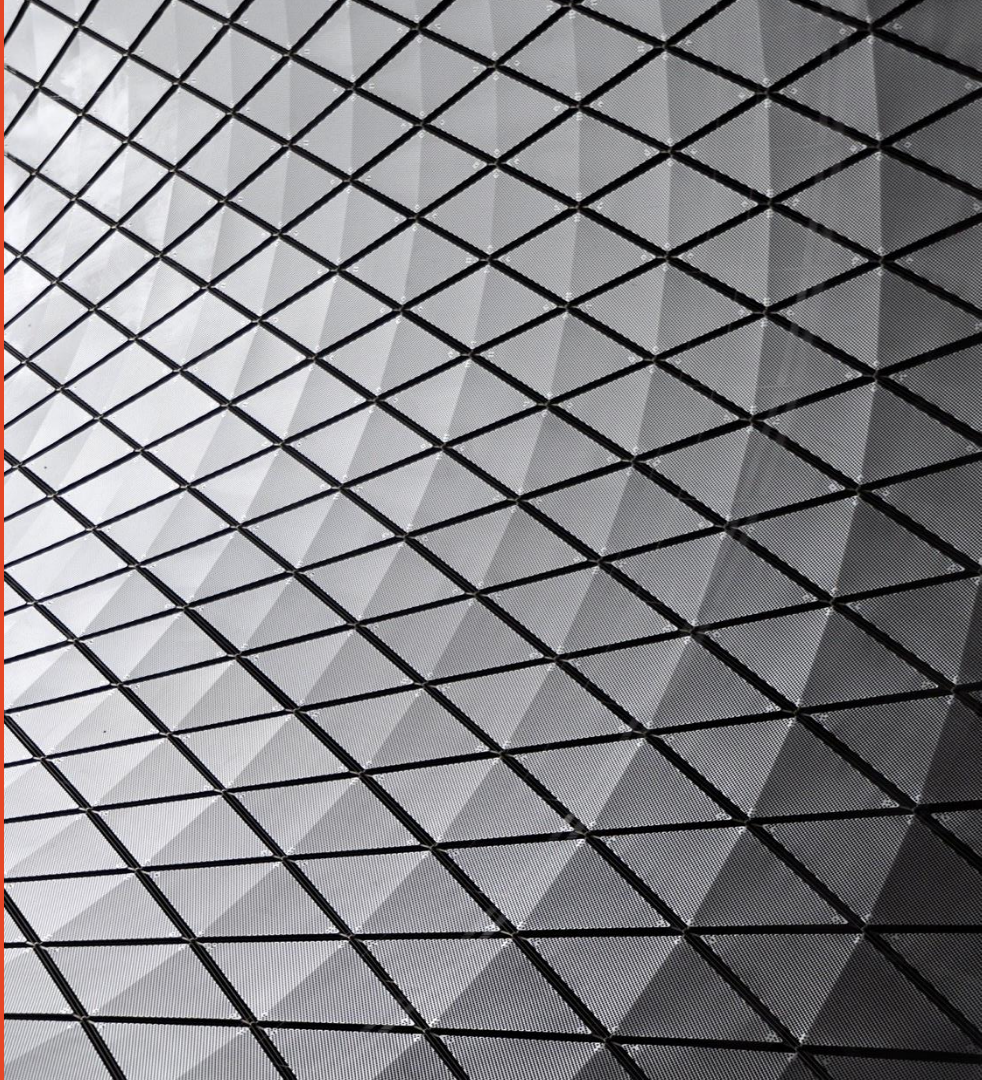
Data management and graphical representation

Workshop: Analysis of Longitudinal Data

12th Nov 2024

Jaroslaw Harezlak
Armando Teixeira-Pinto

THE UNIVERSITY OF
SYDNEY



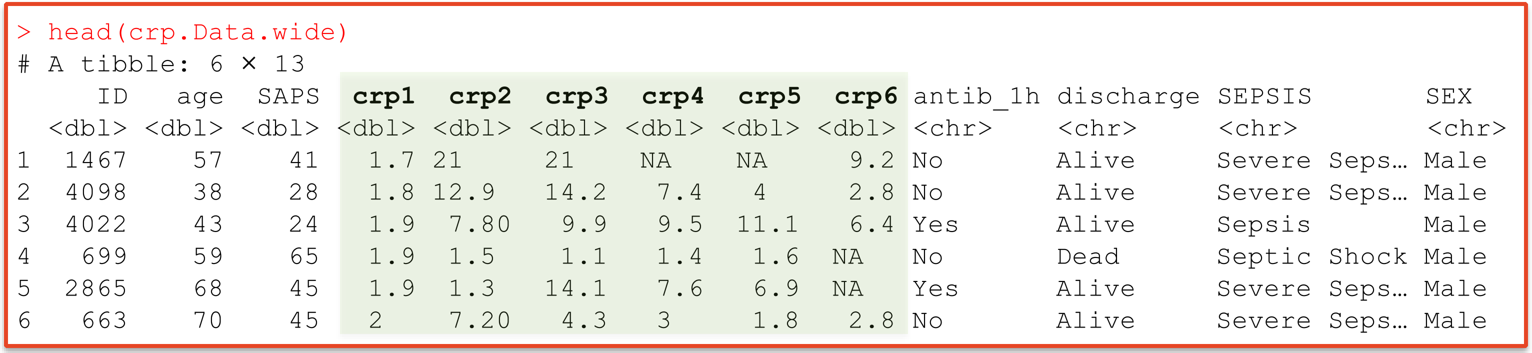
MATERIAL

- http://tinyurl.com/USYD2024

Data

- crp2.csv
- **120 patients admitted to intensive care (ICU) with sepsis.**
- **C-reactive protein** (CRP) is a marker of inflammation and, potentially, a marker of sepsis resolution.
- CRP was measured **over 6 days** or until discharge/death
- Other variables:
  - SAPS - severity score measured within the first 24h in the ICU
  - age - age at admission
  - SEX - Male / Female
  - SEPSIS - category of sepsis (Sepsis/Severe Sepsis/Septic Shock)
  - antib_1h - was antibiotherapy administer within the first hour in ICU (No/Yes)

