GGPLOT, basis

Samuel Murail

15/11/2018

ggplot

R visualization of data library developed by Hadley Wickham (RStudio).
https://ggplot2.tidyverse.org/index.html

- Advantages:
  - Syntax and grammar
  - Simple
  - Flexible
  - Big community
- Disadvantages:
  - No 3D representation
  - No network graph

Prepare data

Individual

```
N = 200

mon_df = data.frame( height=c(rnorm(N, mean=163, sd=11),
                                rnorm(N, mean=175, sd=12)),
                     weight=c(rnorm(N, mean=62, sd=10),
                                rnorm(N, mean=77, sd=10)),
                     gender=rep(c("W","M"), each=N),
                     group=rep(c("A","B"), each=N/2))
```

height	weight	gender	group
158.8	53.9	W	A
164.0	67.7	W	A
172.4	70.7	W	A
163.3	66.2	W	A
175.2	70.9	W	A

Data

Time Series

```
mon_df_2 = data.frame(time=1:N,
              A=cumsum(rnorm(N)),
              B=cumsum(rnorm(N)),
              C=cumsum(rnorm(N)))
```

```
library(reshape2)
mon_df_3 = melt(mon_df_2 ,
            id.vars='time',
            variable.name='experience')
```

time	A	B	C
1	0.06	0.46	-1.39
2	-1.42	2.16	-0.58
3	-0.86	3.20	-0.70
4	-2.30	4.02	-1.64
5	-3.44	3.18	-0.24
6	-2.15	4.15	-0.55
7	-0.79	3.00	1.23

time	experience	value
1	A	0.06
2	A	-1.42
3	A	-0.86

...

time	experience	value
197	C	-11.26
198	C	-13.62
199	C	-12.10
200	C	-10.22

ggplot

Installation:

Like most R packages, the installation is very simple:

```
install.packages("ggplot2")
```

Create a ggplot object, and define the data to use `data =` and the fields to use `aes ()`:

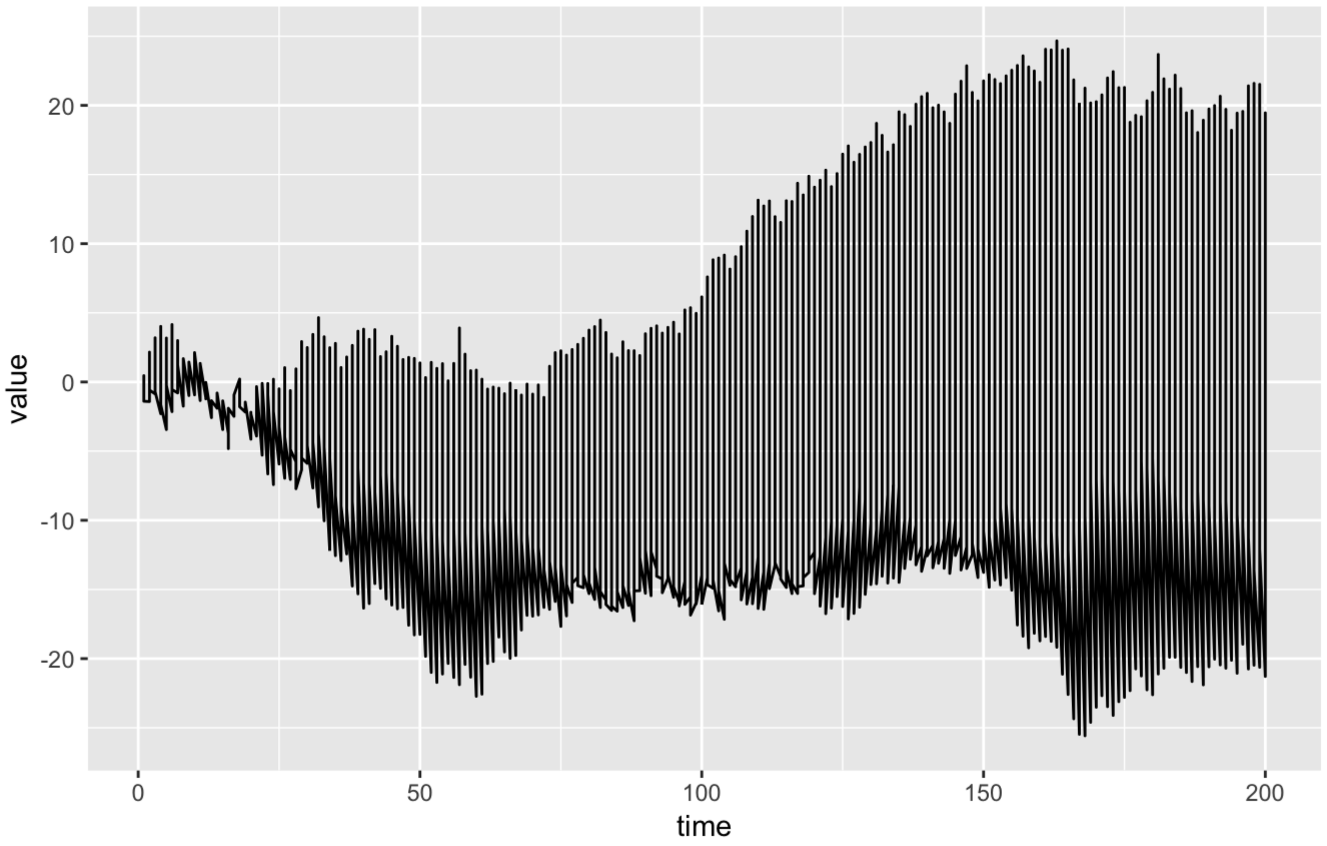
```
library(ggplot2)
graph = ggplot( data=mon_df_3, aes(x=time, y=value))
```

Add functions to this chart like

- `geom_line ()` to add a curve
- change the axis of the abscissa with `scale_y_continuous`
- etc …

