Sweet insets for extreme data

Florent Déry

26/02/2020

Sometimes, we have rare extreme data in a dataset that are informative, but enlarge so much the scale of the response variable we are looking at that we can't see nothing among non-extreme variables.

There is some possibilities when using `base` functions to generate figures (e.g. breaks in axis: through `R` package `plotrix`), but `ggplot2` fans are left with nothing except anger and resentment. However, some turn-arounds exist, such as including insets showing the non-extreme variables in a plot that contains all values.

This mini-tutorial shoes how to do this with `ggplot2` figures, by bulding mainplot and inset figures that you'll either combine through `grid` package or `cowplot` package, often already installed by default on `R`. Make sure you have both packages to run what follows.

1. Generate fake data with extreme values

Figure 1: *Me in X-Country ski* ^^

This dataset (`df`) is fictive and represent the number of falls/hour (`y`) that one "could" take depending on your experience in Cross-Country skiing (`x` ; weeks). I hope that everybody was or will be better than that when they first try X-country ski:

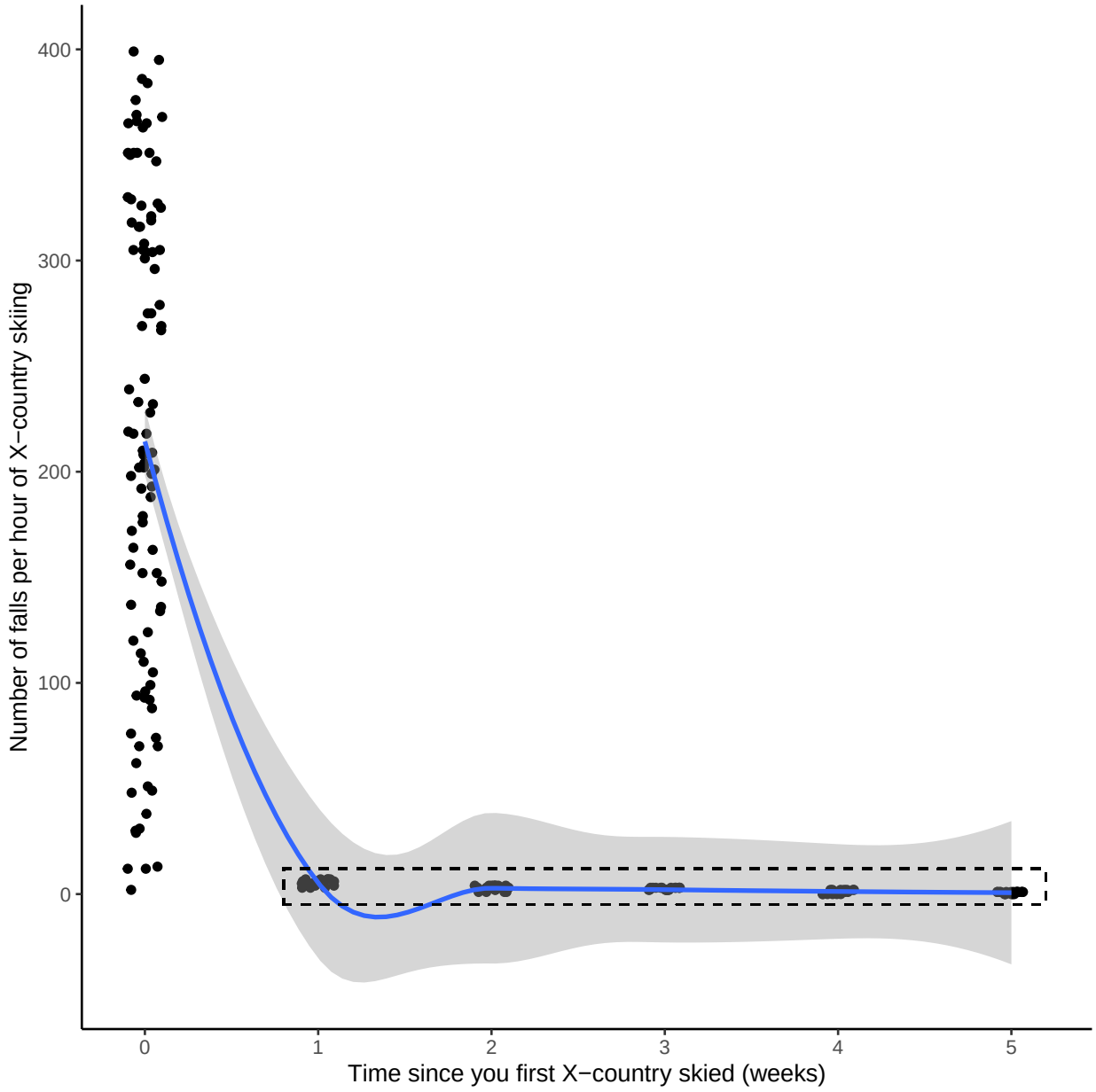
```
setwd("C:/master/other/afs/workshops/workshops/insets_in_plots")
df=data.frame(
  y=c(
    sample(c(0:400), 100, replace=T),
    sample(c(3:7), 20, replace=T),
    sample(c(1:4), 20, replace=T),
    sample(c(2:3), 20, replace=T),
    sample(c(0:2), 20, replace=T),
    sample(c(0:1), 20, replace=T)
```

```
  ),
  x=c(
    rep(0, times=100),
    rep(1, times=20),
    rep(2, times=20),
    rep(3, times=20),
    rep(4, times=20),
    rep(5, times=20)
  )
)
```

