

WELCOME!

Follow

Here you will find daily news and tutorials about R, contributed by over 573 bloggers.

There are many ways to follow us -
By e-mail:

Your e-mail here

Subscribe

19386 readers
BY FEEDBURNER

On Facebook:



R blogge..
26k likes

Like Page

Be the first of your friends to like this



If you are an R blogger yourself you are invited to add your own R content feed to this site (Non-English R bloggers should add themselves- [here](#))

JOBS FOR R-USERS

Quantitative Analyst in Rain & Hail @ Des Moines, Iowa
Research Assistant Position (Pre-doc, 2 years -optionally Post-doc) - @ Wien, Austria
Data Scientist @ Tel Aviv
Data Scientist for Agoda (@ Tel Aviv)
DATA APPLICATIONS ENGINEER - MACHINE LEARNING (@ Bangkok)

Search & Hit Enter

POPULAR SEARCHES

heatmap
web scraping
maps
undefined
hadoop
shiny
twitter
boxplot

Line plots of longitudinal summary data in R using ggplot2

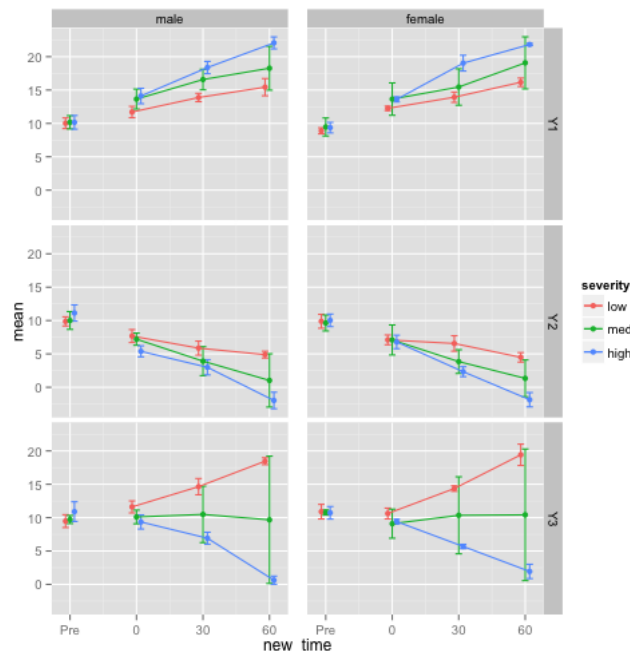
July 9, 2015

By [strictlystat](#)

Like Share 121

(This article was first published on [A HopStat and Jump Away » Rbloggers](#), and kindly contributed to [R-bloggers](#))

I recently had an email for a colleague asking me to make a figure like this in ggplot2 or trellis in R:



As I know more about how to do things in ggplot2, I chose to use that package (if it wasn't obvious from the plot or other posts).

Starting Point

Cookbook R/ has a great starting point for making this graph. The solution there is not sufficient for the desired graph, but that may not be clear why that is. I will go through most of the steps of customization on how to get the desired plot.

Creating Data

To illustrate this, I will create some sample dataset:

TOP 3 POSTS FROM THE PAST 2 DAYS

In-depth introduction to machine learning in 15 hours of expert videos
Building Wordclouds in R
Installing R packages

Search & Hit Enter

TOP 9 ARTICLES OF THE WEEK

1. In-depth introduction to machine learning in 15 hours of expert videos
2. Installing R packages
3. Using apply, sapply, lapply in R
4. Basics of Histograms
5. How to use lists in R
6. How R is used at Zillow to estimate housing values
7. Scatterplots
8. Plotting Time Series in R using Yahoo Finance data
9. Read Excel files from R

SPONSORS



Highland Statistics Ltd

Zero Inflated Models & GLMM
Beginner's Guide to GAM
Beginner's Guide to GLM & GLMM
Beginner's Guide to GAMM



trading
ggplot2
animation
time series
finance
latex
excel
ggplot
googlevis
pca
RStudio
quantmod
eclipse
market research
tutorial
how to import image file to R
alt=
rattle
knitr
coplot
remdr
gis