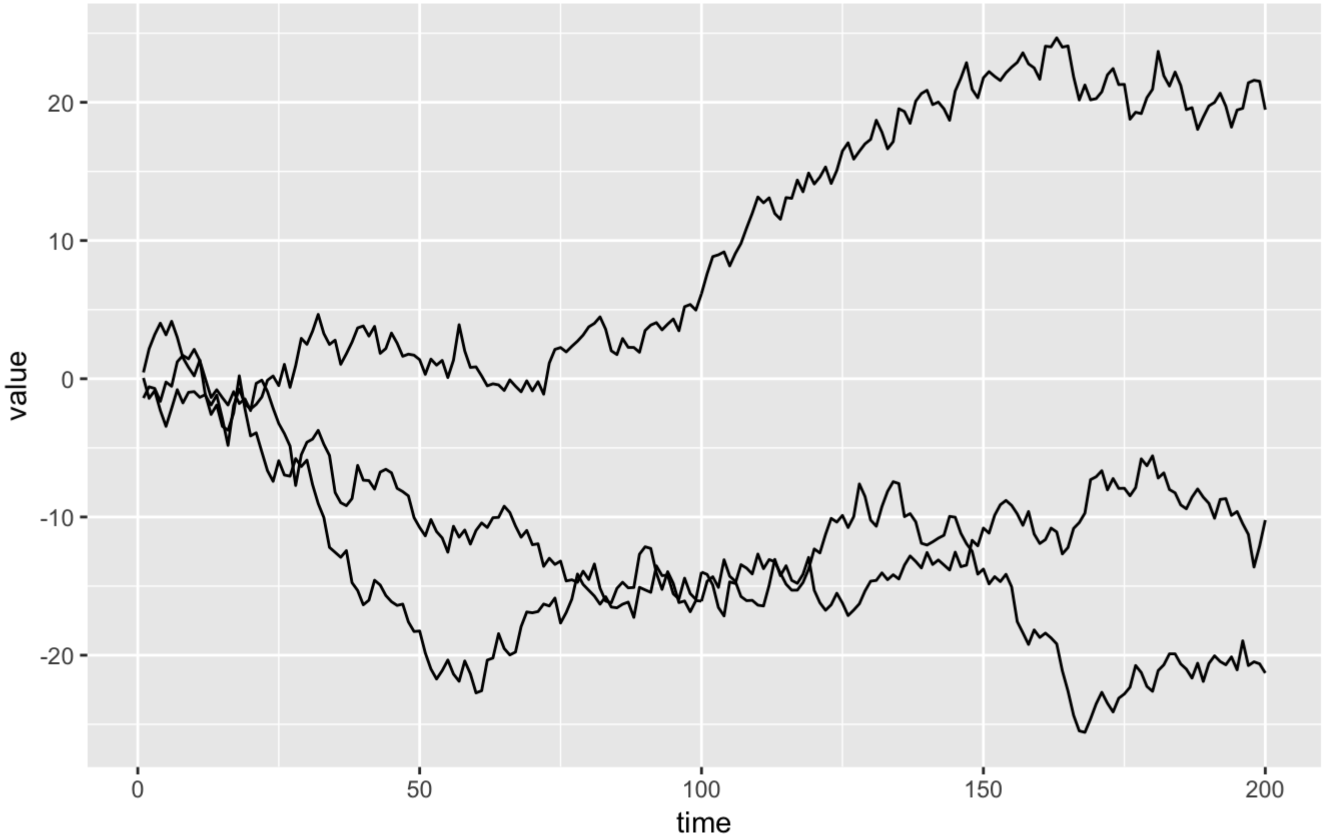
Basis

```
graph + geom_line()
```



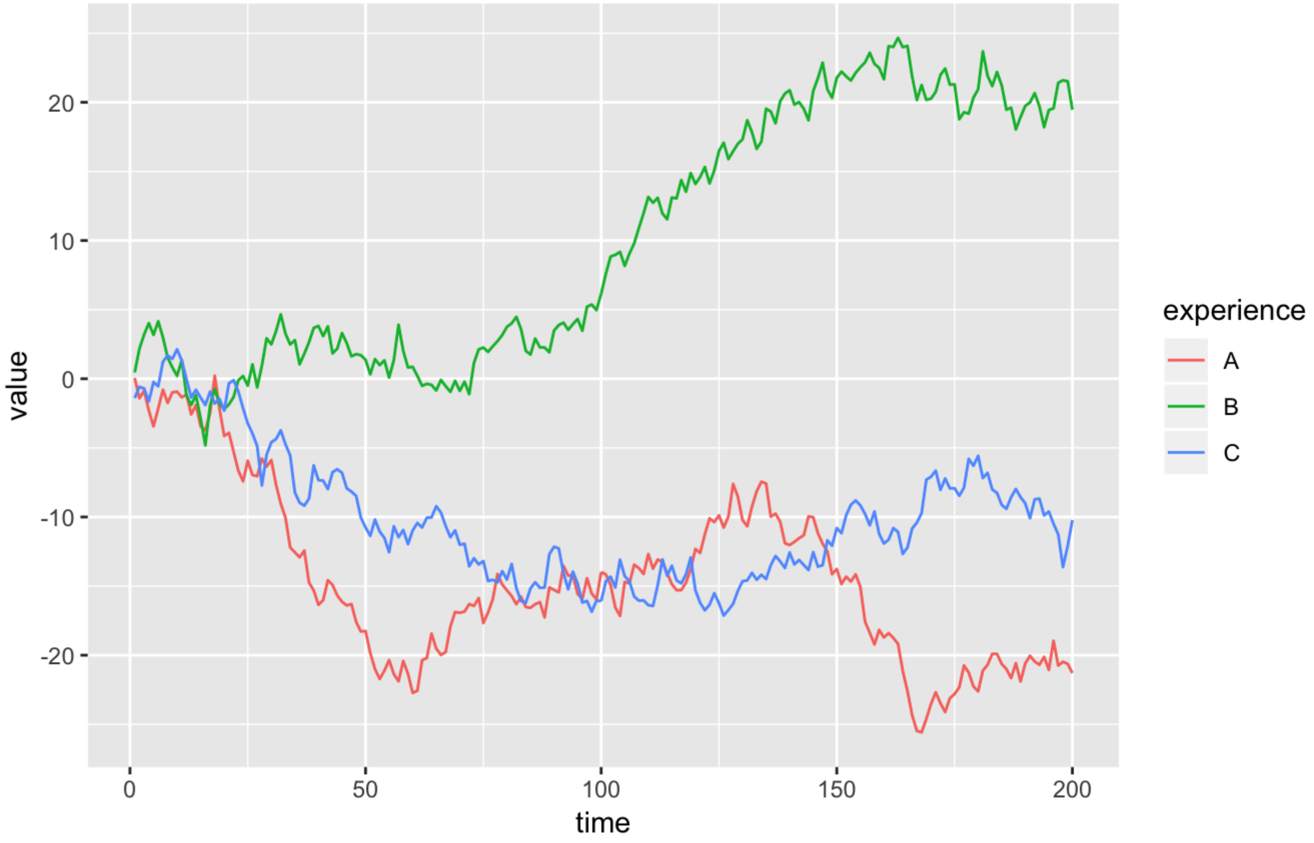
Basis (2)

```
ggplot(mon_df_3, aes(x=time, y=value, group=experience))+
  geom_line()
```



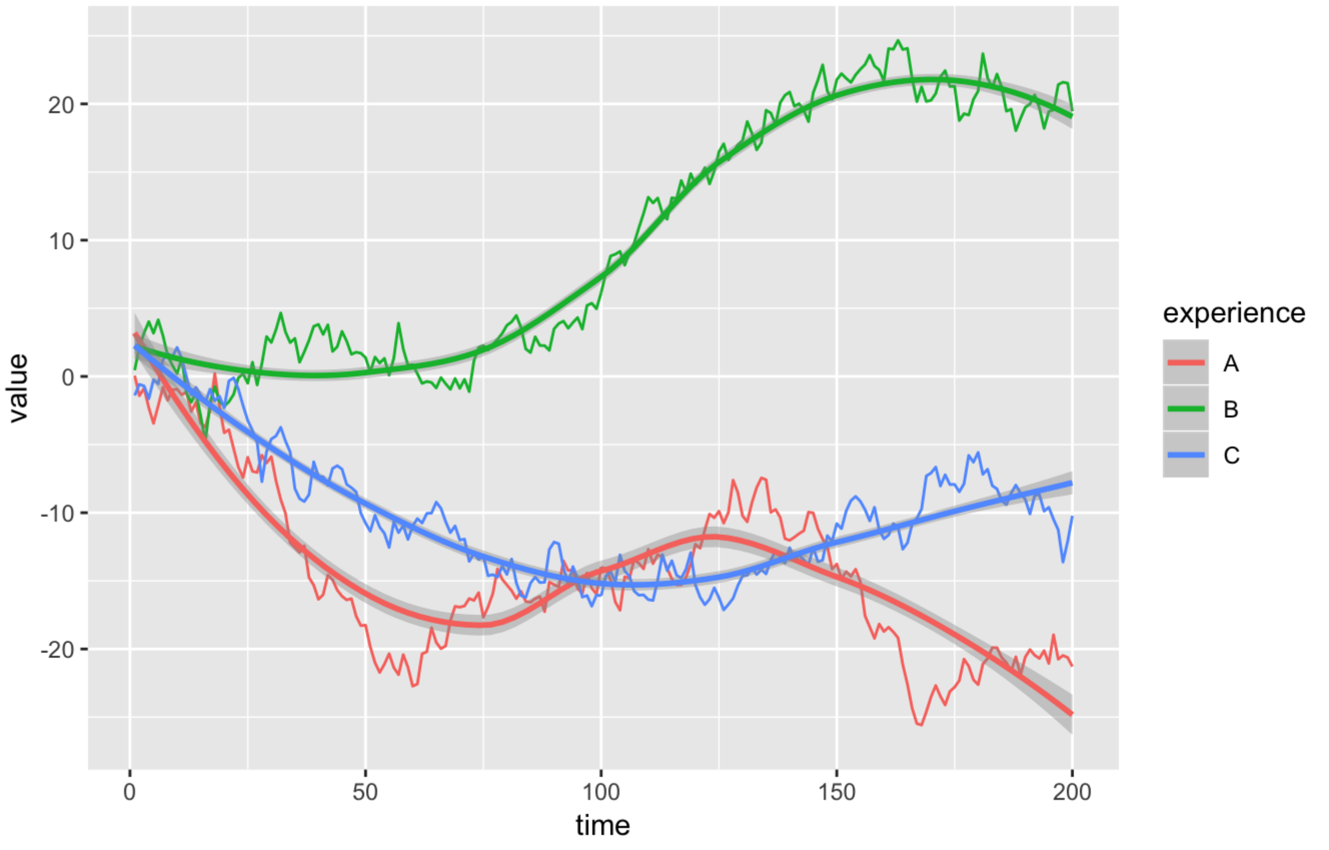
Basis (3)

```
ggplot(mon_df_3, aes(x=time, y=value, color=experience))+
  geom_line()
```