Wide vs Long format



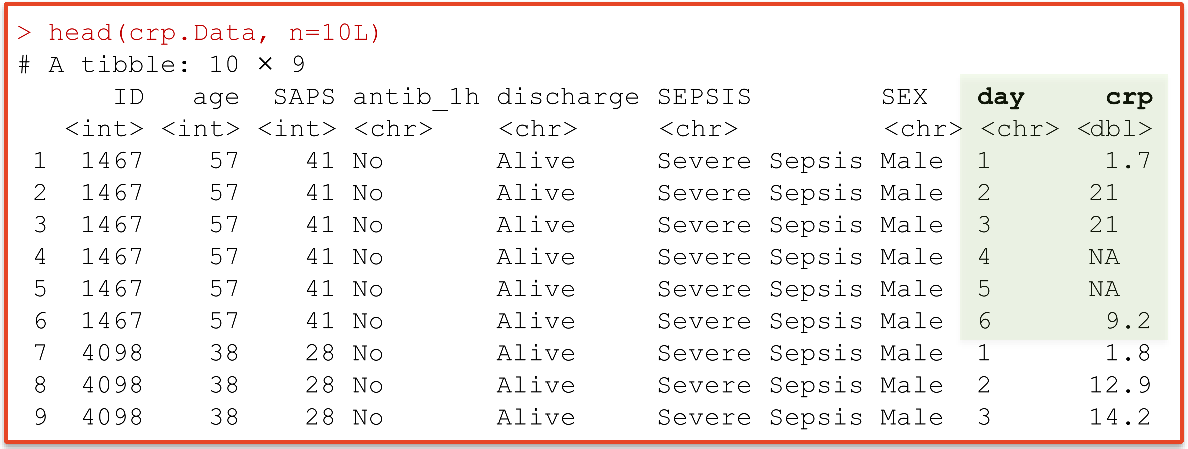
```
> head(crp.Data.wide)
# A tibble: 6 × 13
     ID   age  SAPS  crp1  crp2  crp3  crp4  crp5  crp6 antib_1h discharge SEPSIS       SEX  
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>    <chr>     <chr>        <chr>
1  1467    57    41   1.7 21     21      NA    NA    9.2 No       Alive     Severe Seps… Male 
2  4098    38    28   1.8 12.9   14.2   7.4   4      2.8 No       Alive     Severe Seps… Male 
3  4022    43    24   1.9  7.80   9.9   9.5  11.1    6.4 Yes      Alive     Sepsis       Male 
4   699    59    65   1.9  1.5    1.1   1.4   1.6  NA    No       Dead      Septic Shock Male 
5  2865    68    45   1.9  1.3   14.1   7.6   6.9  NA    Yes      Alive     Severe Seps… Male 
6   663    70    45   2     7.20   4.3  3      1.8   2.8 No       Alive     Severe Seps… Male 
```

WIDE

One

patient

per row



```
> head(crp.Data, n=10L)
# A tibble: 10 × 9
      ID   age  SAPS antib_1h discharge SEPSIS        SEX   day     crp
   <int> <int> <int> <chr>    <chr>     <chr>         <chr> <chr> <dbl>
 1  1467    57    41 No       Alive     Severe Sepsis Male  1       1.7
 2  1467    57    41 No       Alive     Severe Sepsis Male  2      21  
 3  1467    57    41 No       Alive     Severe Sepsis Male  3      21  
 4  1467    57    41 No       Alive     Severe Sepsis Male  4      NA  
 5  1467    57    41 No       Alive     Severe Sepsis Male  5      NA  
 6  1467    57    41 No       Alive     Severe Sepsis Male  6       9.2
 7  4098    38    28 No       Alive     Severe Sepsis Male  1       1.8
 8  4098    38    28 No       Alive     Severe Sepsis Male  2      12.9
 9  4098    38    28 No       Alive     Severe Sepsis Male  3      14.2
```

LONG

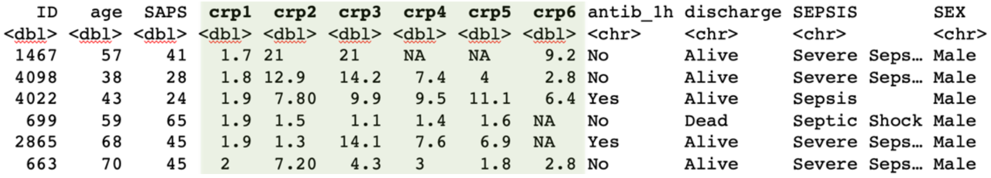
One

patient

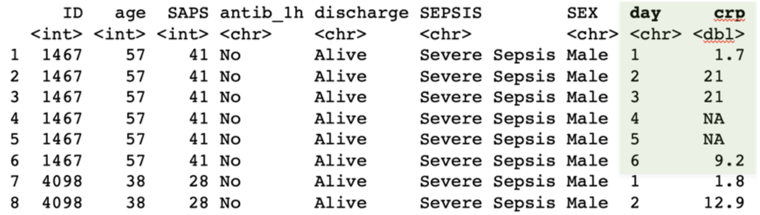
multiple rows

Wide vs Long format

```
library(tidyr)
crp.Data <- pivot_longer(data = crp.Data.wide,                          #dataset
                     cols = c(crp1, crp2, crp3, crp4, crp5, crp6),       #longitudinal measurements
                     #cols = crp1:crp6
                     names_to = "day",                                   #name of time variable
                      names_prefix = "crp”,                              #removes crp from time values
                     values_to = "crp",                                  #name of measurements
                      values_drop_na=TRUE)
```



```
   ID   age  SAPS  crp1  crp2  crp3  crp4  crp5  crp6 antib_1h discharge SEPSIS       SEX
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>    <chr>     <chr>        <chr>
 1467    57    41   1.7 21     21    NA    NA      9.2 No       Alive     Severe Seps… Male
 4098    38    28   1.8 12.9   14.2   7.4   4      2.8 No       Alive     Severe Seps… Male
 4022    43    24   1.9 7.80    9.9   9.5  11.1    6.4 Yes      Alive     Sepsis       Male
  699    59    65   1.9 1.5     1.1   1.4   1.6 NA     No       Dead      Septic Shock Male
 2865    68    45   1.9 1.3    14.1   7.6   6.9 NA     Yes      Alive     Severe Seps… Male
  663    70    45   2   7.20    4.3   3     1.8    2.8 No       Alive     Severe Seps… Male
```



```
     ID   age  SAPS antib_1h discharge SEPSIS        SEX   day     crp
  <int> <int> <int> <chr>    <chr>     <chr>         <chr> <chr> <dbl>
1  1467    57    41 No       Alive     Severe Sepsis Male  1       1.7
2  1467    57    41 No       Alive     Severe Sepsis Male  2      21
3  1467    57    41 No       Alive     Severe Sepsis Male  3      21
4  1467    57    41 No       Alive     Severe Sepsis Male  4      NA
5  1467    57    41 No       Alive     Severe Sepsis Male  5      NA
6  1467    57    41 No       Alive     Severe Sepsis Male  6       9.2
7  4098    38    28 No       Alive     Severe Sepsis Male  1       1.8
8  4098    38    28 No       Alive     Severe Sepsis Male  2      12.9
```

```
library(tidyr)
crp.Data.wide <- pivot_wider(crp.Data,                                   #dataset
                      names_from = "day",                                #time ”names”
                      values_from=crp )                                  #crp values
```

Wide vs Long format

- Note if the measurements were taken at different time points for each patient the long format is the obvious format
- One way we could do it is by indexing the measurement (1 st, 2nd, 3rd ,… measurement) and make it wide based on this index

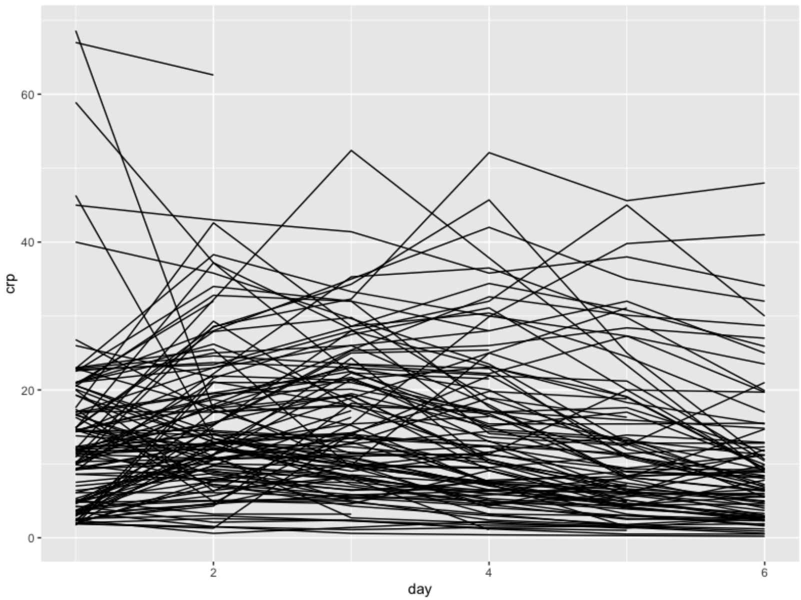
patid	time	GDS
1	0.00	1.44
1	2.19	1.5
1	2.72	0
2	0.00	0.07
2	0.48	0.23
2	2.81	0
3	0.00	0.07
3	0.59	1.92
4	0.00	0.08
4	1.03	0.31
4	1.97	0.69
4	2.49	0.69

patid	time	GDS	Meas_nr
1	0.00	1.44	1
1	2.19	1.5	2
1	2.72	0	3
2	0.00	0.07	1
2	0.48	0.23	2
2	2.81	0	3
3	0.00	0.07	1
3	0.59	1.92	2
4	0.00	0.08	1
4	1.03	0.31	2
4	1.97	0.69	3
4	2.49	0.69	4

