have married and settled in the leafier boroughs (see maps). Young singles – and their cohabiting friends – prefer trendier parts of the city such as Hoxton, Clapham and Peckham. Those living away from their spouse fall into two camps: pricey places in Kensington or more affordable housing near Heathrow, Wembley and Upton Park. Separated couples and divorcees cluster around Tottenham and to the east; widowed spouses stay put.



Areas with above average concentrations of each relationship status, 2011

- Well above average
- Above average
- Average
- Below average



WHO WE ARE

SOURCES: 2011 CENSUS, ONS; OS

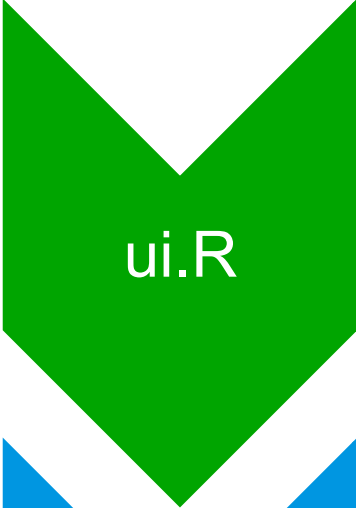
©

Excerpted from *London: The Information Capital* by James Cheshire and Oliver Uberti (Particular Books, 30 October 2014)

data: 3.2 million journeys (from TFL)
 Routing: Ollie O'Brien (@oobr) + OpenStreetMap cc-by-sa
 Buildings: OS Opendata Crown Copyright 2011
 Map: James Cheshire (@spatialanalysis)



Shiny App



ui.R

- user interface script controls the layout and appearance of your app



server.R

- server script contains the instructions that your computer needs to build your app

As an example I have loaded dummy SDTM.DM and SDTM.PC datasets, however this would usually be able to read from BCE

Instructions when using this in BCE

Before using this R Shiny app for the first time you will need to make sure you have run the following code and followed the instructions in BCE:

```
'rsync -a ~/.ssh rkav792610:'
```

Once you have done this, please select your data below.

Data set selection:

Root level:

SDTM ▼

CDP or Study number:

/srv/shiny-server/public/_SHINY_COMPETITION/nixonc

PC Data Set name:

pc ▼

DM Data Set name:

dm ▼

About this R Shiny application

This R Shiny Application has been created to help you check your SDTM PK data so that it is ready for you to start programming with, and load into PKS.

The aim of this app is to get the programmer to visualize the SDTM data using plots and specific data tables in order to further understand the data that has been sent.

My hope is that this concept can be applied to other SDTM's, and then onto VAD's.

Data Checker

[PC Data Checks](#)[Comments in the Dataset](#)[Plot of VISIT against PCTPTREF](#)[Plot of PCTPT against PCTPTNUM](#)[By Patient Plot of VISIT against PCSTRESN](#)

Summary

Total Number of Patients/Subjects in SDTM.PC

9

Total Number of non-missing values in the variable VISITDY =

0

Total Number of MISSING values in the variable VISITDY

164

The unique values in the variable PCCAT are:

- ANALYTE

The unique values in the variable PCTEST are:

- RO5545965-000
-

The unique values in the variable PCSPEC are:

- Plasma
-

Data Checker

- PC Data Checks
- Comments in the Dataset
- Plot of VISIT against PCTPTREF
- Plot of PCTPT against PCTPTNUM
- By Patient Plot of VISIT against PCSTRESN

25

▼

 records per page

Search:

USUBJID	PCREFID	PCTEST	PCCAT	PCSCAT	VISIT	PCTPT	COVAL	COVAL1	COVAL2	PCREASND
BP28844-261732-1001	101	RO5545965-000	ANALYTE	PK Sampling Information	Day 1	PREDOSE				
BP28844-261732-1001	102	RO5545965-000	ANALYTE	PK Sampling Information	Day 1	1.5 H				
BP28844-261732-1001	103	RO5545965-000	ANALYTE	PK Sampling Information	Day 1	2 H				
BP28844-261732-1001	104	RO5545965-000	ANALYTE	PK Sampling Information	Day 1	3 H				
BP28844-261732-1001	105	RO5545965-000	ANALYTE	PK Sampling Information	Day 1	4 H	SAMPLE NUMBER 105 TAKEN			

Data Checker

[PC Data Checks](#)
[Comments in the Dataset](#)
[Plot of VISIT against PCTPTREF](#)

[Plot of PCTPT against PCTPTNUM](#)

[By Patient Plot of VISIT against PCSTRESN](#)