RECENT POSTS

RcppGSL 0.3.0
"A 99% TVaR is generally a 99.6% VaR"
Building Wordclouds in R
Bio7.2.3 Released!
Get to know Cortana
Analytics: Workshop and webinars
The IQUIT R video series
Legoplots in R (3D barplots in R)
Plotting Time Series in R using Yahoo Finance data
New release: Choroplethr v3.2.0
Coloring (and Drawing) Outside the Lines in ggplot
Measuring business health with Delta LifeCycle Grids and R
Spatio-Temporal Kriging in R
abcf 0.9-3
R packages with unlimited licenses
New R titles available in Chinese

OTHER SITES

Statistics of Israel
SAS blogs
Jobs for R-users

```
1 N <- 30
2 id <- as.character(1:N) # create ids
3 sexes = c("male", "female")
4 sex <- sample(sexes, size = N/2, replace = TRUE) #
5 diseases = c("low", "med", "high")
6 disease <- rep(diseases, each = N/3) # disease severity
7 times = c("Pre", "0", "30", "60")
8 time <- rep(times, times = N) # times measured
9 t <- 0:3
10 ntimes = length(t)
11 y1 <- c(replicate(N/2, rnorm(ntimes, mean = 10+2*t,
12 replicate(N/2, rnorm(ntimes, mean = 10+4*t,
13 y2 <- c(replicate(N/2, rnorm(ntimes, mean = 10-2*t,
14 replicate(N/2, rnorm(ntimes, mean = 10-4*t,
15 y3 <- c(replicate(N/2, rnorm(ntimes, mean = 10+t^2,
16 replicate(N/2, rnorm(ntimes, mean = 10-t^2,
17
18 data <- data.frame(id=rep(id, each=ntimes), sex=rep(sex, each=ntimes),
19 severity=rep(disease, each=ntimes), Y1=c(y1), Y2=c(y2), Y3=c(y3)) #
20
21 ##### factor the variables so in correct order
22 data$sex = factor(data$sex, levels = sexes)
23 data$time = factor(data$time, levels = times)
24 data$severity = factor(data$severity, levels = diseases)
25 head(data)
```

	id	sex	severity	time	Y1	Y2
1	1	female	low	Pre	9.262417	11.510636
2	1	female	low	0	10.223988	8.592833
3	1	female	low	30	13.650680	5.696405
4	1	female	low	60	15.528288	5.313968
5	2	female	low	Pre	9.734716	11.190081
6	2	female	low	0	12.892207	7.897296

We have a longitudinal dataset with 30 different people/units with different ID. Each ID has a single sex and disease severity. Each ID has 4 replicates, measuring 3 separate variables (Y1, Y2, and Y3) at each time point. The 4 time points are previous (Pre)/baseline, time 0, 30, and 60, which represent follow-up.

Reformatting Data

In ggplot2, if you want to plot all 3 Y variables, you must have them in the same column, with another column indicating which variable you want plot. Essentially, I need to make the data "longer". For this, I will reshape the data using the `reshape2` package and the function `melt`.

```
1 library(reshape2)
2 long = melt(data, measure.vars = c("Y1", "Y2", "Y3"))
3 head(long)
```

	id	sex	severity	time	variable	value
1	1	female	low	Pre	Y1	9.262417
2	1	female	low	0	Y1	10.223988
3	1	female	low	30	Y1	13.650680
4	1	female	low	60	Y1	15.528288
5	2	female	low	Pre	Y1	9.734716
6	2	female	low	0	Y1	12.892207

It may not be clear what has been reshaped, but reordering the data.frame can illustrate that each Y variable is now a separate row:

```
1 head(long[ order(long$id, long$time, long$variable)])
```

	id	sex	severity	time	variable	value
1	1	female	low	Pre	Y1	9.262417
2	121	1 female	low	Pre	Y2	11.510636
3	241	1 female	low	Pre	Y3	9.047127
4	2	1 female	low	0	Y1	10.223988
5	122	1 female	low	0	Y2	8.592833
6	242	1 female	low	0	Y3	11.570381
7	3	1 female	low	30	Y1	13.650680
8	123	1 female	low	30	Y2	5.696405
9	243	1 female	low	30	Y3	13.954316
10	4	1 female	low	60	Y1	15.528288
11						