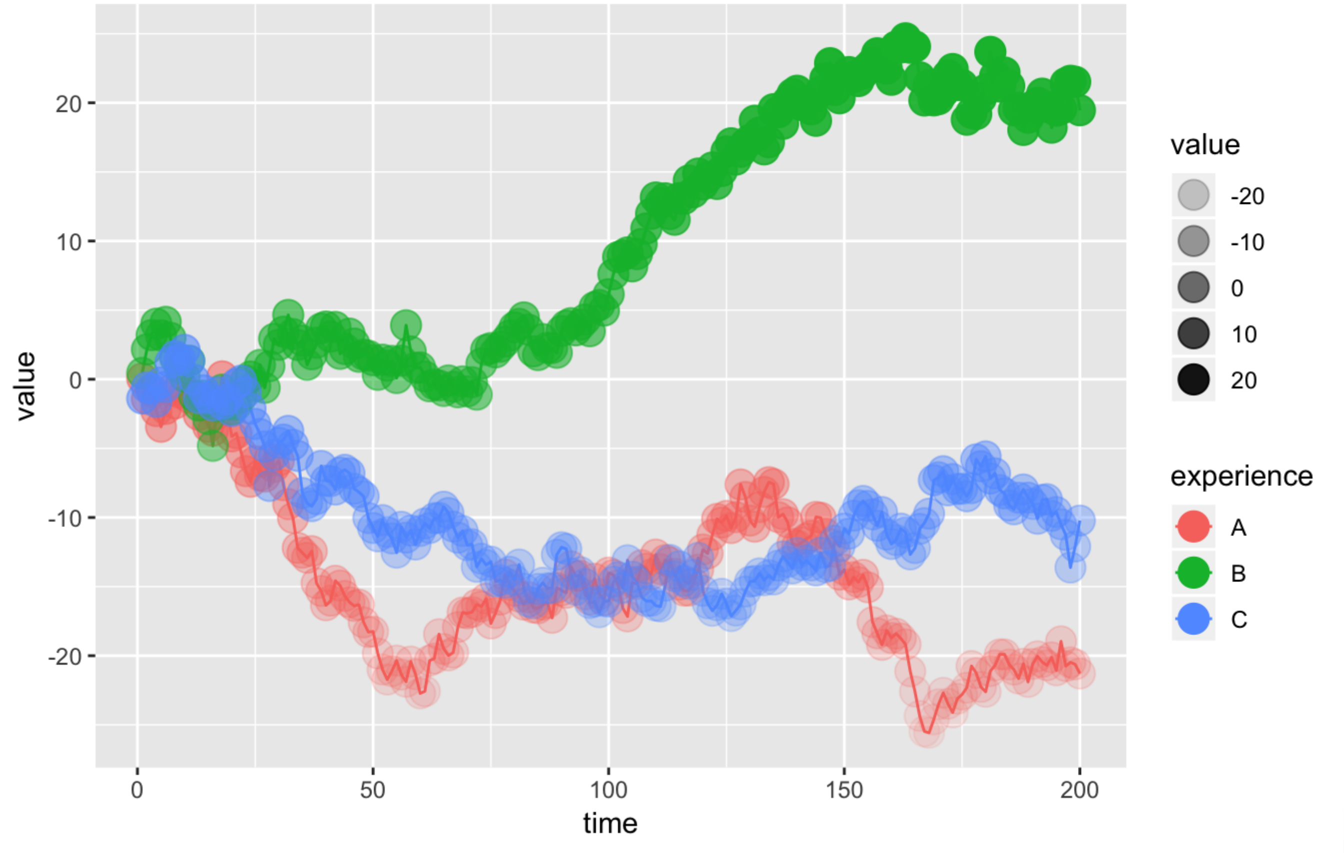
Local regression

```
ggplot(mon_df_3, aes(x=time, y=value, color=experience))+
  geom_line()+
  geom_smooth(method='loess')
```



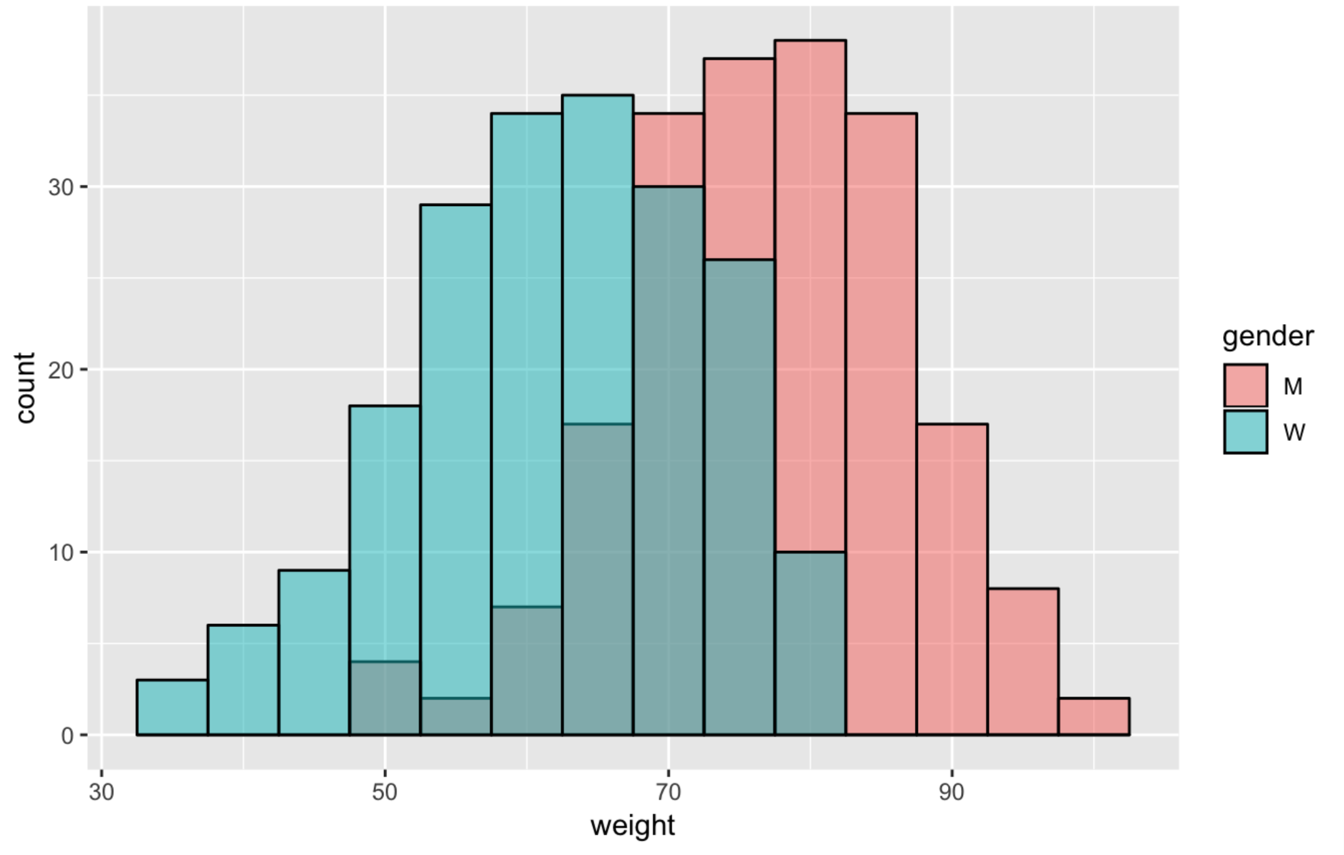
Aesthetic

```
ggplot(mon_df_3, aes(x=time, y=value, color=experience))+
  geom_line()+
  geom_point( aes(alpha=value), size=5)
```