Choose patients to display

- ☐ 1001
- ☐ 1002
- ☐ 2001
- ☐ 2002
- ☐ 3001
- ☐ 3002
- ☐ 4001
- ☐ 4002
- ☐ 4003

Individual Patient plot, Visit by PK Concentration

PK Concentration

Visit

Data Checker

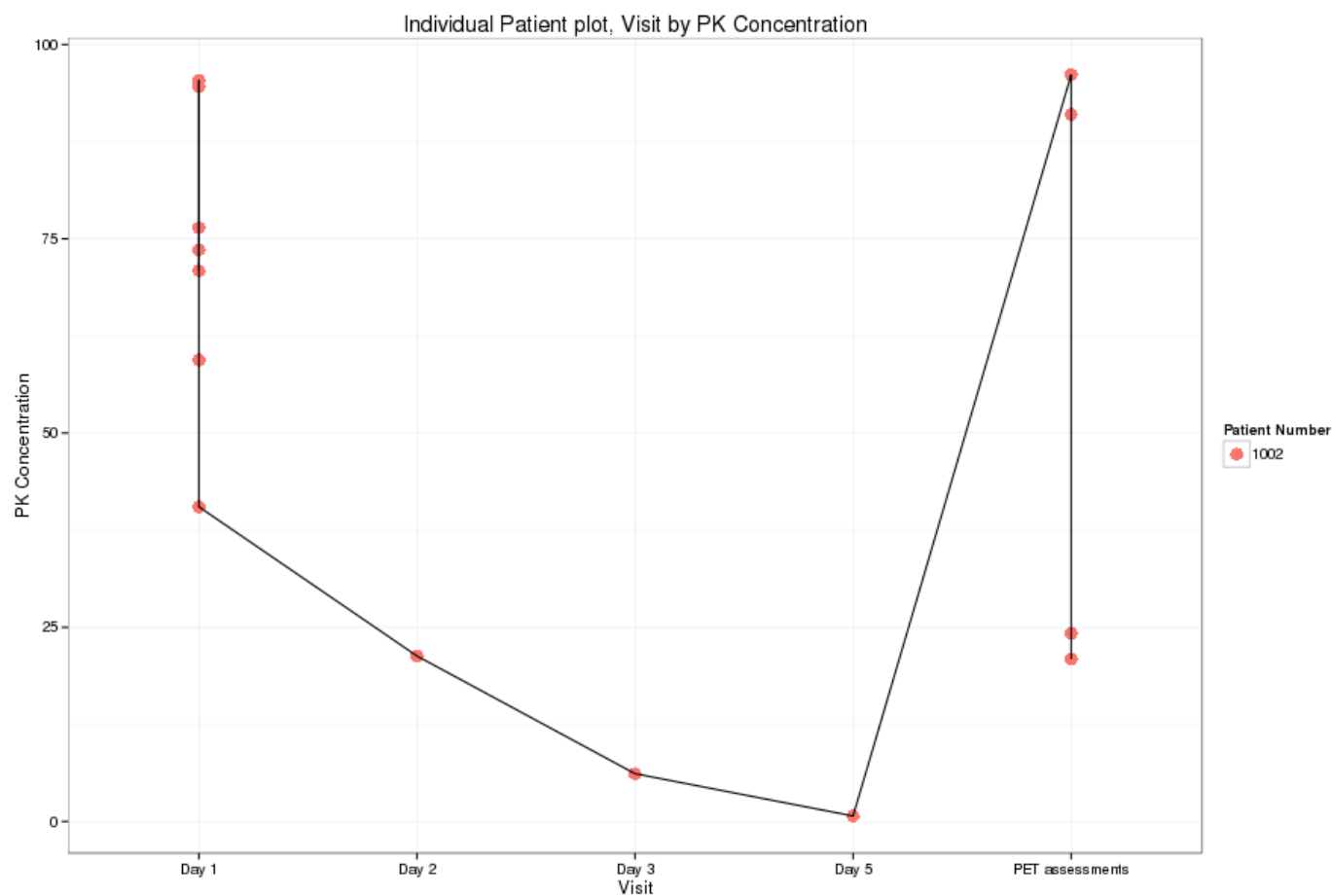
[PC Data Checks](#)
[Comments in the Dataset](#)
[Plot of VISIT against PCTPTREF](#)

[Plot of PCTPT against PCTPTNUM](#)

[By Patient Plot of VISIT against PCSTRESN](#)