Spaghetti plots

- Individual trajectories over time
- CRP over the 5 days in the ICU

```
#basic plot
ggplot(data = crp.Data,
        aes(x=day, y=crp, group=ID)) +
   geom_line()
```

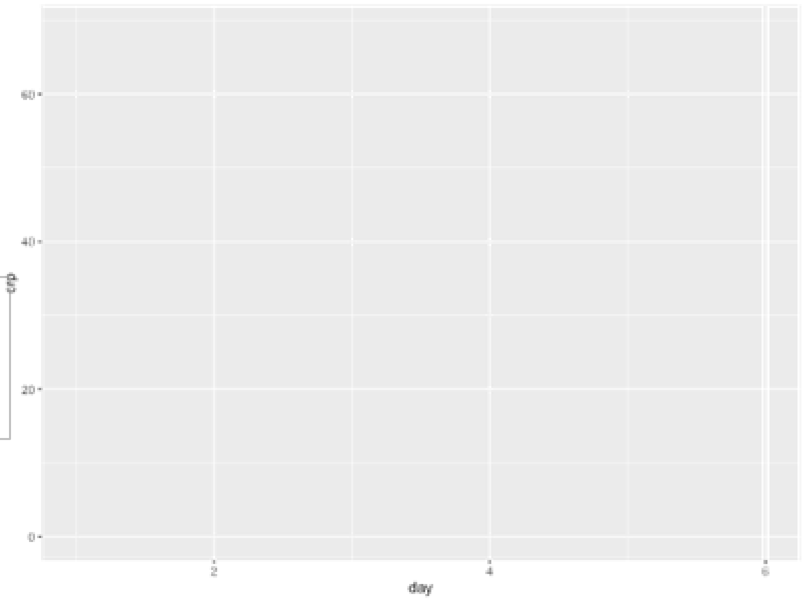


Spaghetti plots

- Individual trajectories over time
- CRP over the 5 days in the ICU

```
#basic plot
ggplot(data = crp.Data,
        aes(x=day, y=crp, group=ID))
```

Sets the plot "map"

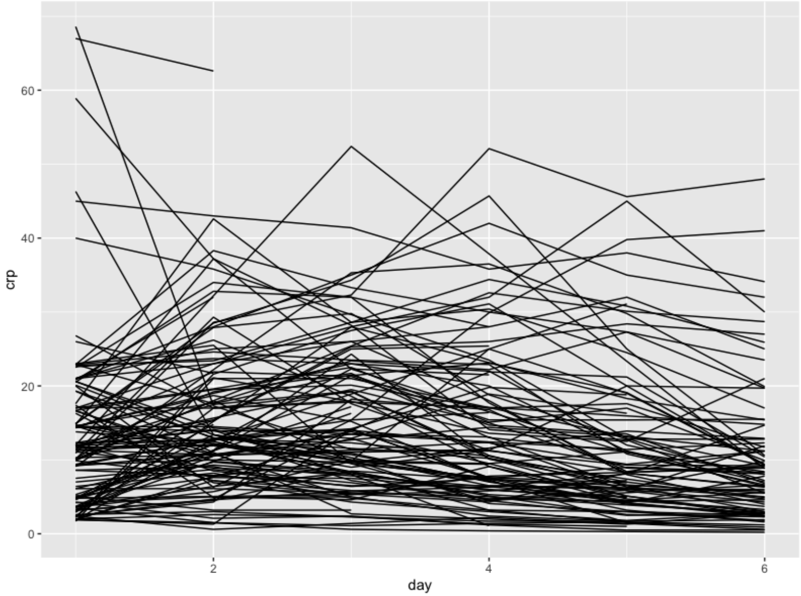


Spaghetti plots

- Individual trajectories over time
- CRP over the 5 days in the ICU

```
#basic plot
ggplot(data = crp.Data,
        aes(x=day, y=crp, group=ID)) +
  geom_line()
```

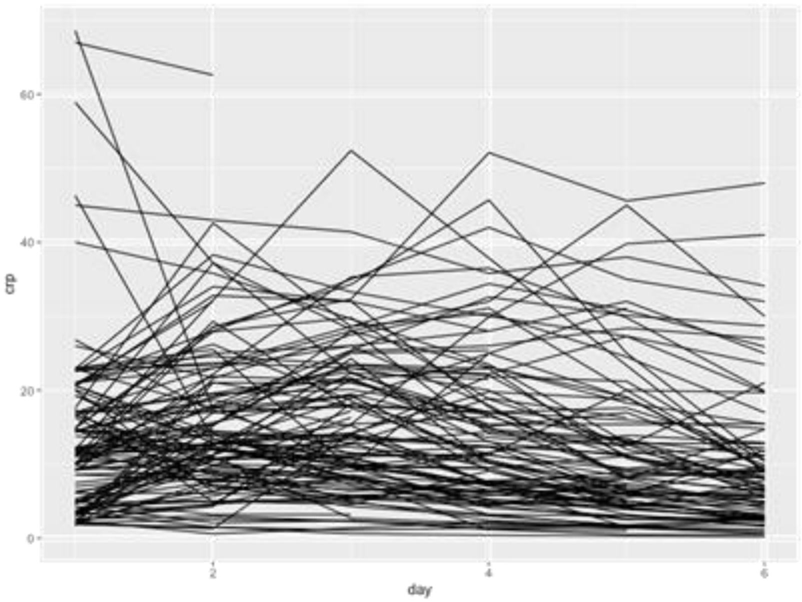
Sets the geometry
(in this case, lines)



Spaghetti plots

basic plot

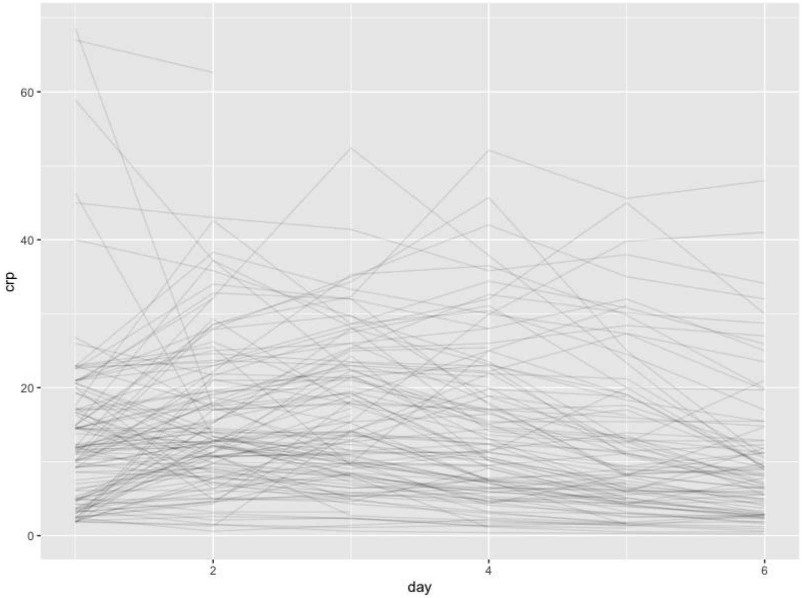
```
ggplot(data = crp.Data,
        aes(x=day, y=crp, group=ID)) +
   geom_line()
```