Creating Summarized data frame

We will make a data.frame with the means and standard deviations for each group, for each sex, for each Y variable, for separate time points. I will use `plyr` to create this data.frame, using `ddply` (first d representing I'm putting in a data.frame, and the second d representing I want data.frame out):



REVOLUTION
ANALYTICS

R
for the Enterprise

www.revolutionanalytics.com

Werden Sie zum Expe[R]ten mit der
R-Akademie von



Beratung | Software
Training | Lösungen



Plotly: collaborative, publication-quality graphing.

STATISTICS
VIEWS
Bringing Statistics Together

NYC DATA SCIENCE
ACADEMY
Sep 21 - Dec 11 | FULL TIME PROGRAM
12 - WEEK DATA SCIENCE BOOTCAMP
Hands-on R, Python & Hadoop Training
Mentorship from Top Data Scientists
Job Placement at Our 300+ Hiring Firms
Scholarship & Financial Aid Available
DEADLINE FOR APPLICATION: August 21st

DataCamp
Learn R Interactively

#ODSC 50 Talks
14 Workshops
6 Pre-conference Workshops
SAN FRANCISCO | NOV. 14TH - 15TH

```

1 library(plyr)
2 agg = ddply(long, .(severity, sex, variable, time),
3             c(mean=mean(x$value), sd = sd(x$value))
4             })
5 head(agg)

```

```

1 severity sex variable time mean sd
2 1 low male Y1 Pre 9.691420 1.1268324
3 2 low male Y1 0 12.145178 1.1218897
4 3 low male Y1 30 14.304611 0.3342055
5 4 low male Y1 60 15.885740 1.7616423
6 5 low male Y2 Pre 9.653853 0.7404102
7 6 low male Y2 0 7.652401 0.7751223

```

There is nothing special about means/standard deviations. It could be any summary measures you are interested in visualizing.

We will also create the Mean + 1 standard deviation. We could have done standard error or a confidence interval, etc.

```

1 agg$lower = agg$mean + agg$sd
2 agg$upper = agg$mean - agg$sd

```

Now, agg contains the data we wish to plot.

Time is not on your side

Time as a factor

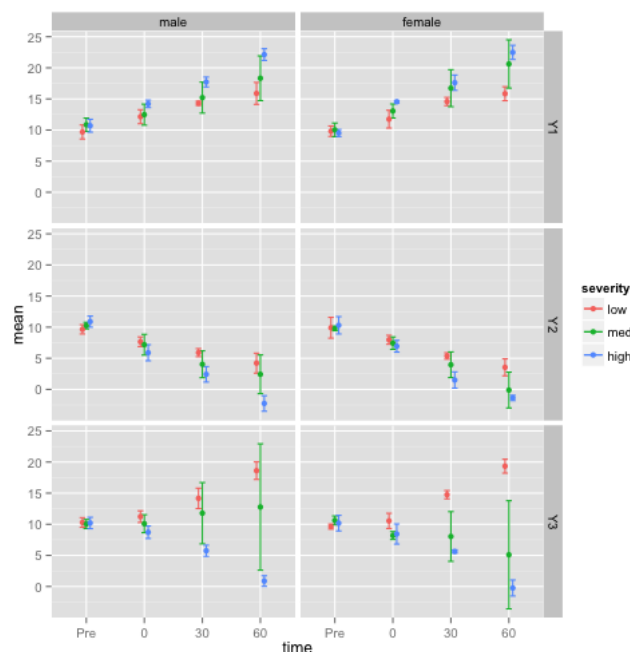
If you look at the plot we wish to make, we want the lines to be connected for times 0, 30, 60, but not for the previous data. Let's try using the time variable, which is a factor. We create pd, which will be a ggplot2 object, which tells that I wish to plot the means + error bars slightly next to each other.

```

1 class(agg$time)
2 [1] "factor"

1 pd <- position_dodge(width = 0.2) # move them .2 to
2
3 gbase = ggplot(agg, aes(y=mean, colour=severity)) +
4   geom_errorbar(aes(ymin=lower, ymax=upper), width=
5   geom_point(position=pd) + facet_grid(variable ~ s
6   gline = gbase + geom_line(position=pd)
7   print(gline + aes(x=time))

