Univeariate analysis: Historgrams and Density plots

2024-09-05

```r
params = list(
  truc= "Science des Données",
  year= 2023 ,
  curriculum= "L3 MIASHS",
  university= "Université Paris Cité",
  homepage= "https://stephane-v-boucheron.fr/courses/scidon",
  moodle= "https://moodle.u-paris.fr/course/view.php?id=13227",
  path_data = './DATA',
  country_code= '...',
  country= '...',
  datafile= '...'
  )

attach(params)
```

```r
stopifnot(
  require(patchwork),
  require(glue),
  require(here),
  require(tidyverse),
  require(ggmosaic),
  require(skimr),
  require(plotly),
  require(DT),
  require(GGally),
  require(ggforce),
  require(ggfortify),
  require(vcd)
)

tidymodels::tidymodels_prefer(quiet = TRUE)

old_theme <-theme_set(theme_minimal(base_size=9, base_family = "Helvetica"))
```

- **L3 MIASHS**
- **Université Paris Cité**
- Année 2023-2024
- Course Homepage
- Moodle

! Objectives

Density estimation

Histogram

A histogram is a piecewise constant density estimator.

Sliding window estimator

Let $h > 0$ be a bandwidth, let $x_1, \ldots, x_n$ be a sample, the sliding window density is defined by

$$\hat{f}_n(x) = \sum_{i=1}^{n} \frac{1}{2h} \mathbb{I}_{[-1/2,1/2]} \left(\frac{x - x_i}{h}\right)$$

ou

$$\hat{f}_n(x) = \frac{1}{2h} \left(F_n(x+h) - F_n(x-h)\right)$$

Kernel density estimator

Simulations

Question

Simulate $N = 10$ samples of size $n = 500$ from a mixture of two Gaussian distributions $\lambda\mathcal{N}(0,1) + (1-\lambda)\mathcal{N}(\mu, \sigma^2)$.
Henceforth, λ is the *mixing* parameter. $\mathcal{N}(0,1)$ is the standard Gaussian and $\mathcal{N}(\mu, \sigma^2)$ is the non-standard Gaussian component of our *mixture* distribution,

Mixture distributions

```r
mu <- 2 ; sigma <- 0.5  # parameters o the non-standard Gaussian
N <- 10 ; n <- 10000  # number of replicates ; sample sizes
lambda <- .4     # mixing parameter

dmix <- \(x) lambda*dnorm(x)+ (1-lambda)*dnorm(x, mu, sigma)
```

We can first adopt a naive approach to simulation

```r
x <- rep(0, n*N)

for (i in seq(1,n*N)){
  cpn <- sample(c(1,2), 1, prob = c(lambda, 1-lambda))
  x[i] <- ifelse(cpn==1, rnorm(1), rnorm(1, mu, sigma))
}
```

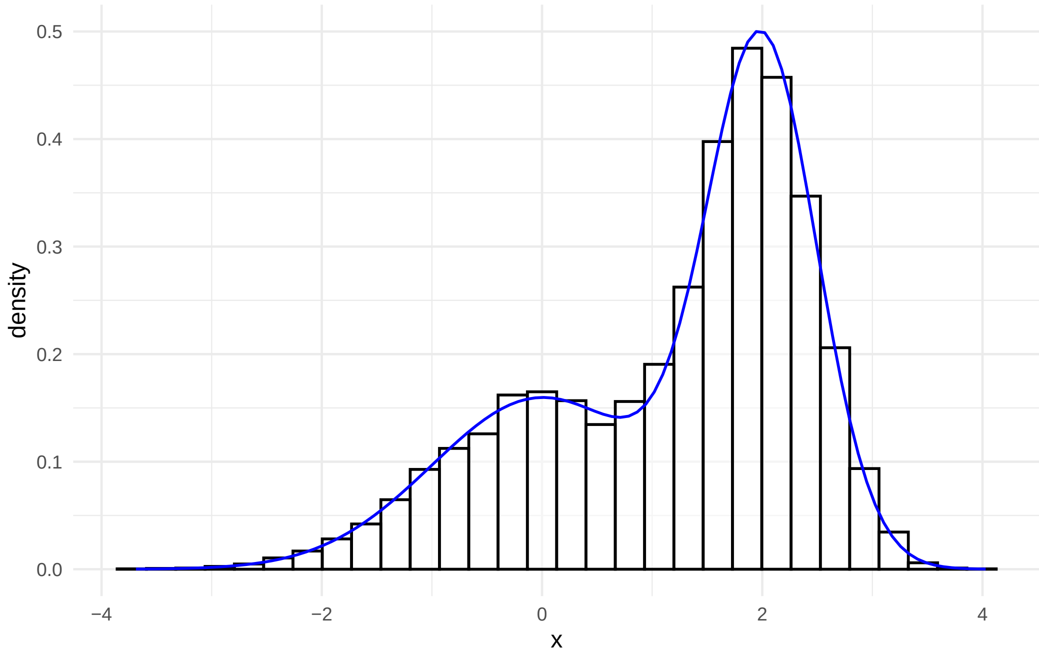
```r
c_x <- sample(c(1,2), n*N, replace=T, prob = c(lambda, 1-lambda))    # sample the Bernoulli
x <- c(0, mu)[c_x] + c(1, sigma)[c_x] * rnorm(n*N) # opportunistic sampling
```

```r
M <- matrix(x, nrow = n, ncol = N)

df <- as.data.frame(M)
df <- as_tibble(df)
```