---

2. Generate the base figure

```
mainplot <- ggplot(data = df, aes(y = y, x = x) ) +
  geom_jitter(width = 0.1, height = 0) +
  geom_smooth()+
  geom_rect(data=df,aes(xmin=0.8,
                        xmax = 5.2,
                        ymin = -5,
                        ymax = 12), color = "black", fill=NA,
            linetype = 2)+
  xlab('Time since you first X-country skied (weeks)')+
  ylab('Number of falls per hour of X-country skiing')+
  theme_bw()+
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour="black", linetype="solid"),
        panel.background = element_rect(fill = "white",
                                        colour = "white",
                                        size = 0.5,
                                        linetype = "solid"),
        panel.border = element_blank())
mainplot
```



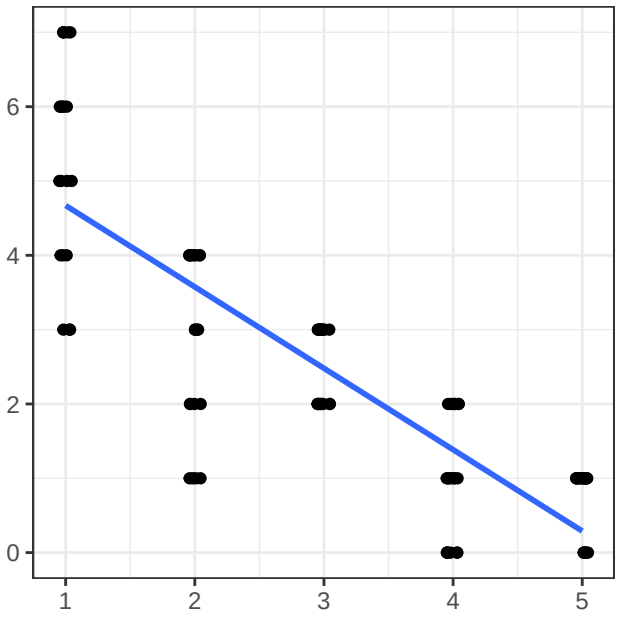
From that, We don't see if there is an amelioration or not after the first trial (dashed rectangle), although we are still amazed that this wanna be skier kept trying after its first time!

---

3. Generate the inset figure

```
inset=ggplot(data=df[df$x>0,],
             aes(x=x, y=y)) +
  geom_jitter(width = 0.05, height = 0) +
  geom_smooth(method = lm, se = F)+
```

```
  xlab('')+
  ylab('')+
  theme_bw()
inset
```



Ok, So now we have a better idea! Let's combine both!

---

4. Adding an inset ggplot figure to another main ggplot figure using `grid`!

```
png(file="figures/fast_learner_fast_skier.png",
    w=7,
    h=7,
    res=720,
    units = 'in')# Where we save our figure as a PNG

  grid::grid.newpage()

  v1<-grid::viewport(width = 1, height = 1, x = 0.5, y = 0.5)
  # Viewport 1 (plot area for the main plot)
  v2<-grid::viewport(width = 0.6, height = 0.5, x = 0.65, y = 0.71)
  # Viewport 2 (plot area for the inset plot)

  print(mainplot,vp=v1)
  print(inset,vp=v2)

dev.off()#clear device and save figure
```