```



None of the lines are connected! This is because time is a factor. We will use gbase and gline with different times to show how the end result can be achieved.

Time as a numeric

We can make time a numeric variable, and simply replace Pre with -1 so that it can be plotted as well.

```

1 agg$num_time = as.numeric(as.character(agg$time))
2 agg$num_time[ is.na(agg$num_time) ] = -1
3 unique(agg$num_time)

```

Save **25%** on R Books

CRC Press
Taylor & Francis Group

Use Promo Code **CZP40**

Plus Free Shipping

Data Analysis Modeling

Cloud Solution

Free Sign Up!

STATWORX

Consulting
Schulung
Data Mining

Mehr erfahren

Try the FASTEST ML for R

Click for a Free Trial

Become a **Certified Data Scientist**

ENROLL NOW

simplilearn

DATA FESTIVAL

SEPT. 18 - 23RD

DATA SCIENCE DAY
BIG DATA DAY
INTERNET OF THINGS DAY

SIX

SIX

EARN YOUR MASTER'S IN DATA SCIENCE online.

Berkeley
UNIVERSITY OF CALIFORNIA

LEARN MORE

PE, DOWNSHUT, RICE, LINE, BAR, AREA, TREEMAP, GRAPH, FUNNEL, MAP

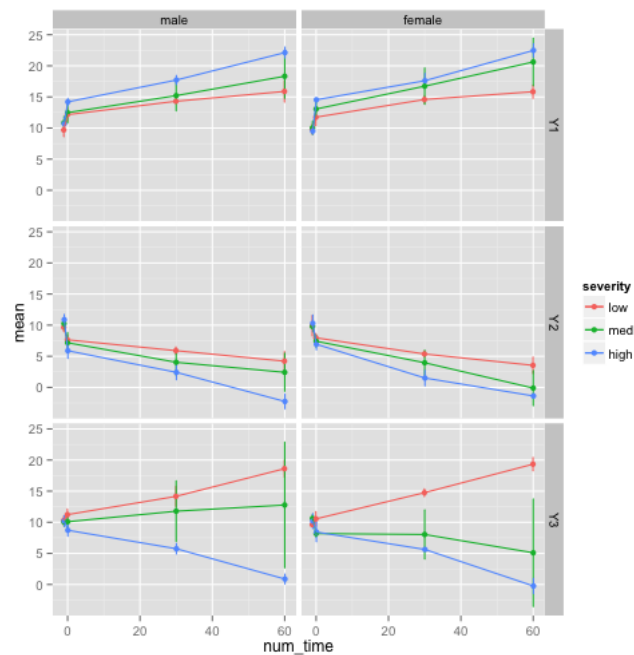
Search & Hit Enter


```
1 [1] -1 0 30 60
```

Full list of contributing R-bloggers

In a [previous post](#), I have discussed as an aside of creating a plot in `ggplot2` and then creating adding data to the `data.frame`. You must use the `%%` to update the data in the object.