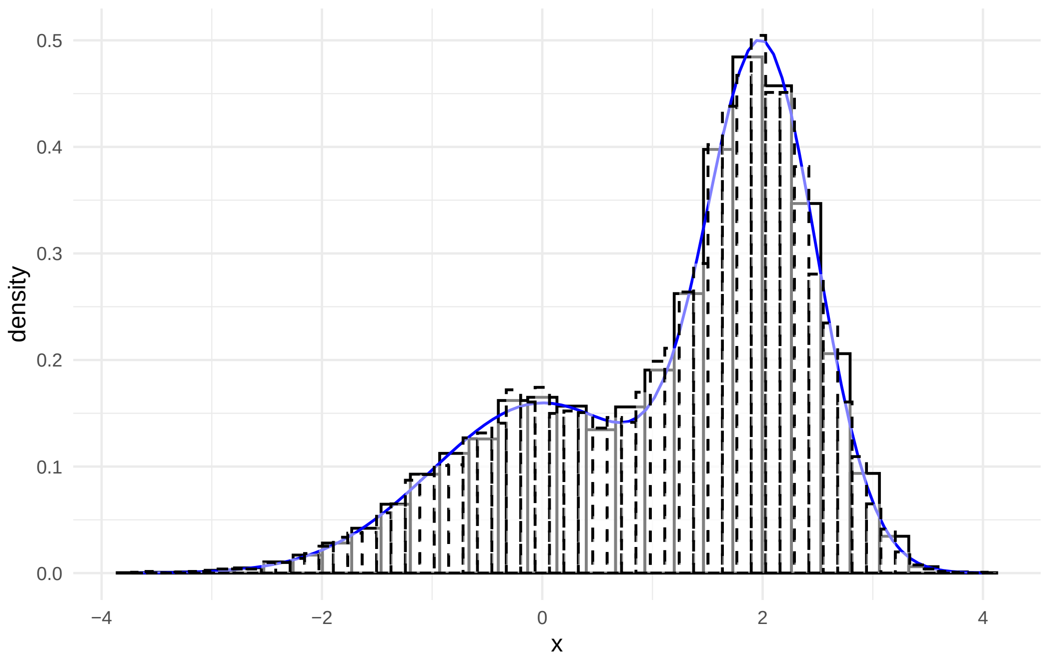
Question

Plot regular histograms for different sample replicates.
Try different number of `bins` or `binwidths`.

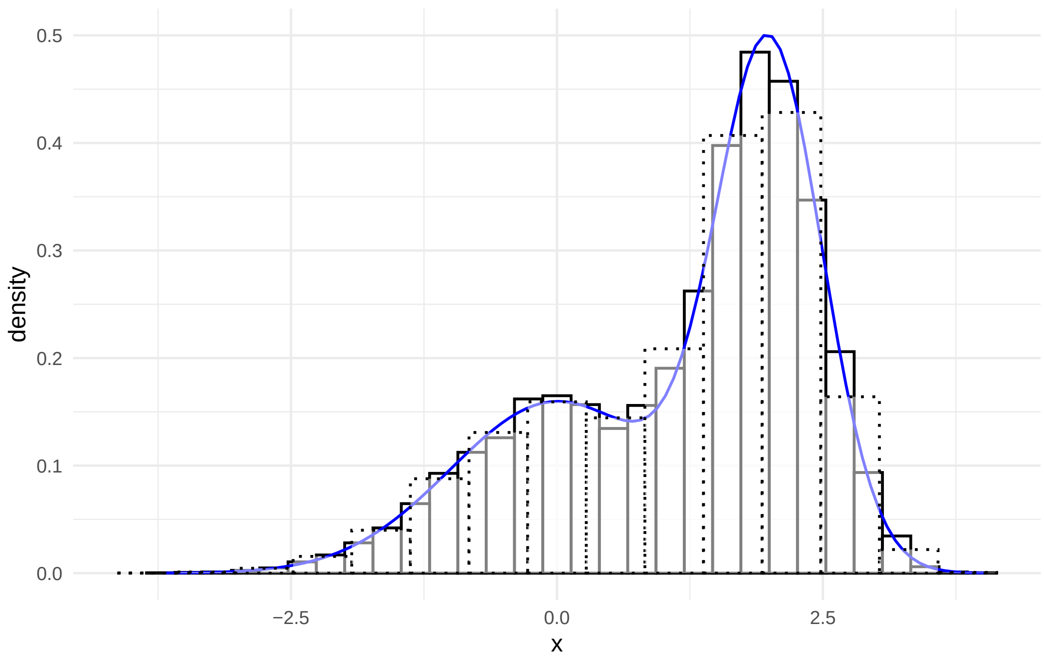
```r
p <- df |>
  ggplot() +
  aes(x=V1, y=after_stat(density)) +
  geom_histogram(bins= 30, fill="white", color="black", linetype=1, alpha=.5) +
  xlab("x") +
  stat_function(inherit.aes = F,
                fun = dmix,
                color="blue")

p
```




```r
p +
  geom_histogram(bins= 60, fill="white", color="black", linetype=2, alpha=.5)
```