Spaghetti plots

add transparency

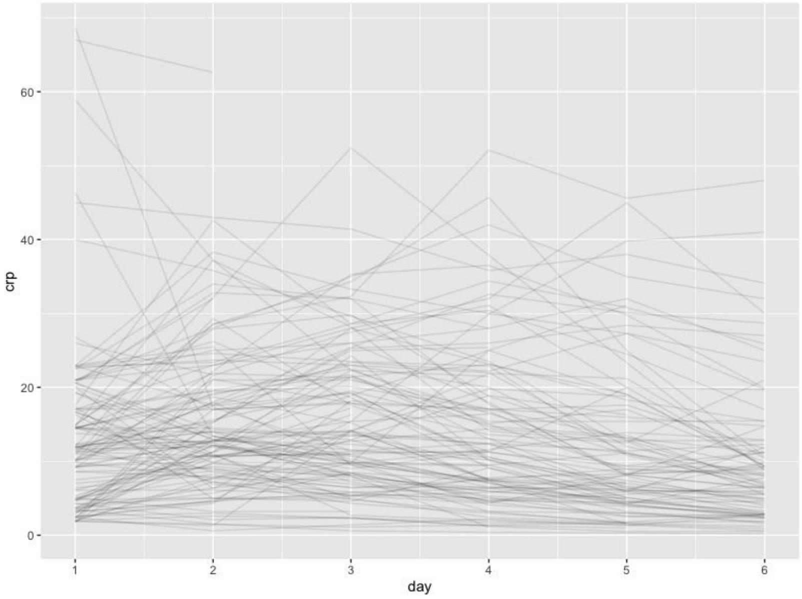
```
ggplot(data = crp.Data,
          aes(x=day, y=crp, group=ID)) +
   geom_line(alpha=.1)
```



Spaghetti plots

change the breaks in the x-axis

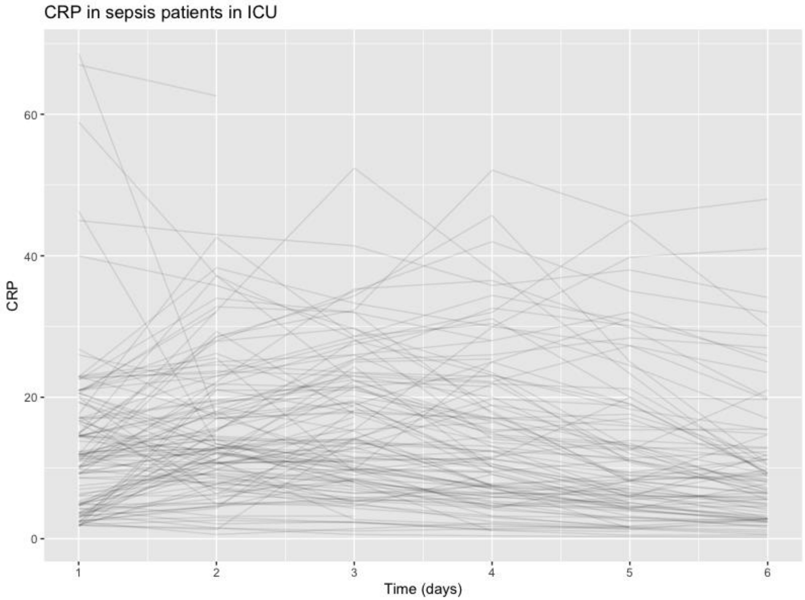
```
ggplot(data = crp.Data,
           aes(x=day, y=crp, group=ID)) +
    geom_line(alpha=.1) +
    scale_x_continuous(breaks=c(1:6))
```



Spaghetti plots

change the labels and title

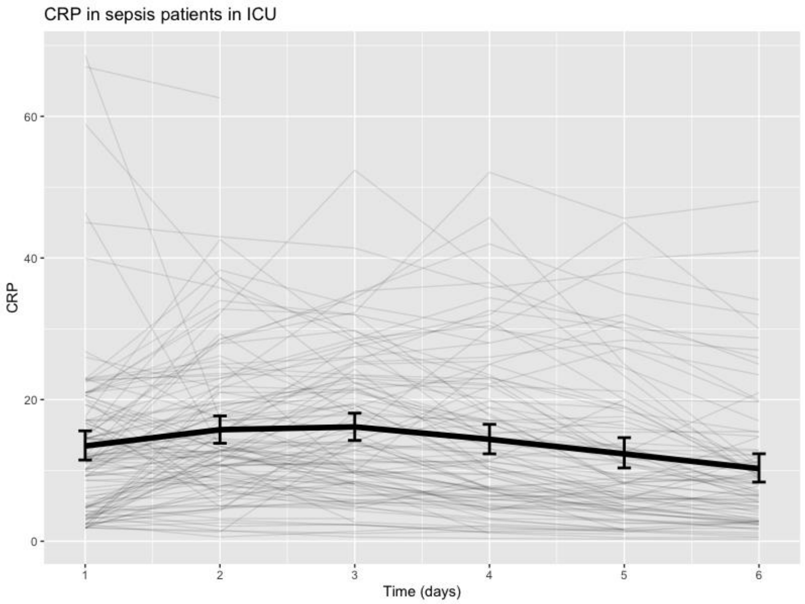
```
ggplot(data = crp.Data,
        aes(x=day, y=crp, group=ID)) +
   geom_line(alpha=.1) +
   scale_x_continuous(breaks=c(1:6))
   labs(title="CRP in sepsis patients in ICU",
         x = "Time (days)",
         y = "CRP")
```



Spaghetti plots – mean profile plots

add the mean profile plot and 95%CI

```
ggplot(data = crp.Data,
         aes(x=day, y=crp, group=ID)) +
geom_line(alpha=.1) +
scale_x_continuous(breaks=c(1:6)) +
labs(title="CRP in sepsis patients in ICU",
      x = "Time (days)",
      y = "CRP") +
stat_summary(fun=mean, na.rm=T,
             aes(group="none"),
             geom="line", lwd=2,
             show.legend = F) +
stat_summary(fun.data=mean_cl_boot,
             aes(group="none"),
             geom="errorbar",
             lwd=1, width=.1,
             show.legend = F)
```



Adds the 95% Conf Interval

The overall setup is with group by ID. If we don't change the grouping, the smoothing line will be done for each patient