Choose patients to display

- ☐ 1001
- ☒ 1002
- ☐ 2001
- ☐ 2002
- ☐ 3001
- ☐ 3002
- ☐ 4001
- ☐ 4002
- ☐ 4003



Data Checker

PC Data Checks

Comments in the Dataset

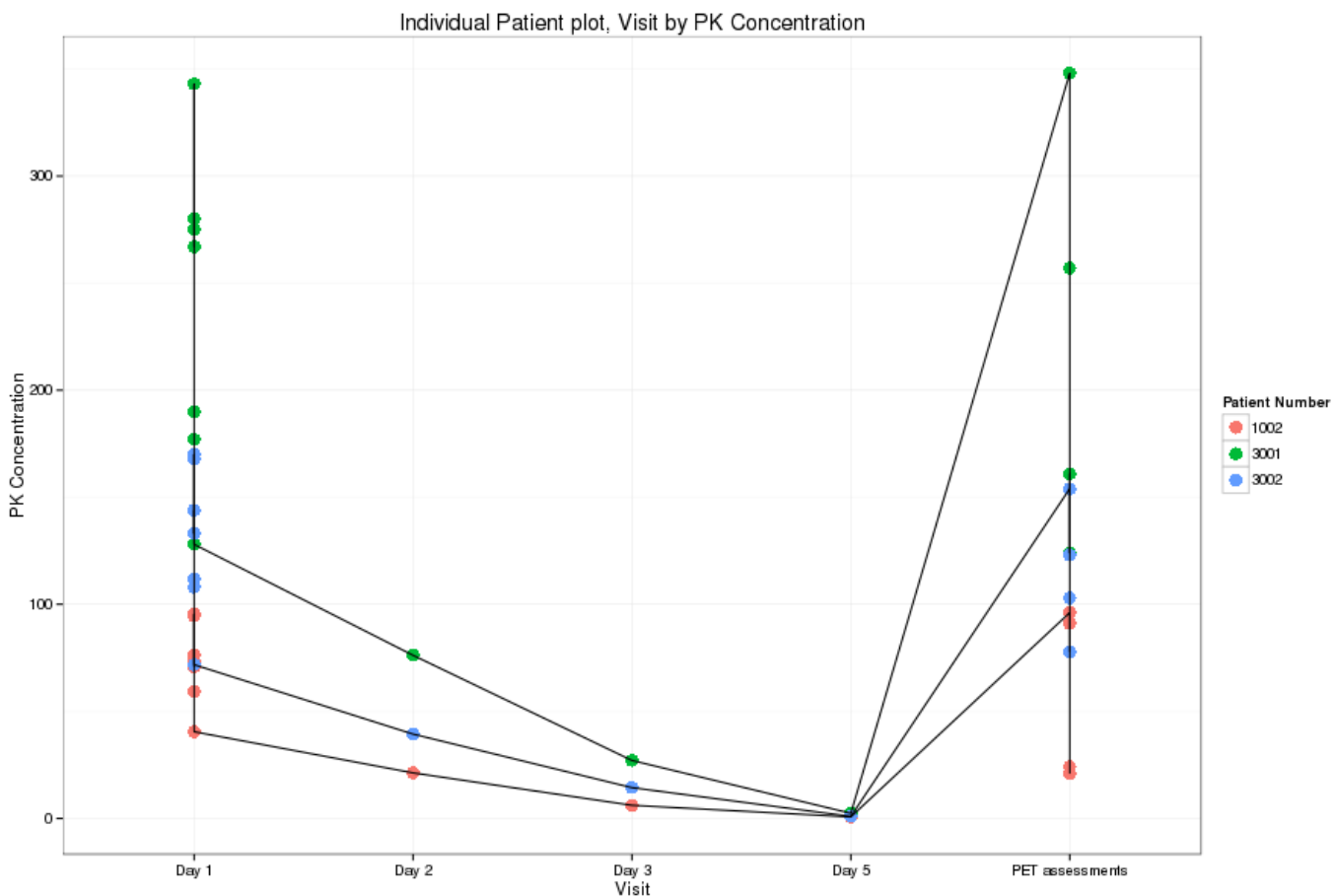
Plot of VISIT against PCTPTREF

Plot of PCTPT against PCTPTNUM

By Patient Plot of VISIT against PCSTRESN

Choose patients to display

- ☐ 1001
☒ 1002
☐ 2001
☐ 2002
☒ 3001
☒ 3002
☐ 4001
☐ 4002
☐ 4003



Shiny Examples

- [Rotten Tomatos](#)
- [Word Cloud](#)

Summary



A word cloud featuring various R plotting packages and concepts. The words are arranged in a dense, overlapping manner. The colors of the words include shades of purple, blue, teal, and grey. The words are: ggplot, Plots, Interactive, Statistical Analysis, Base, Grammar Of Graphics, R, Customize, ggplot2, Flexible, qplot, Lattice, and Shiny.

ggplot Plots
Interactive
Statistical Analysis R
Base Grammar Of Graphics
Customize ggplot2
Flexible qplot Lattice
Shiny

References

- [Coursera – R Course](#)
- [R Shiny Website with tutorial](#)
- [R Shiny Gallery](#)
- [RStudio](#)
- [Base R](#)
- [Lattice Package](#)
- [Ggplot2 Package](#)

***Doing now what patients need
next***