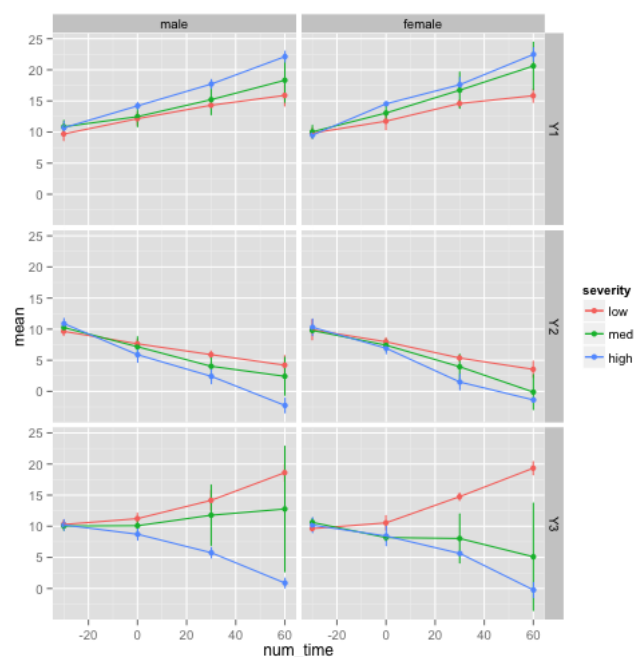
```
1 gline = gline %% agg
2 print(gline + aes(x=num_time))
```



If you look closely, you can see that `Pre` and time 0 are very close and not labeled, but also connected. As the scale on the x-axis has changed, the width of the error bar (set to 0.3), now is too small and should be changed if using this solution.

Although there can be a discussion if the `Pre` data should be even on the same plot or the same timeframe, I will leave that for you to dispute. I don't think it's a terrible idea, and I think the plot works because the `Pre` and 0 time point data are not connected. There was nothign special about -1, and here we use -30 to make it evenly spaced:

```
1 agg$num_time[ agg$num_time == -1 ] = -30
2 gline = gline %% agg
3 print(gline + aes(x=num_time))
```



That looks similar to what we want. Again, `Pre` is connected to the data, but we also now have a labeling problem with the x-axis somewhat. We still must change the width of the error bar in this scenario as well.

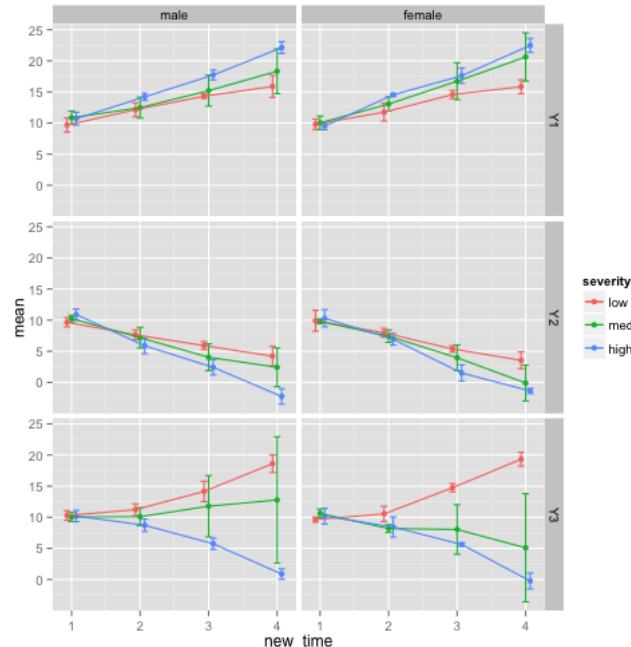
Time as a numeric, but not the actual time point

In the next case, we simply use `as.numeric` to the factor to create a variable `new_time` that will be 1 for the first level of `time` (in this case `Pre`) to the number of time points, in this case 4.

```
1 | agg$new_time = as.numeric(agg$time)
2 | unique(agg$new_time)

1 | [1] 1 2 3 4

1 | gline = gline %>% agg
2 | print(gline + aes(x = new_time))
```



Here we have something similar with the spacing, but now the labels are not what we want. Also, `Pre` is still connected. The width of the error bars is now on a scale from 1-4, so they look appropriate.