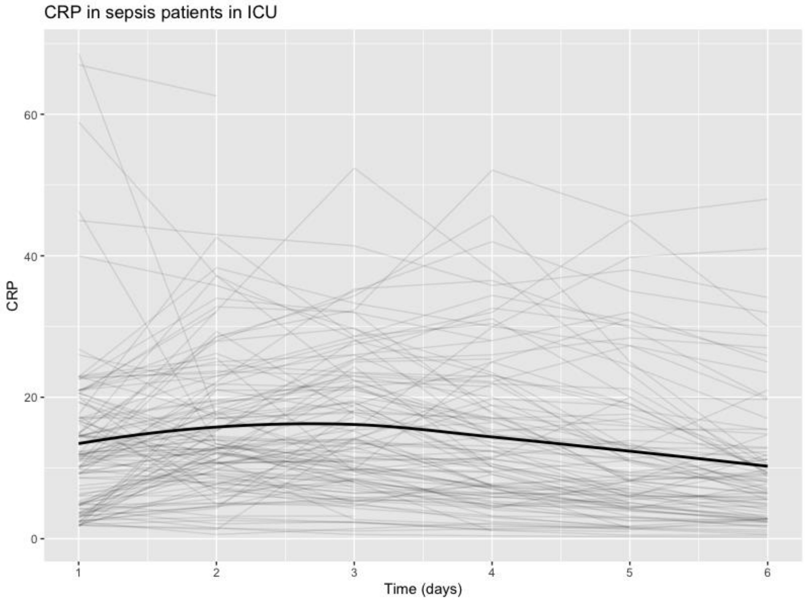
Spaghetti plots

add a trend (smooth) line

```
ggplot(data = crp.Data,
           aes(x=day, y=crp, group=ID)) +
    geom_line(alpha=.1) +
    scale_x_continuous(breaks=c(1:6))
    labs(title="CRP in sepsis patients in ICU",
           x = "Time (days)",
           y = "CRP") +
    geom_smooth(method = loess
                aes(group="none"),
                se=F,
                color="black")
```

No confidence interval around the line

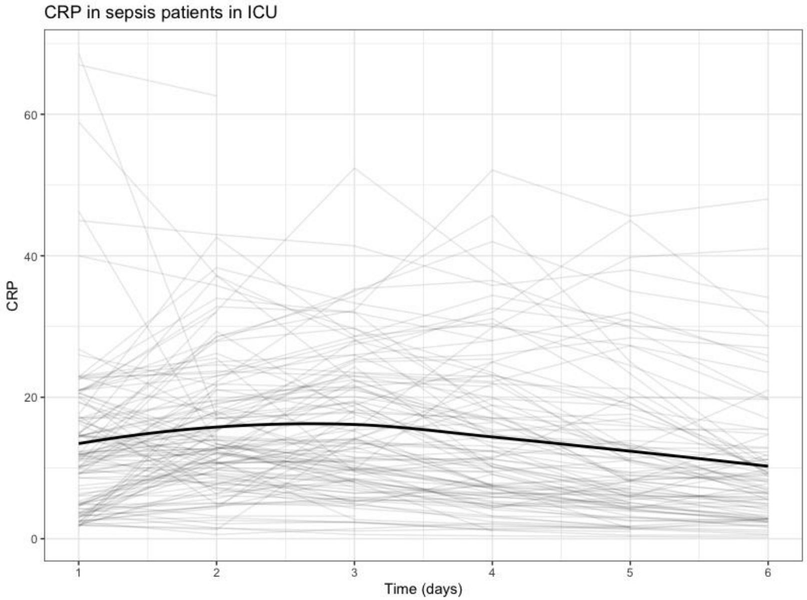
The overall setup is with group by ID. If we don't change the grouping, the smoothing line will be done for each patient



Spaghetti plots

overall look with theme (in this case black and white)

```
ggplot(data = crp.Data,
        aes(x=day, y=crp, group=ID)) +
   geom_line(alpha=.1) +
   scale_x_continuous(breaks=c(1:6))
   labs(title="CRP in sepsis patients in ICU",
         x = "Time (days)",
         y = "CRP") +
   geom_smooth(method = loess
              aes(group="none"),
              se=F,
              color="black") +
   theme_bw()
```

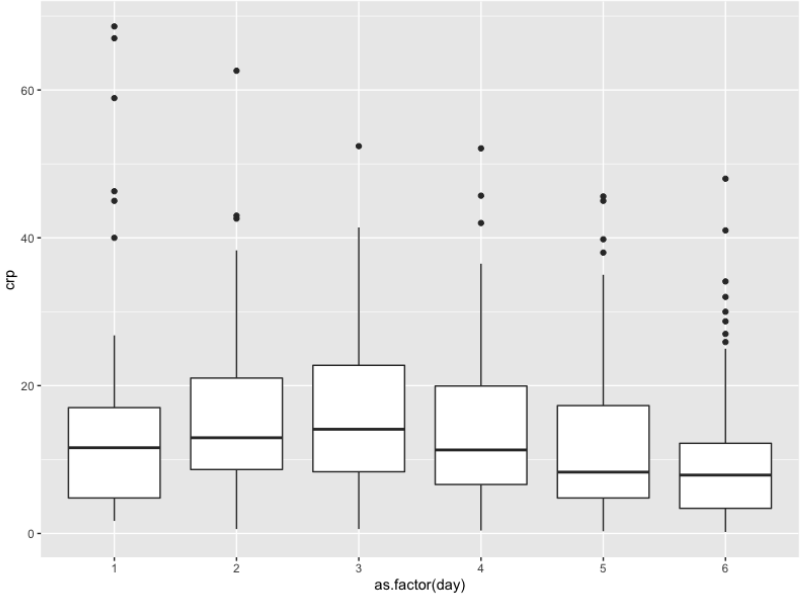


Boxplot

- Observation times are the same for all the patients
- We can plot the distribution for each time point

```
ggplot(data = crp.Data,
       aes(x=as.factor(day), y=crp)) +
    geom_boxplot()
```

Need to make day categorical

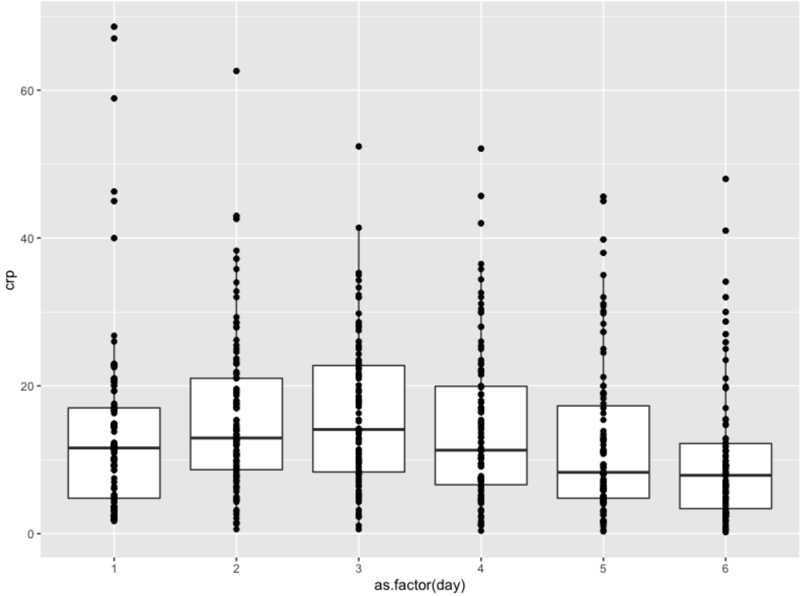


Boxplot

- Observation times are the same for all the patients
- We can plot the distribution for each time point

```
ggplot(data = crp.Data,
       aes(x=as.factor(day), y=crp)) +
    geom_boxplot() +
    geom_point()
```