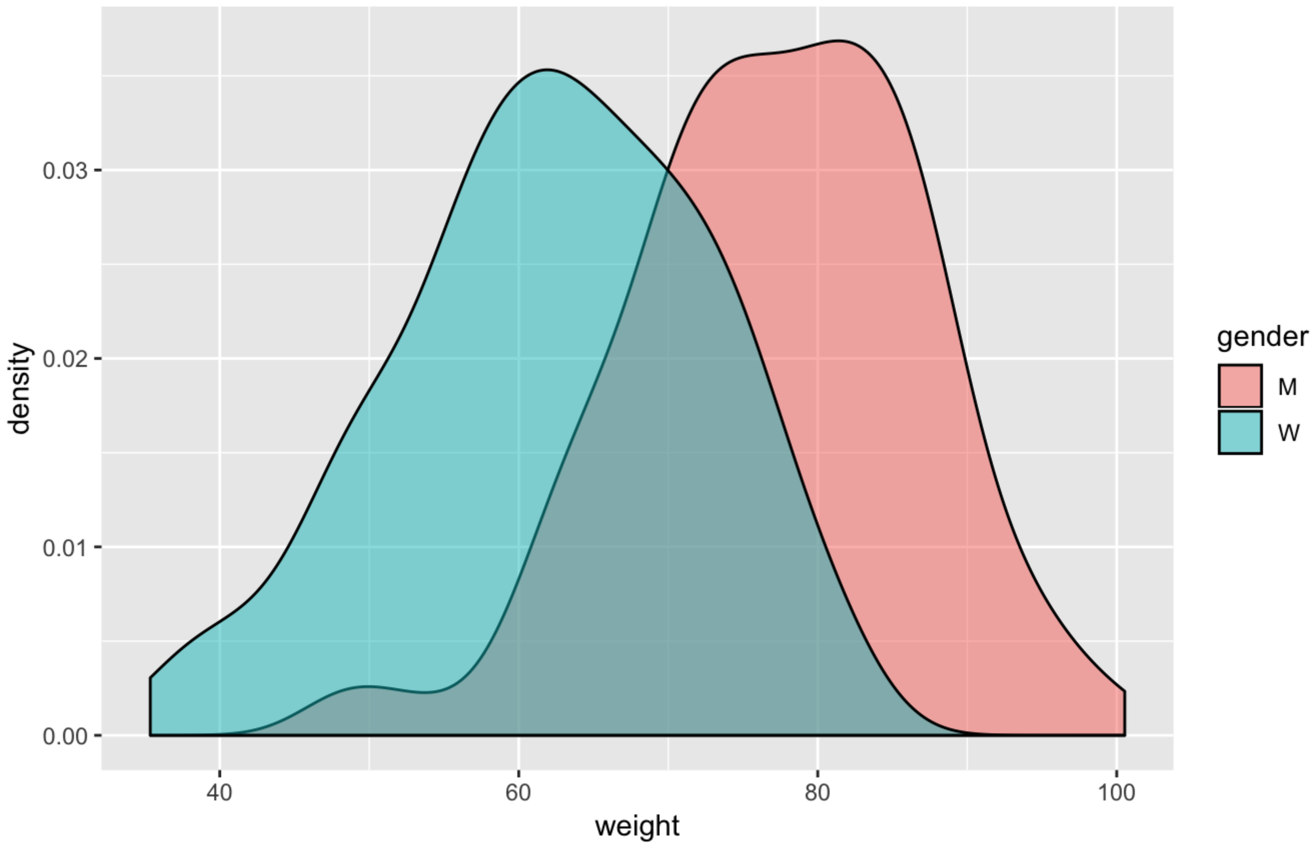
Distribution

```
ggplot(mon_df, aes(x=weight, fill=gender))+
  geom_histogram(binwidth=5, position='identity', alpha=0.5,
                 color="black")
```



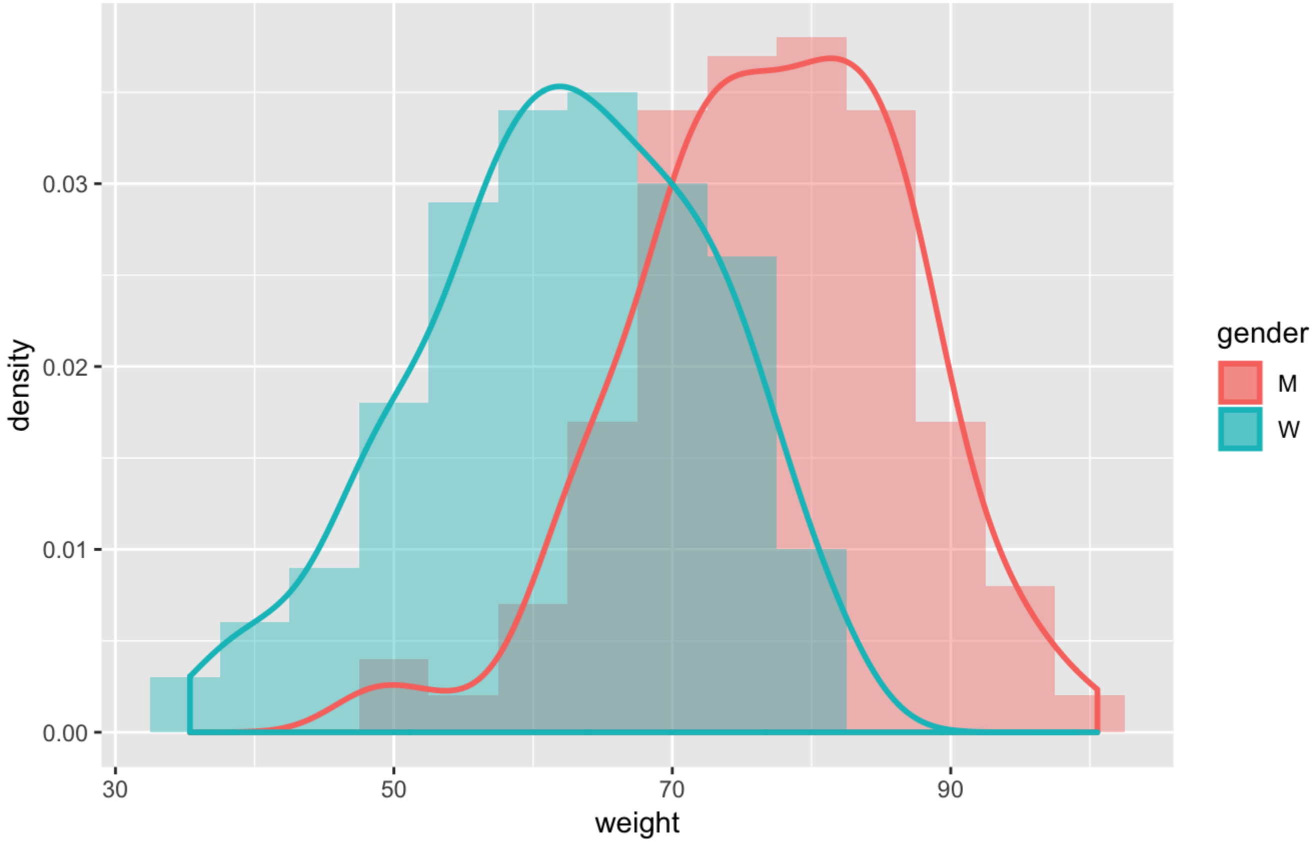
Distribution (2)

```
ggplot(mon_df, aes(x=weight, fill=gender))+
  geom_density(alpha=0.5)
```