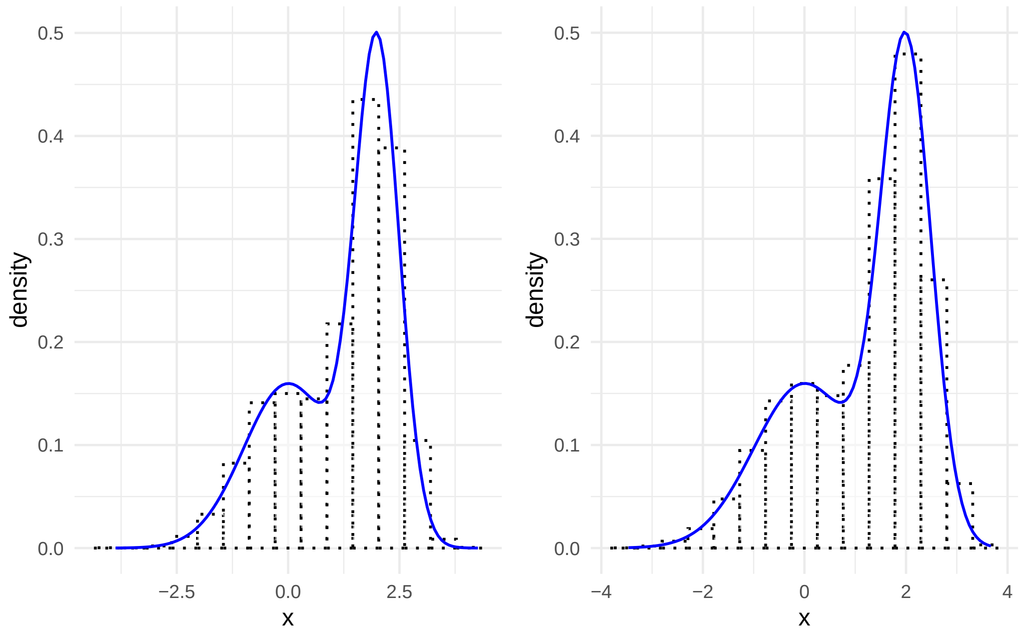

```r
p +
  geom_histogram(bins= 15, fill="white", color="black", linetype=3, alpha=.5)
```