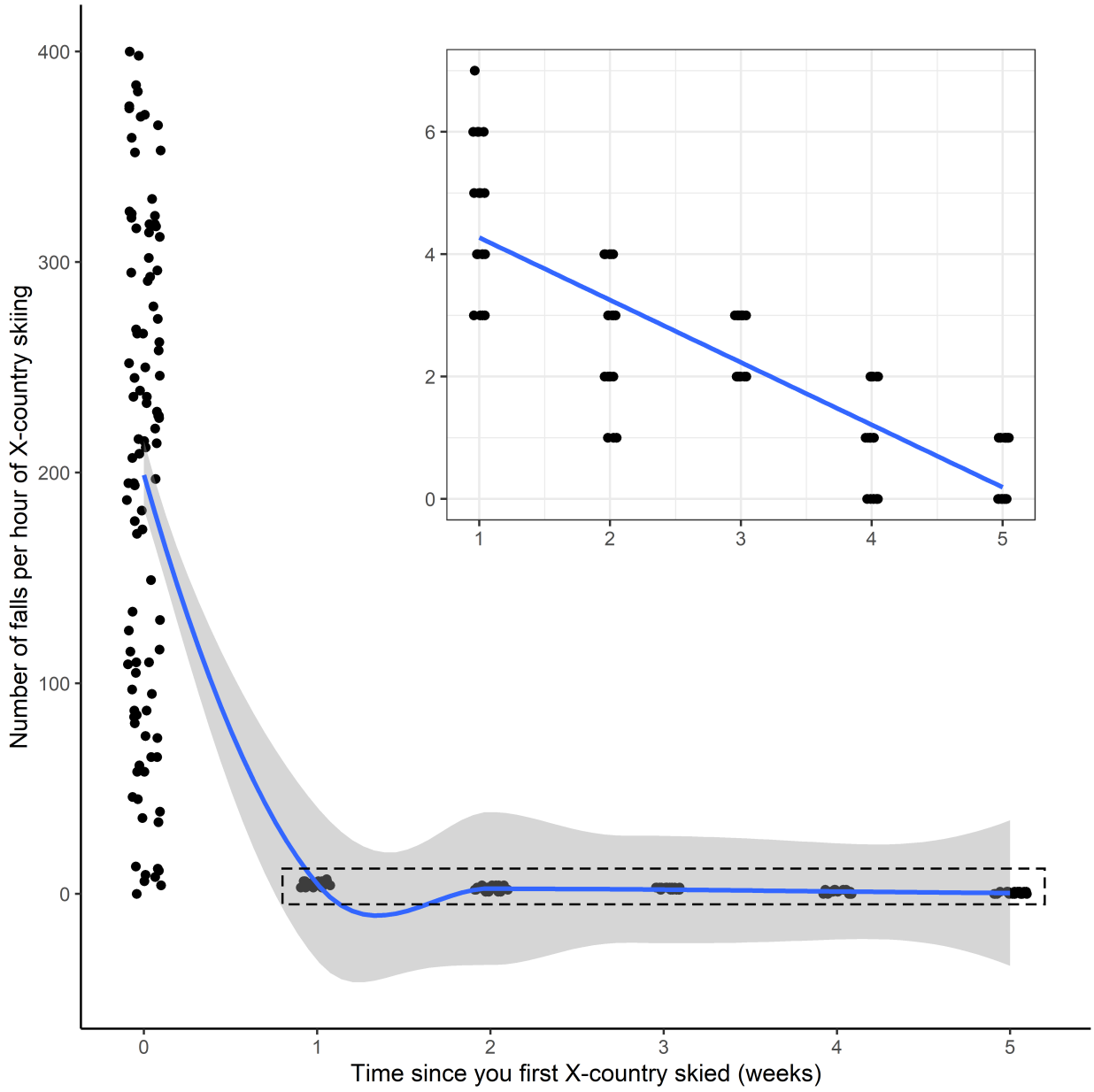


Figure 2: Ski learning curve, with `grid` ^^

5. Adding an inset ggplot figure to another main ggplot figure using `cowplot`!

And even adding arrows and rectangle!

```
combined=ggdraw()+
  draw_plot(mainplot, x=0,y=0,height=1, width = 1)+
  draw_plot(inset, x=0.35,y=.45, height = 0.5, width=0.6)+
  draw_line(x= c(.26, 0.4),
            y= c(0.18, 0.52),
            linetype = 2,
            arrow = arrow(length = unit( .01, "npc"), type = "closed")
            )+
  draw_line(x=c(0.95,0.94),
            y=c(0.18,.52),
            linetype = 2,
            arrow = arrow(length = unit( .01, "npc"), type = "closed"))

ggsave(combined, filename ="figures/fast_learner_fast_skier2.png",
       width = 7,
       height = 7) #alternate way to save it
```