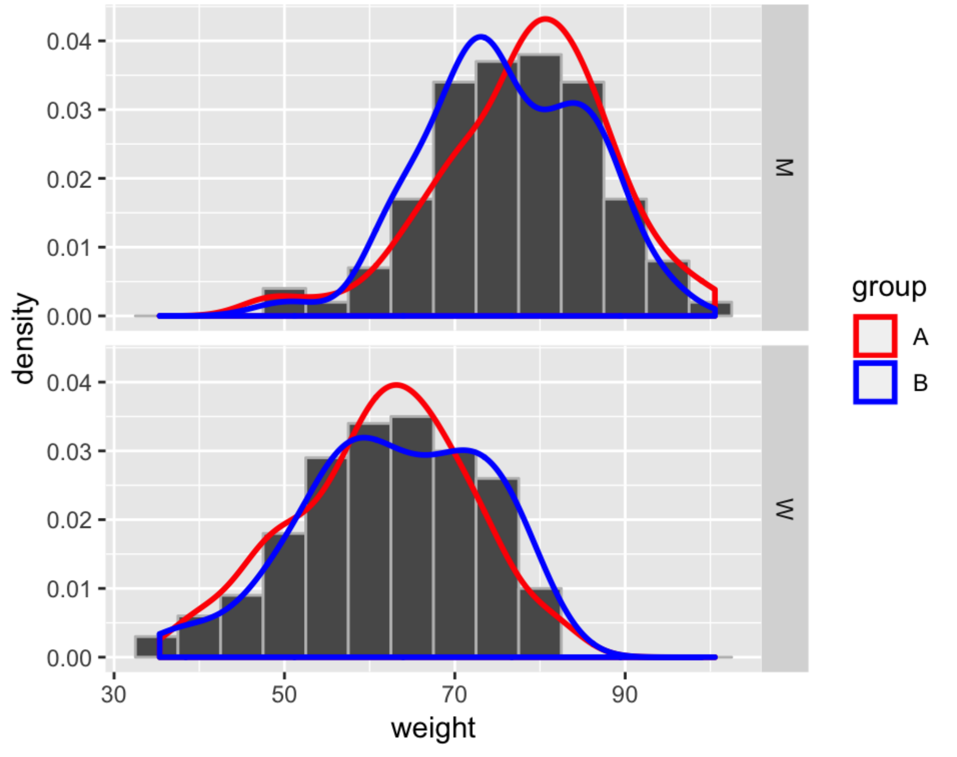
Distribution (3)

```
ggplot(mon_df, aes(x=weight))+
  geom_histogram(aes(y=..density.., fill=gender), position='identity',
                 alpha=0.4, binwidth=5)+
  geom_density(aes(color=gender), alpha=0.5, size=1)
```



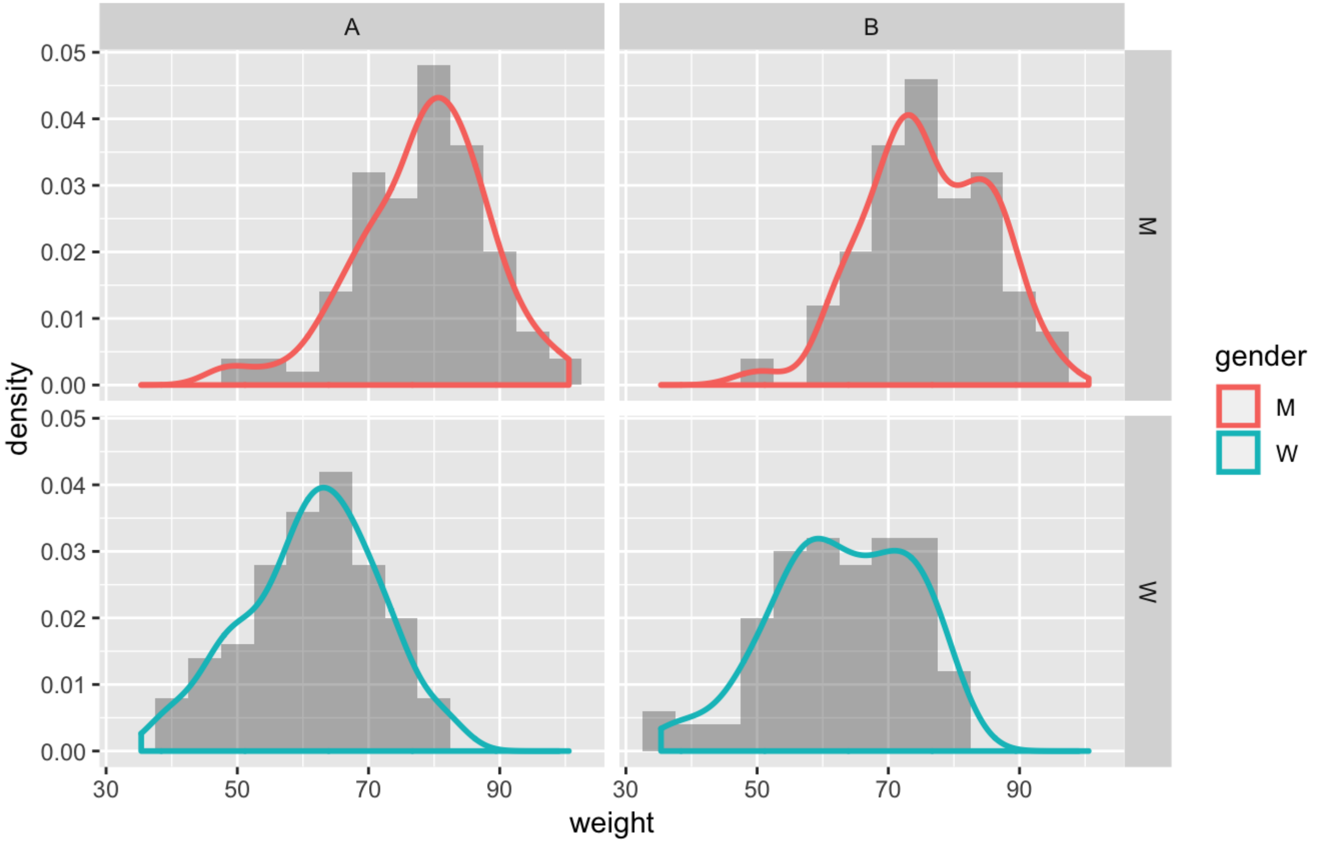
Panels

```
ggplot(mon_df, aes(x=weight))+
  geom_histogram(aes(y=..density..), position='identity',
                 binwidth=5, color="gray")+
  geom_density(aes(color=group), size=1)+
  scale_color_manual(values=c("red", "blue"))+
  facet_grid(gender~.)
```



Panels (2)

```
ggplot(mon_df, aes(x=weight))+
  geom_histogram(aes(y=..density..), position='identity', alpha=0.4, binwidth=5)+
  geom_density(aes(color=gender), alpha=0.5, size=1)+
  facet_grid(gender~group)
```



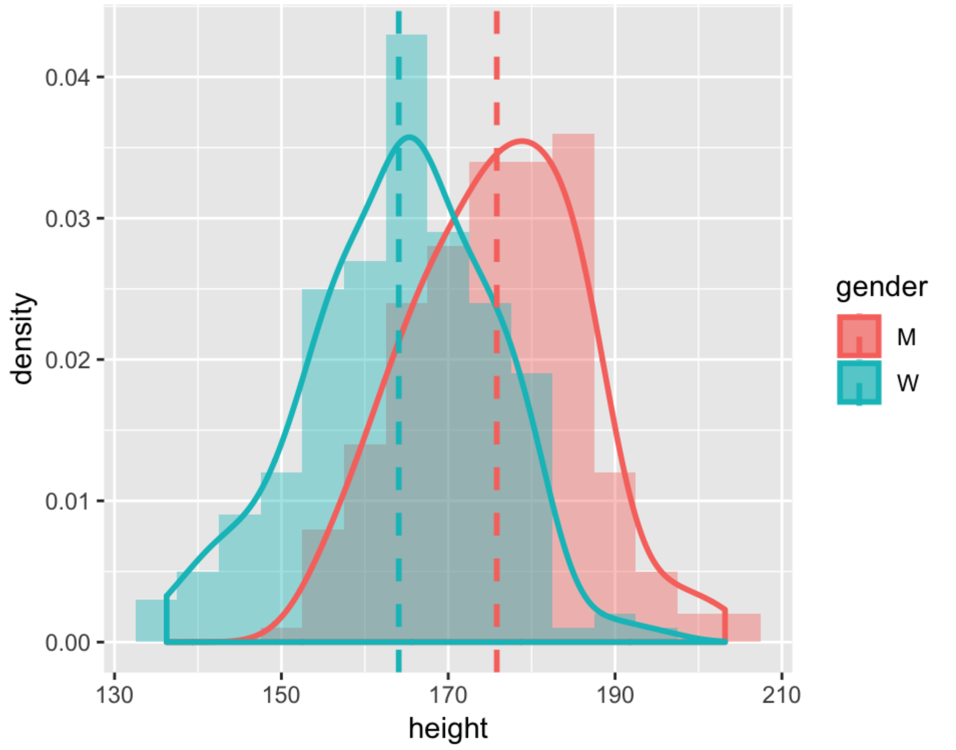
Add mean value

```
library(plyr)

mean_df = ddply(mon_df, "gender", summarise, avg_weight=mean(weight),
                avg_height=mean(height))
```