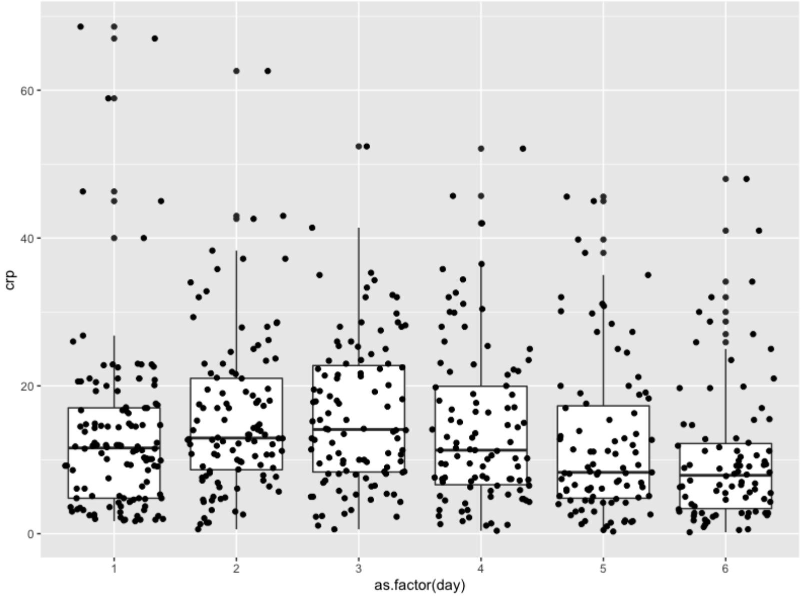
Adds the observations



Boxplot

- Observation times are the same for all the patients
- We can plot the distribution for each time point

```
ggplot(data = crp.Data,
       aes(x=as.factor(day), y=crp)) +
    geom_boxplot() +
    geom_jitter()
```



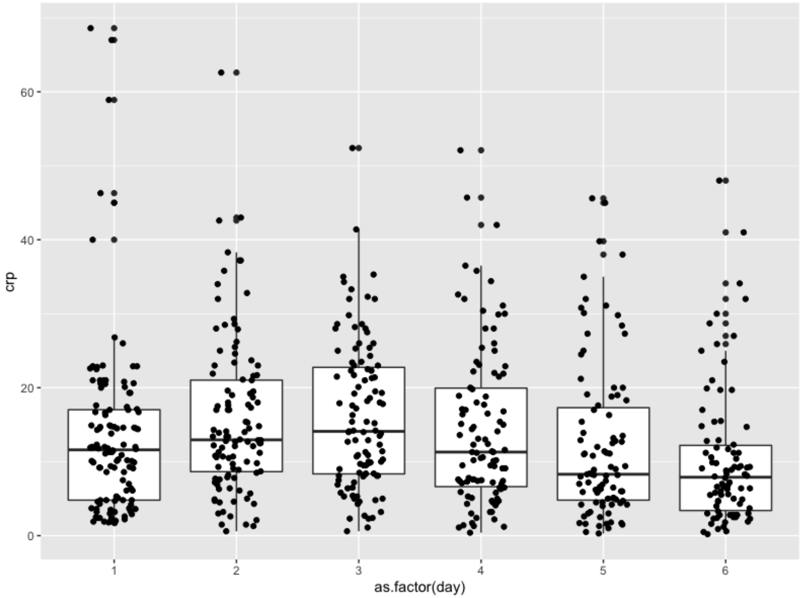
Similar to geom_point() but the points are some "jittered"

Boxplot

- Observation times are the same for all the patients
- We can plot the distribution for each time point

```
ggplot(data = crp.Data,
       aes(x=as.factor(day), y=crp)) +
    geom_boxplot() +
    geom_jitter(width = .2)
```

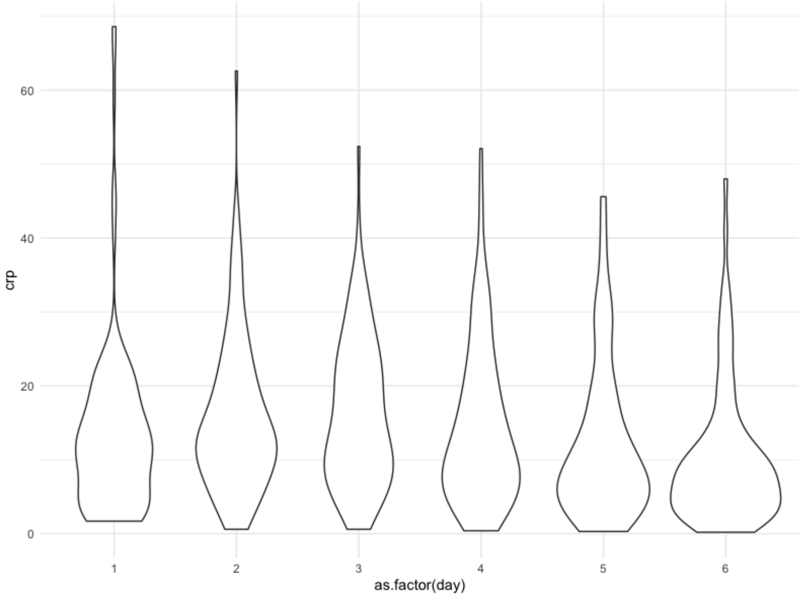
Amount of “jittering”



Violin plots

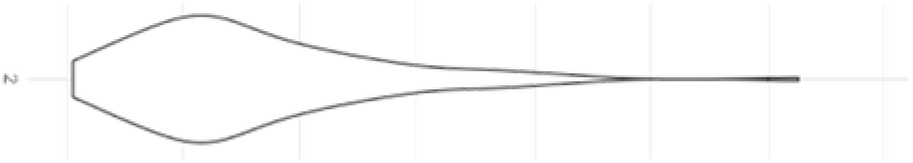
- An alternative to the boxplot is the violin plot

```
ggplot(data = crp.Data,
       aes(x=as.factor(day), y=crp)) +
    geom_violin() +
    theme_minimal()
```

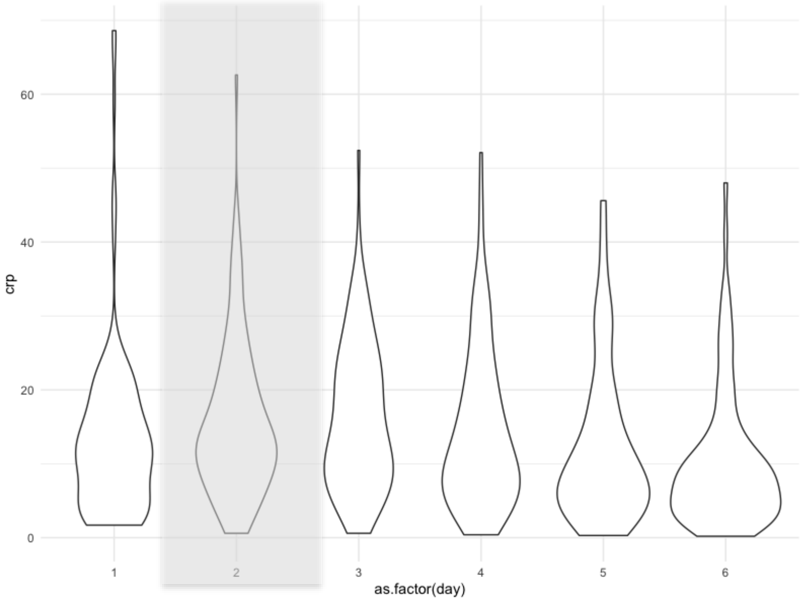


Violin plots

- An alternative to the boxplot is the violin plot

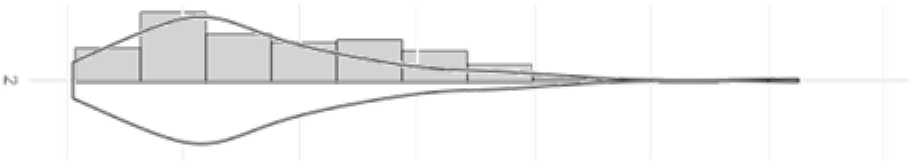


```
ggplot(data = crp.Data,
       aes(x=as.factor(day), y=crp)) +
    geom_violin() +
    theme_minimal()
```