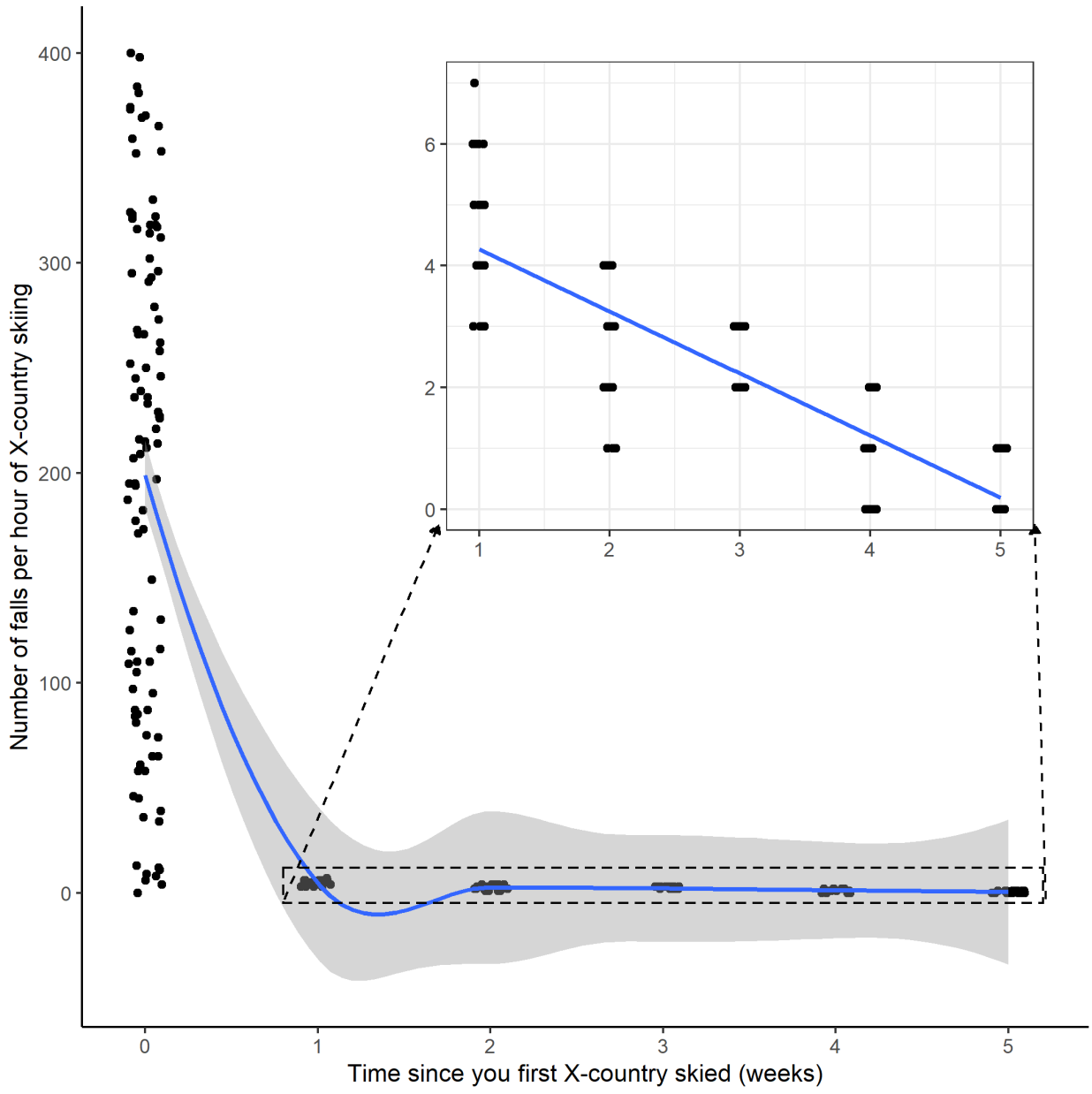


Figure 3: Ski learning curve, with `ggplot2` ^^