Creating a Separate data.frame

Here, we will create a separate `data.frame` for the data that we want to connect the points. We want the times 0-60 to be connected and the `Pre` time point to be separate.

```
1 | sub_no_pre = agg[ agg$time != "Pre",]
```

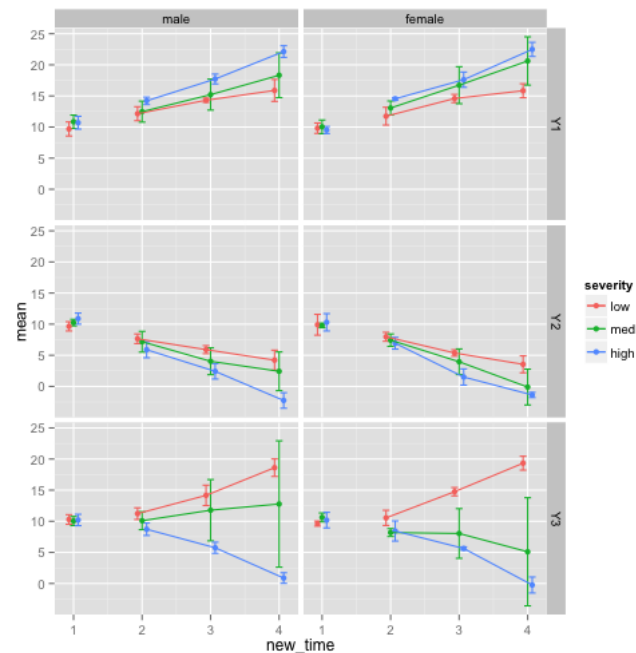
Multiple data sets in plot function

Note, previously we did:

```
1 | gline = gbase + geom_line(position=pd)
```

This assumes that `geom_line` uses the same `data.frame` as the rest of the plot (`agg`). We can fully specify the arguments in `geom_line` so that the line is only for the non-`Pre` data:

```
1 | gbase = gbase %>% agg
2 | gline = gbase + geom_line(data = sub_no_pre, position=pd,
3 |                           aes(x = new_time, y = mean))
4 | print(gline + aes(x = new_time))
```



Note, the arguments in `aes` should match the rest of the plot for this to work smoothly and correctly.

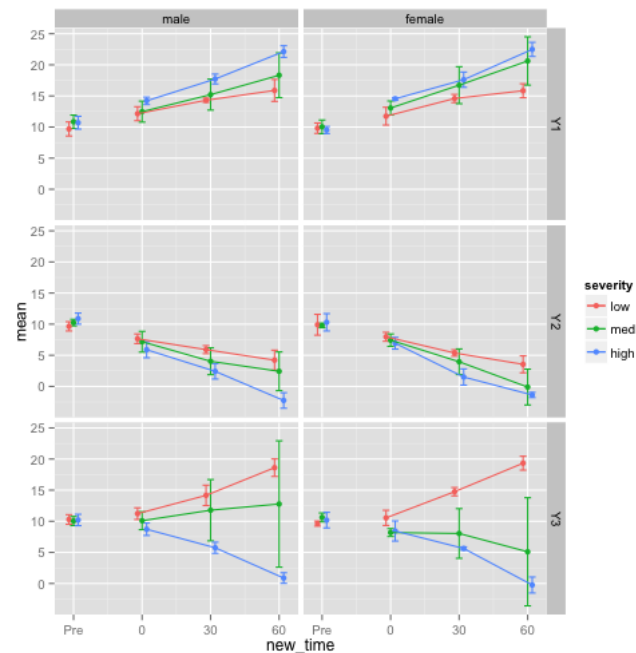
Relabeling the axes

Now, we simply need to re-label the x-axis so that it corresponds to the correct times:

```
1 g_final = gline + aes(x=new_time) +
2   scale_x_continuous(breaks=c(1:4), labels=c("Pre",
```

We could be more robust in this code, using the levels of the factor:

```
1 time_levs = levels(agg$time)
2 g_final = gline + aes(x=new_time) +
3   scale_x_continuous(
4     breaks= 1:length(time_levs),
5     labels = time_levs)
6 print(g_final)
```