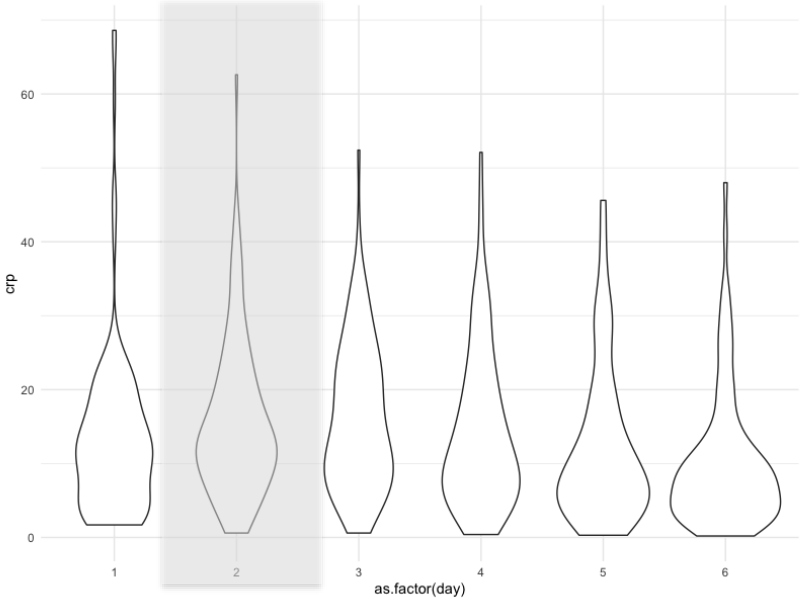


Violin plots

- An alternative to the boxplot is the violin plot



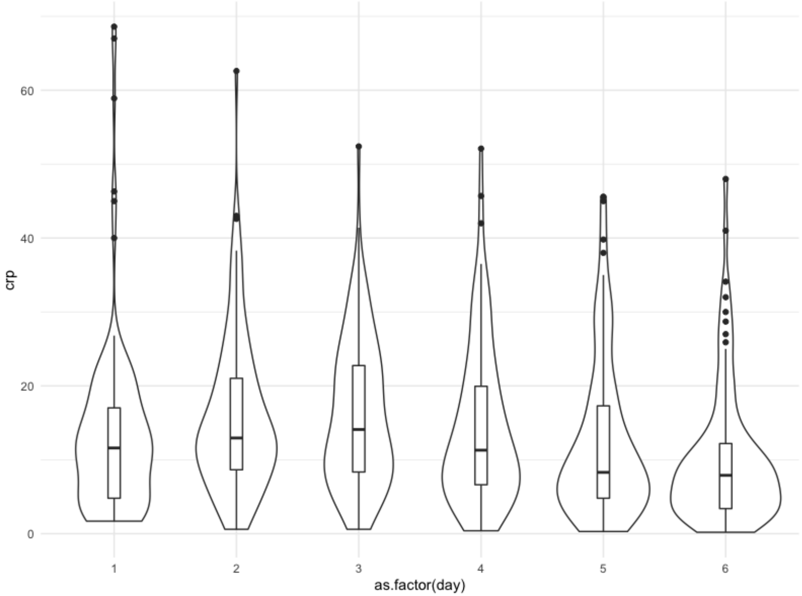
```
ggplot(data = crp.Data,
       aes(x=as.factor(day), y=crp)) +
    geom_violin() +
    theme_minimal()
```



Violin plots

- An alternative to the boxplot is the violin plot
- It is common to add a boxplot to the ”violins"

```
ggplot(data = crp.Data,
       aes(x=as.factor(day), y=crp)) +
    geom_violin() +
    geom_boxplot(width = 0.1) +
    theme_minimal()
```



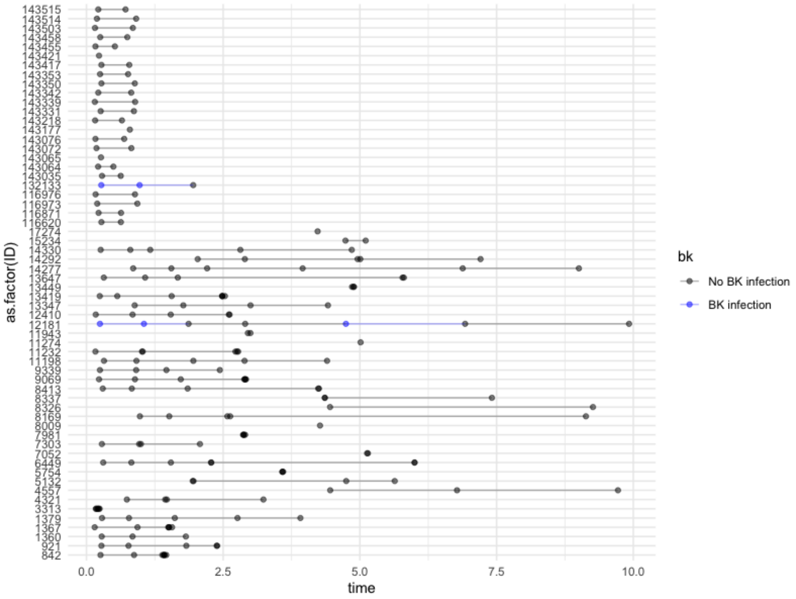
Categorical data

- There is no standard way of plotting these data
- Depending on the type of data, there might be some options that work well
- Let’s see an example of BK infection status over time in patients that had a kidney transplant

ID	BK	Time
1	No BK infection	3.5
1	No BK infection	7.1
1	No BK infection	9.2
1	BK infection	11.0
2	BK infection	2.7
2 ⋮	No BK infection ⋮	6.4 ⋮

Categorical data

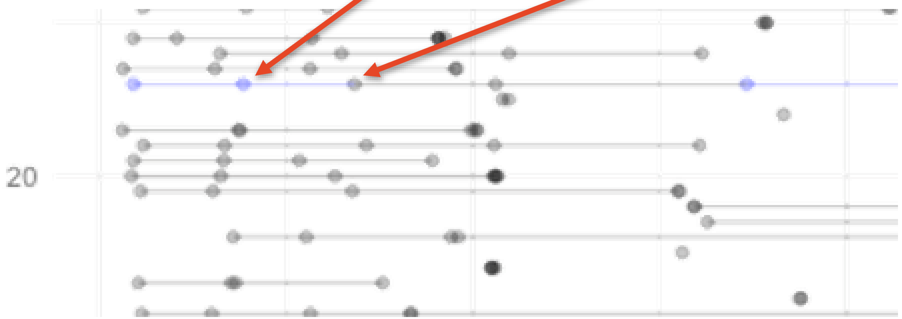
- If we have continuous time
- We can plot (some) patients' values across the time and have a different colour for events



```
ggplot(data = bk.Data[1:200,],
       aes(x=time,
           y=as.factor(ID),
           colour=bk)) +
  scale_colour_manual(values = c("No BK infection" = "black",
                                 "BK infection" = "blue"))    +
  geom_point(alpha=.5) +
  geom_line(aes(group=ID), alpha=.3) +
  theme_minimal()
```