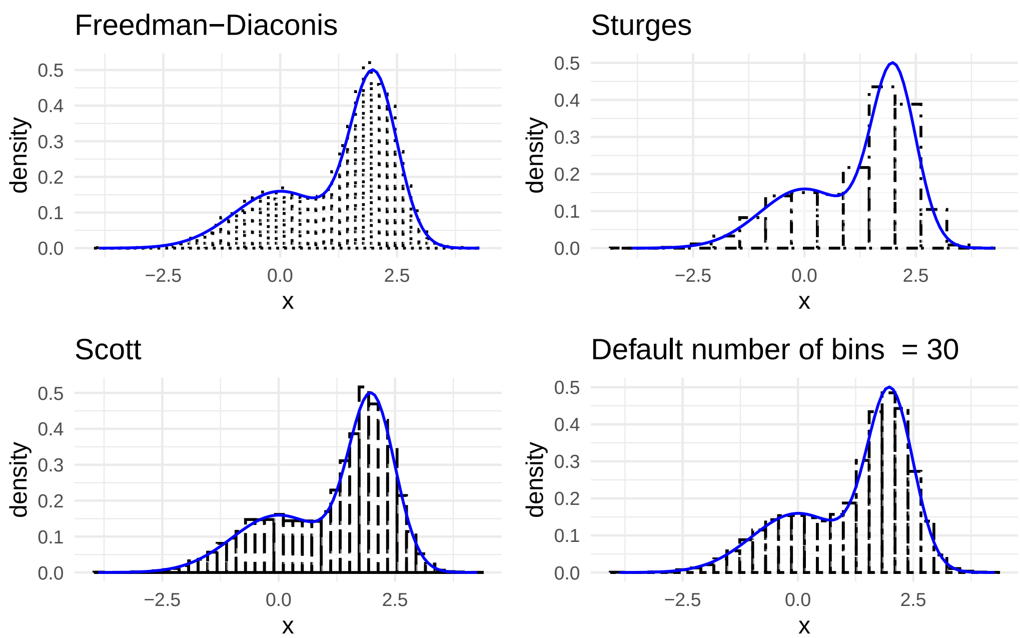

```r
my_histo <- function(df, col, dfun, ...){
  df  |>
  ggplot() +
  aes(x={{col}}, y=after_stat(density)) +
  geom_histogram(...) +
  xlab("x") +
  stat_function(inherit.aes = F,
                fun = dfun, color="blue")

}
```

```r
p2 <- my_histo(df, V2, dmix,  bins= 15, fill="white", color="black", linetype=3, alpha=.5)
p3 <- my_histo(df, V3, dmix,  bins= 15, fill="white", color="black", linetype=3, alpha=.5)
```

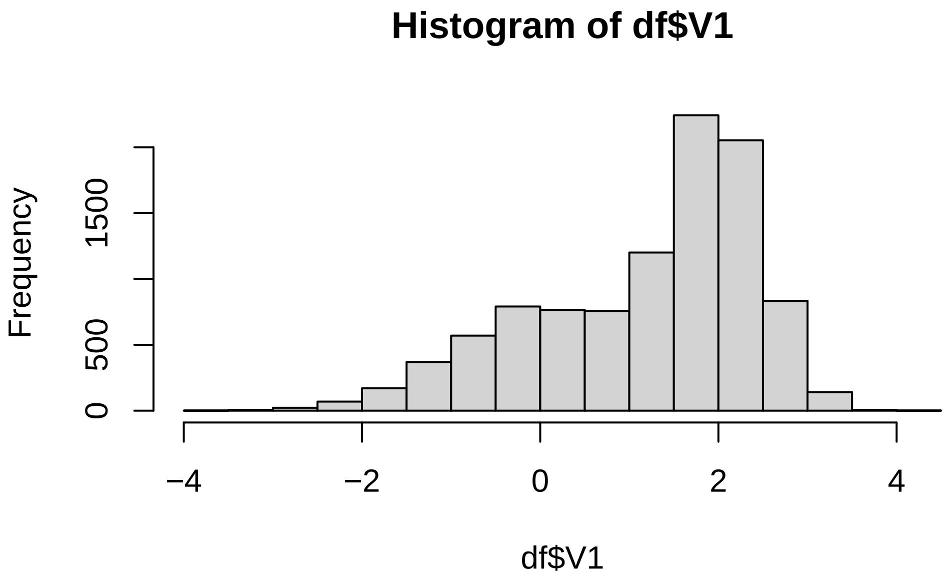
```r
p2 + p3
```




```r
pfd <- my_histo(df, V2, dmix,  bins= nclass.FD(df$V2), fill="white", color="black", linety

psturges <- my_histo(df, V2, dmix,  bins= nclass.Sturges(df$V2), fill="white", color="blac

pscott <- my_histo(df, V2, dmix,  bins= nclass.scott(df$V2), fill="white", color="black",

p30 <- my_histo(df, V2, dmix,  bins= 30, fill="white", color="black", linetype=6, alpha=.5

(pfd + psturges) / (pscott + p30)
```




```r
hist(df$V1)
```