gender	avg_weight	avg_height
M	77.21	175.82
W	62.15	164.07

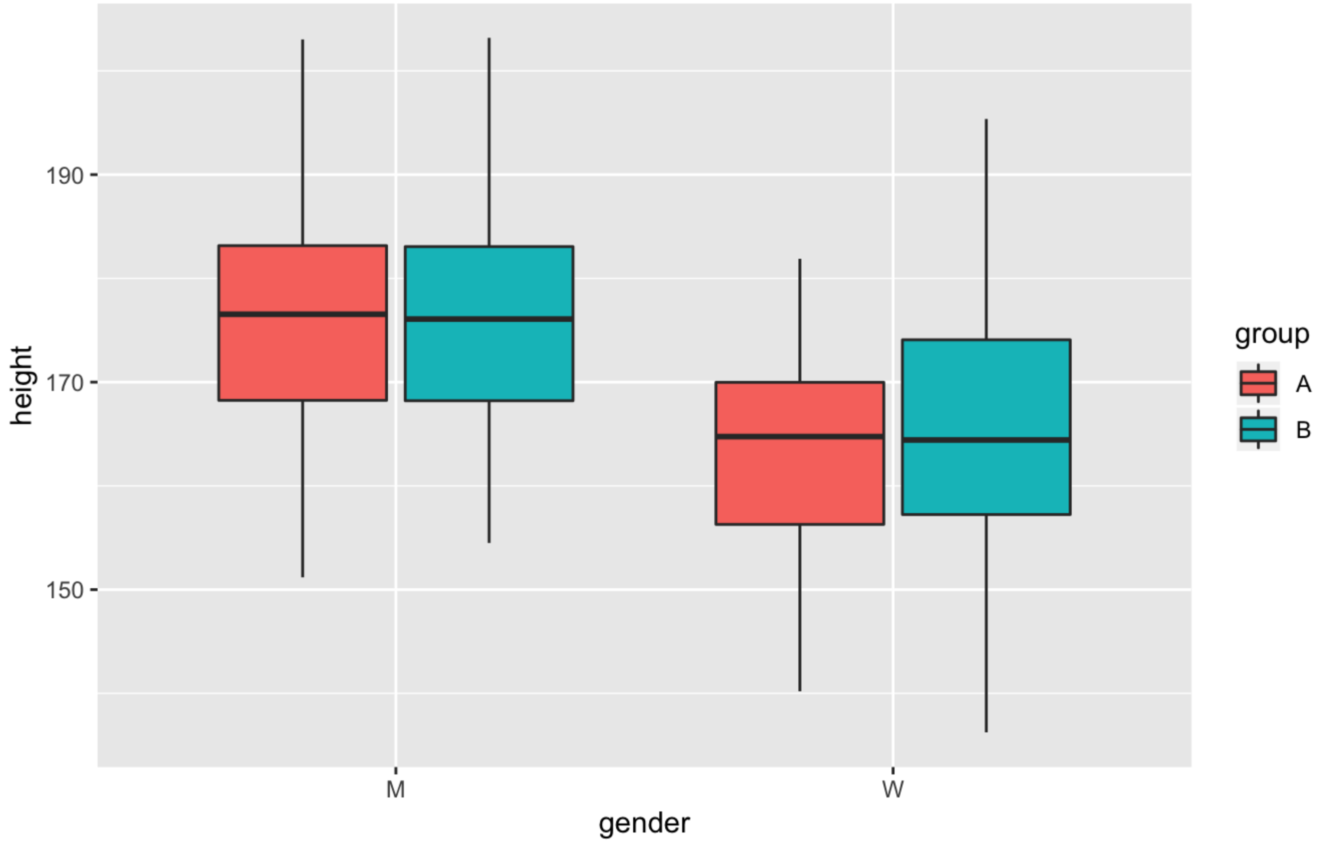
Add mean value (2)

```
ggplot(mon_df, aes(x=height))+
  geom_histogram(aes(y=..density.., fill=gender), position='identity',
                 alpha=0.4, binwidth=5)+
  geom_density(aes(color=gender), alpha=0.5, size=1)+
  geom_vline(data=mean_df, aes(xintercept=avg_height, color=gender),
             linetype="dashed", size=1)
```



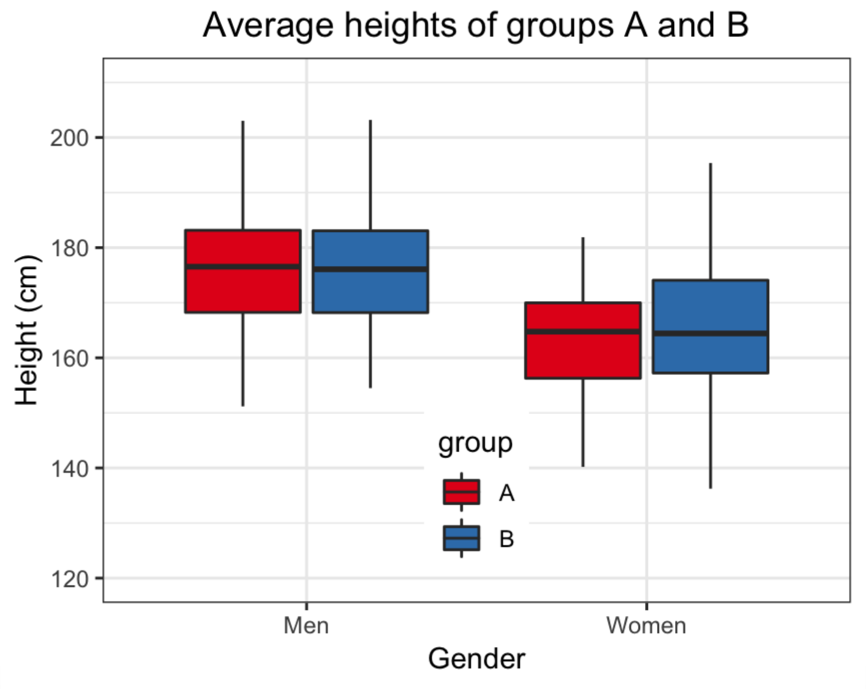
Boxplot

```
ggplot(mon_df, aes(y=height, x=gender, fill=group))+
  geom_boxplot()
```



Boxplot (2)

```
ggplot(mon_df, aes(y=height, x=gender, fill=group))+
  geom_boxplot()+
  scale_fill_brewer(palette="Set1")+
  scale_x_discrete(name ="Gender", labels=c("M"="Men", "W"="Women"))+
  scale_y_continuous("Height (cm)", limits=c(120, 210))+
  ggtitle("Average heights of groups A and B")+
  theme_bw()+
  theme(legend.position=c(0.5, 0.2), plot.title=element_text(hjust=0.5))
```



Simulation data

Used data from a molecular dynamics simulation (DM) article from Laurent *et al.* [1].