Categorical data

- If we have continuous time
- We can plot (some) patients' values across the time and have a different colour for events



Infection

No Infection

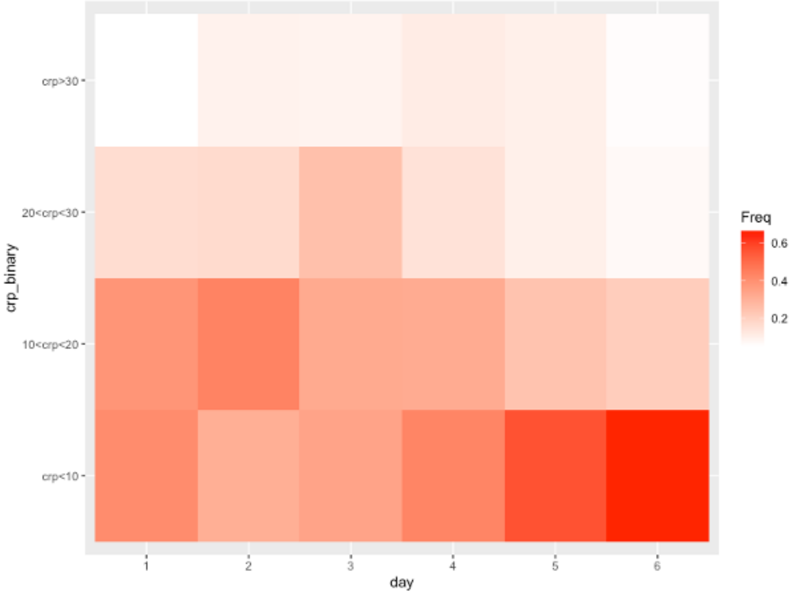
```
ggplot(data = bk.Data[bk.Data$ID<80,],
       aes(x=days_after_transplant,
           y=ID,
           colour=bk)) +
  scale_color_manual(values = c("No BK infection" = "black",
                                "BK infection" = "blue"))   +
  geom_point(alpha=.3) + geom_line(aes(group=ID), alpha=.3) +
  theme_minimal()
```

Categorical data

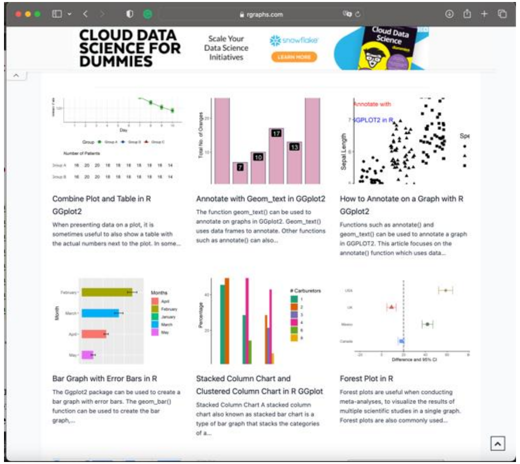
- Or use a heatmap to represent the distribution of the categories by time

```
new.CKR <- crp.Data %>%
  mutate(crp_binary = ifelse(crp<10, "crp<10",
                             ifelse(crp<20, "10<crp<20",
                                    ifelse(crp<30, "20<crp<30",
                                           "crp>30")))) %>%
  mutate(crp_binary=factor(crp_binary,
                            levels=c("crp<10","10<crp<20",
                                     "20<crp<30", "crp>30"))) %>%
  select(c("crp_binary", "day")) %>%
  table() %>%
  prop.table(., margin=2) %>% as.data.frame()

ggplot(new.CKR, aes(x=day, y=crp_binary, fill=Freq))+
  scale_fill_gradient(low = "white", high = "red") +
  geom_tile()
```



Resources



Combine Plot and Table in R GGplot2

When presenting data on a plot, it is sometimes useful to also show a table with the actual numbers next to the plot. In some...

Annotate with Geom_text in GGplot2

The function geom_text() can be used to annotate on graphs in GGplot2. Geom_text() uses data frames to annotate. Other functions such as annotate() can also...

How to Annotate on a Graph with R GGplot2

Functions such as annotate() and geom_text() can be used to annotate a graph in GGPLOT2. This article focuses on the annotate() function which uses data...

Bar Graph with Error Bars in R

The Ggplot2 package can be used to create a bar graph with error bars. The geom_bar() function can be used to create the bar graph,...

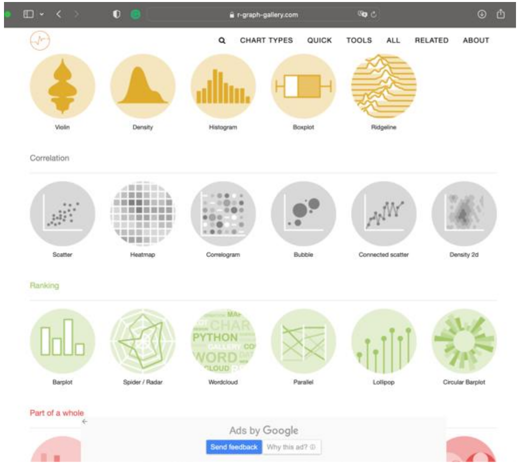
Stacked Column Chart and Clustered Column Chart in R GGplot

Stacked Column Chart A stacked column chart also known as stacked bar chart is a type of bar graph that stacks the categories of a...

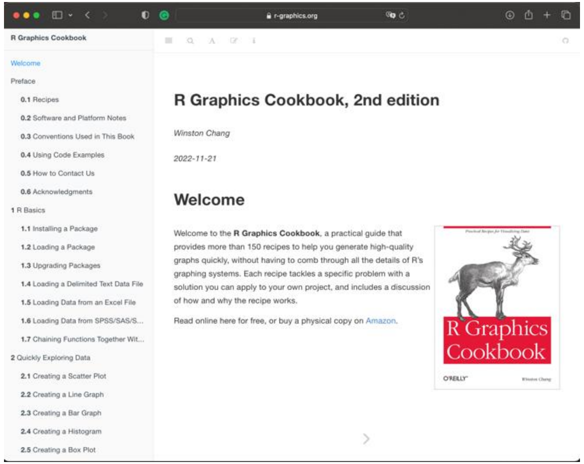
Forest Plot in R

Forest plots are useful when conducting meta-analyses, to visualize the results of multiple scientific studies in a single graph. Forest plots are also commonly used...

https://rgraphs.com



https://r-graph-gallery.com



R Graphics Cookbook

R Graphics Cookbook, 2nd edition

Winston Chang

2022-11-21

Welcome

Welcome to the **R Graphics Cookbook**, a practical guide that provides more than 150 recipes to help you generate high-quality graphs quickly, without having to comb through all the details of R's graphing systems. Each recipe tackles a specific problem with a solution you can apply to your own project, and includes a discussion of how and why the recipe works.

Read online here for free, or buy a physical copy on Amazon.

https://r-graphics.org