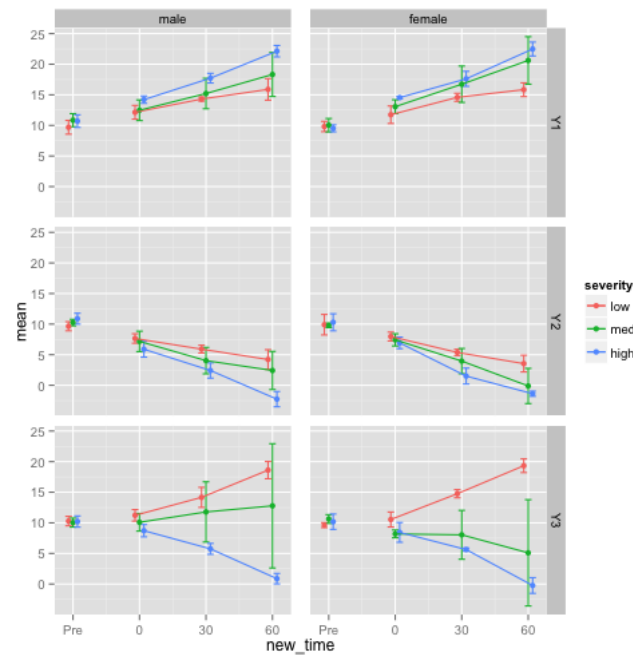
Give me a break

My colleague also wanted to separate the panels a bit. We will use the `panel.margin` arguments and use the `unit` function from the `grid` package to define how far apart we want the axes.

```

1 library(grid)
2 g_final = g_final + theme(panel.margin.x = unit(1, '
3                               panel.margin.y = unit(0.5,
4 print(g_final)

```



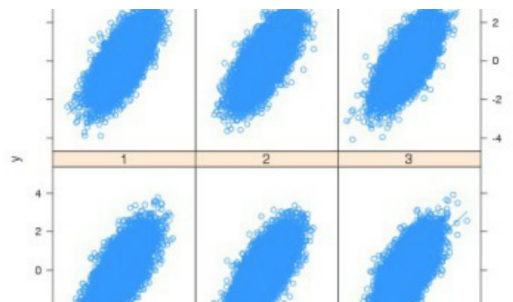
Additional options and conclusoin

I believe legends should be inside a plot for many reasons (I may write about that). Colors can be changed (see [scale_colour_manual](#)). Axis labels should be changed, and the Y should be labeled to what they are (this is a toy example).

Overall, this plot seems to be what they wanted and the default options work okay. I hope this illustrates how to customize a ggplot to your needs and how you may need to use multiple `data.frames` to achieve your desired result.

Comments: 5

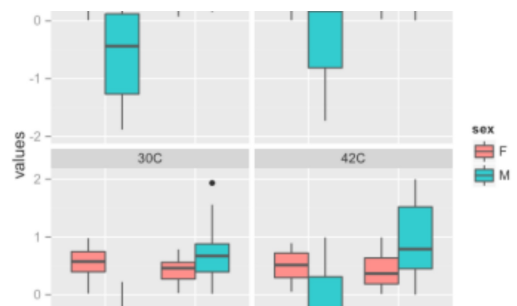
Related



Lattice when modeling, ggplot when publishing
In "R bloggers"



Normality and Testing for Normality
In "R bloggers"



ggplot2 multiple boxplots with metadata
In "R bloggers"

👍 121

👍 Like

🔗 Share

To **leave a comment** for the author, please follow the link and comment on his blog: [A HopStat and Jump Away » Rbloggers](#).

R-bloggers.com offers **daily e-mail updates** about R news and tutorials on topics such as: visualization (ggplot2, Boxplots, maps, animation), programming (RStudio, Sweave, LaTeX, SQL, Eclipse, git, hadoop, Web Scraping) statistics (regression, PCA, time series, trading) and more...

If you got this far, why not **subscribe for updates** from the site?
Choose your flavor: e-mail, twitter, RSS, or facebook...

👍 Like 🔗 Share 🗨 121

Comments are closed.