- 3 simulations of GLIC (*Open, Closed, LC*) of 1 microseconds.
- simulation analysis centralized in a R data frame

Load data:

```
load("template_data_GLIC.Rda")
```

Filter data:

```
Sim_data_df=subset(df_select, type == "MD")
Cryst_data_df=subset(df_select, type == "Cryst")
Sim_data_df=subset(Sim_data_df, time%%2 == 0)
```

[1] Laurent B, Murail S, Shahsavar A, Sauguet L, Delarue M, Baaden M. *Sites of Anesthetic Inhibitory Action on a Cationic Ligand-Gated Ion Channel*. **Structure** (2016), *24(4), 595-605.*

Simulation data (2)

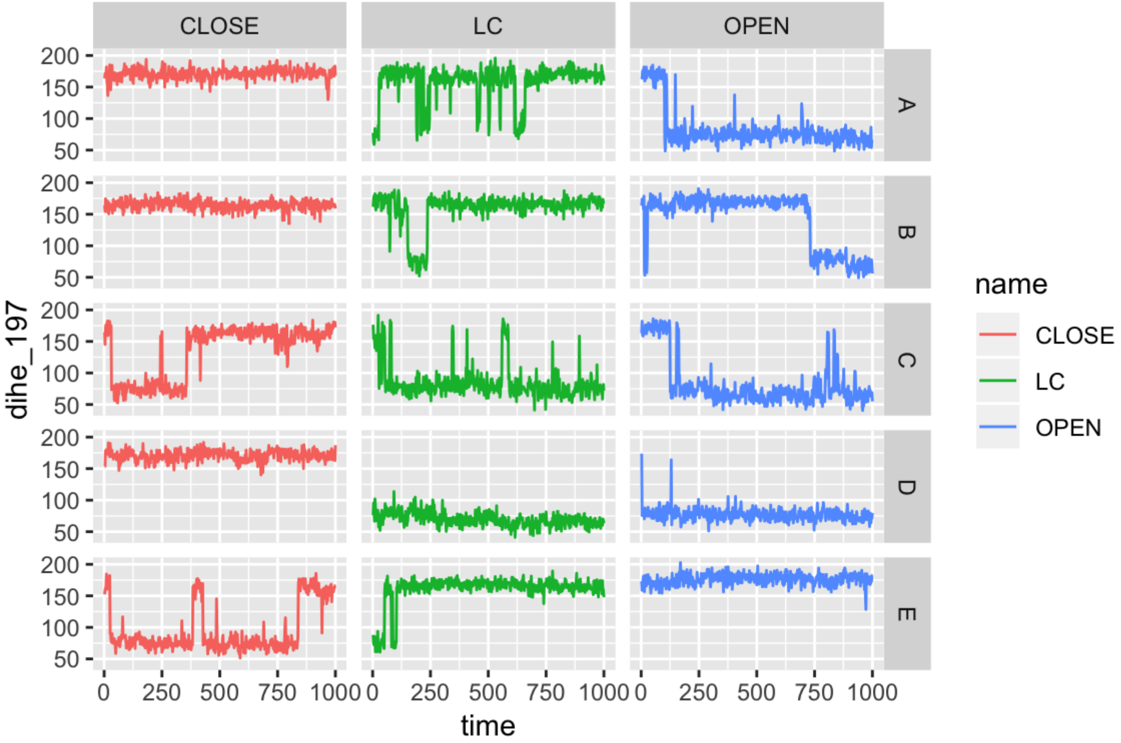
	Br_Occupancy	Cav_Vol	dihe_197	chain	sim_id	time	type	name
4	0	88.9	163.6	A	1_A	2	MD	CLOSE
8	0	44.4	171.4	A	1_A	4	MD	CLOSE
12	0	21.1	171.9	A	1_A	6	MD	CLOSE
16	0	61.0	169.0	A	1_A	8	MD	CLOSE
20	0	94.1	181.9	A	1_A	10	MD	CLOSE
24	0	28.8	159.5	A	1_A	12	MD	CLOSE
28	0	71.5	153.4	A	1_A	14	MD	CLOSE
32	0	46.1	136.6	A	1_A	16	MD	CLOSE
36	0	73.8	161.9	A	1_A	18	MD	CLOSE
40	0	50.9	146.6	A	1_A	20	MD	CLOSE

```
## [1] "Number of rows: " "30035"
```

Simulation data (3)

Dihedral angle time series:

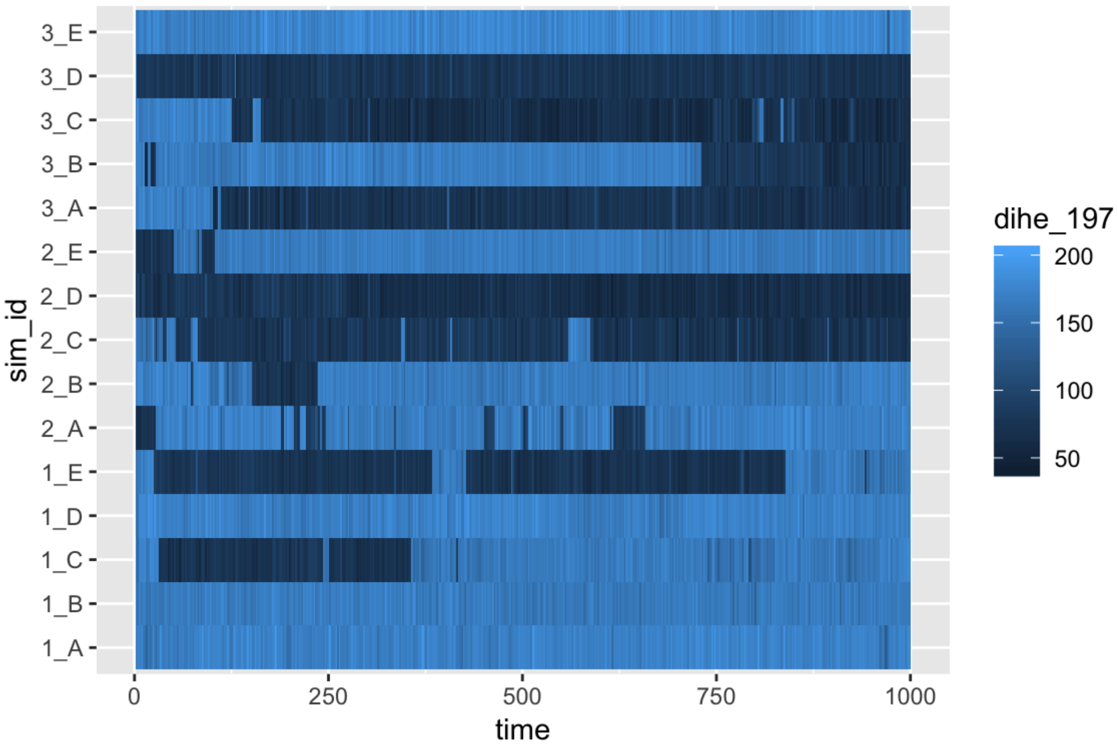
```
ggplot(Sim_data_df, aes(x=time, y=dihe_197, group=sim_id, color=name)) +
  geom_line()+
  facet_grid(chain ~ name)
```



Simulation data (4)

Dihedral angle time series as a matrix with continuous values:

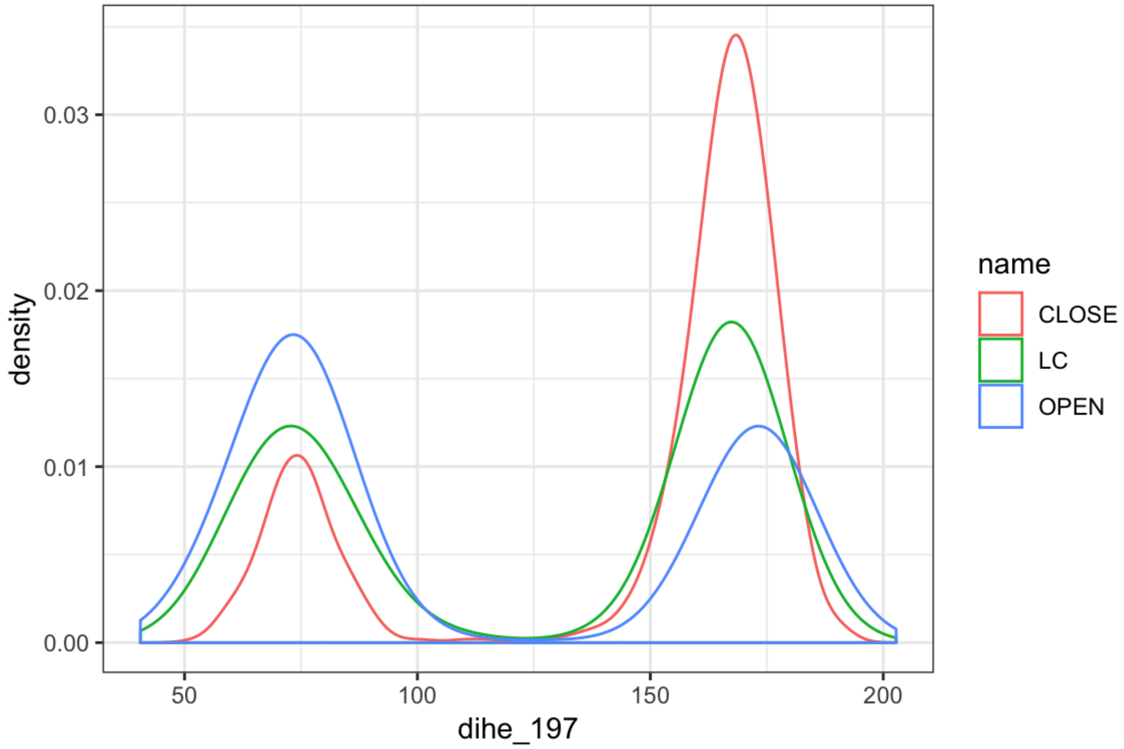
```
ggplot(Sim_data_df, aes(x=time, y=sim_id, fill=dihe_197)) +
  geom_tile()
```



Simulation data (5)

Dihedral angle distribution:

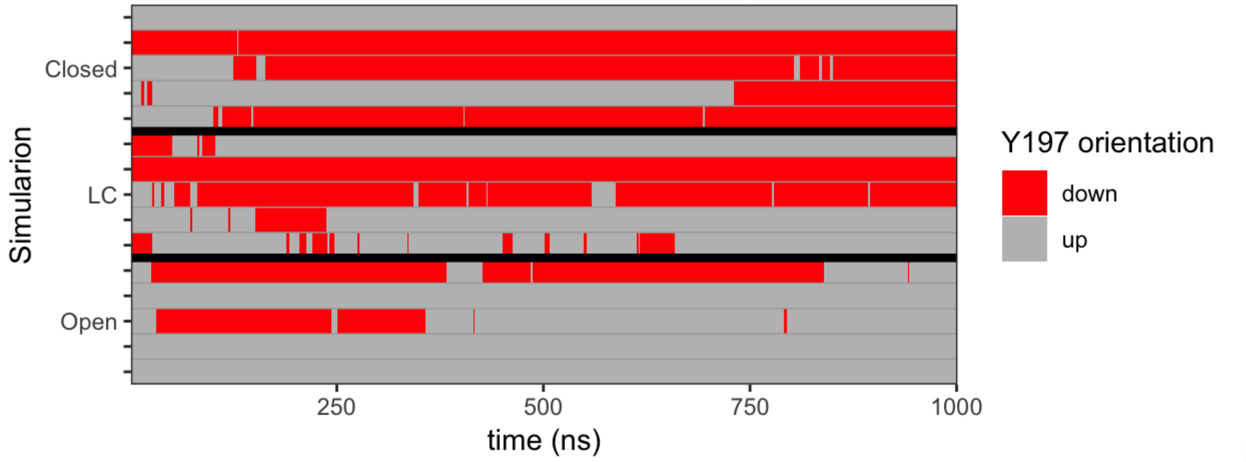
```
ggplot(Sim_data_df, aes(x=dihe_197, color=name)) +
  geom_density()+
  theme_bw()
```



Simulation data (6)

Dihedral angle time series as a matrix with discrete values:

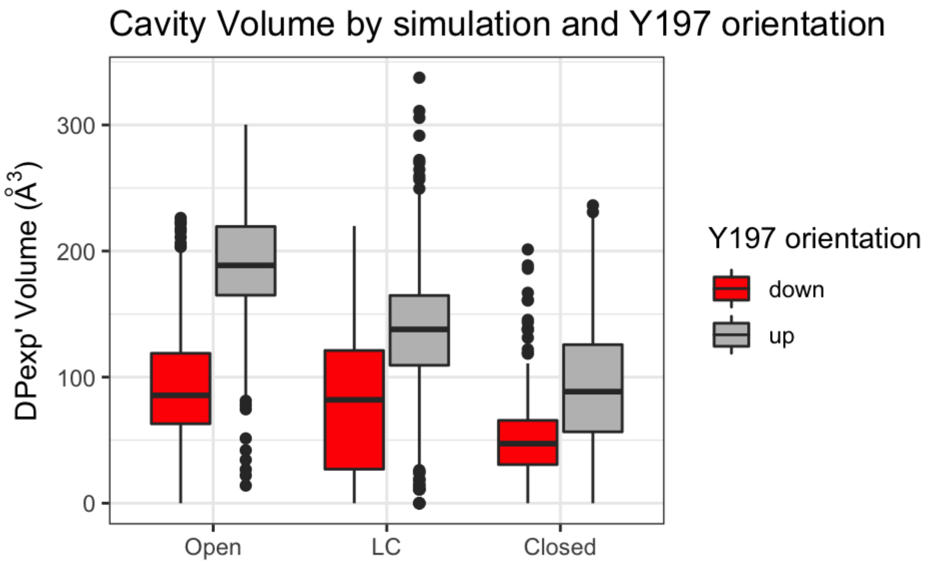
```
# Add a new column:
Sim_data_df$dihe_197_bin=as.factor(cut(Sim_data_df$dihe_197,breaks=seq(0,240,120)))

ggplot(Sim_data_df, aes(x=time, y=sim_id, fill=dihe_197_bin)) +
  geom_tile()+
  scale_fill_manual(name="Y197 orientation", labels=c("down", "up"), values=c("red", "gray"))+
  geom_hline(data=data.frame(y=1:14), aes(yintercept=y+0.5), size=0.25, color="darkgray")+
  geom_hline(data=data.frame(y=c(5,10)), aes(yintercept=y+0.5), size=1.5)+
  scale_x_continuous(name="time (ns)", expand=c(0, 0))+
  scale_y_discrete(name="Simularion", expand=c(0, 0), label=c('','',"Open",'','','','',"LC",'','','','',"Closed",'',''
  theme_bw()
```



Simulation data (7)

Boxplot for publication:

```
ggplot(Sim_data_df, aes(y=Cav_Vol, x=name, fill=dihe_197_bin)) +
  geom_boxplot() +
  ggtitle("Cavity Volume by simulation and Y197 orientation") +
  scale_y_continuous(name=expression(paste("DPexp' Volume (",ring(A),''^{3},")")))+
  scale_fill_manual("Y197 orientation",labels=c("down", "up"),values=c("red","gray"))+
  scale_x_discrete("", limits=c("OPEN","LC","CLOSE"), label=c("Open", "LC", "Closed"))+
  theme_bw()
```



Saving graphics

Extremly simple:

```
ggsave('my_graph.pdf')
ggsave('my_graph.png')
```

Be carefull with vectoriel graphics like `.pdf` or `.ps`, as if the graphics contains more than 1.000-10.000 point/lines, it could be really slow to open and print. In that case preferably use bitmap like files : `.png`, `.jpg`.

plotly

Interactive graphic but only works in html format.

```
library(ggplot2)
library(plotly)
p = ggplot(mon_df, aes(weight, height, size=gender, color=group)) +
  geom_point(alpha=0.5) +
  theme_bw()
ggplotly(p)
```

Going further

- Great choices of graphic representations:
  - Polar coordinates
  - Geographic maps
  - matrices
  - …
- References:
  - Ggplot cheatsheet
  - ggplot2: Elegant Graphics for Data Analysis by Hadley Wickham
- Graphics Gallery:
  - The R Graph Gallery