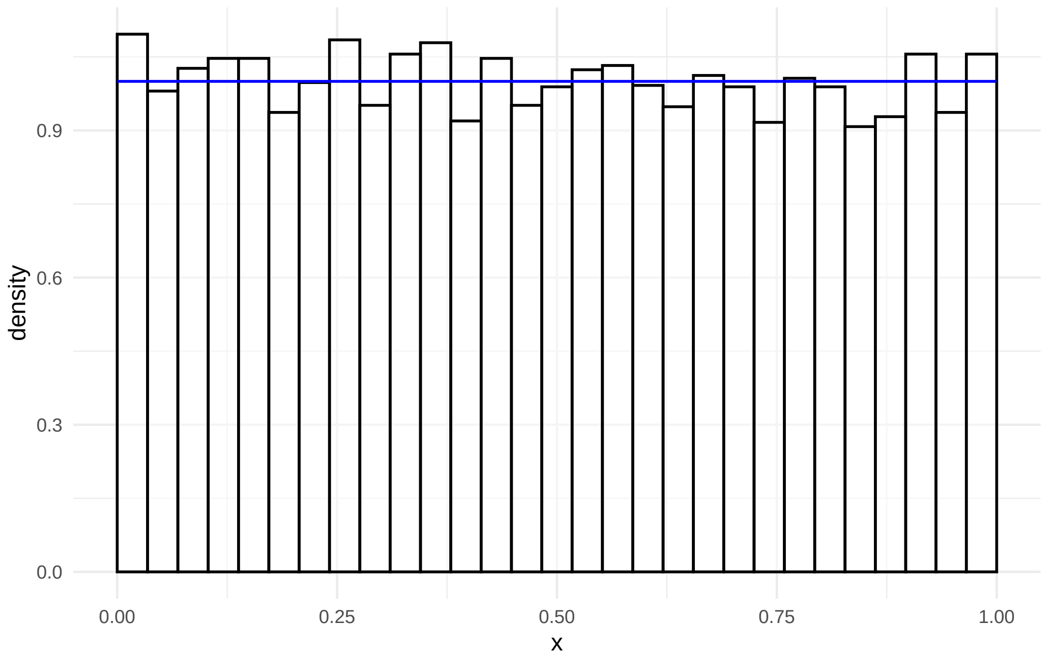


Question

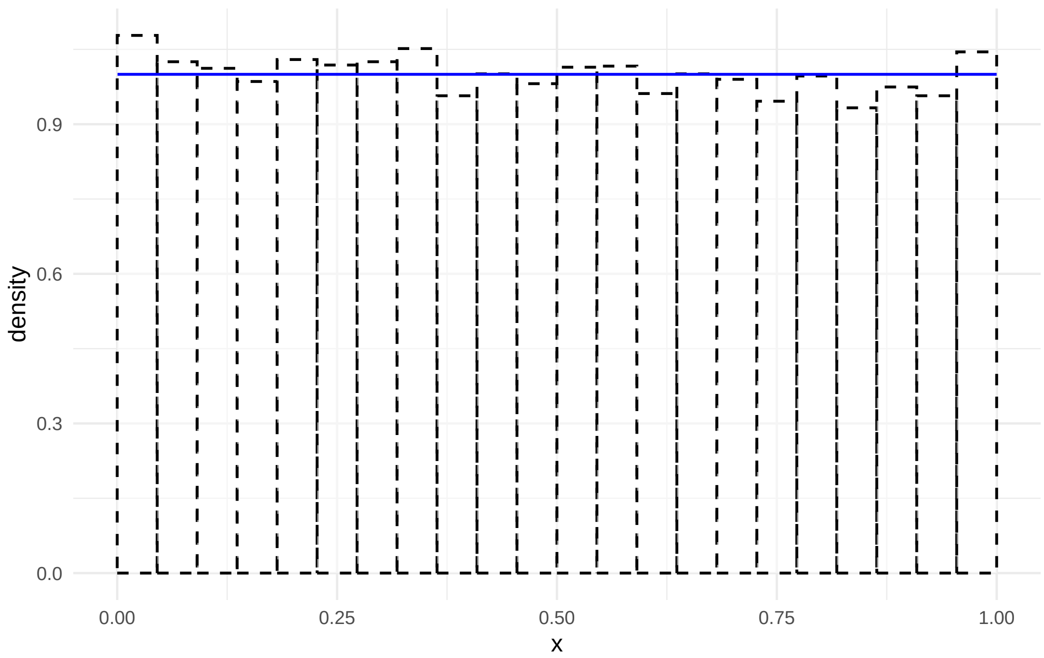
Repeat the above operations, but sample according the uniform distribution on $[0,1]$ but choose the breaks so that the intervals all have the same probability under the sampling distribution.

```
N <- 100
M <- matrix(runif(N * n), nrow=n)
df <- as.data.frame(M)
df <- as_tibble(df)
```

```
breaks <- seq(0, 1, length.out=30)

my_histo(df, V2, dunif,  breaks=breaks, fill="white", color="black", linetype=1, alpha=.5)
```




```
my_histo(df, V2, dunif,
         breaks=seq(0, 1, length.out=nclass.scott(df$V2)+1),
         fill="white", color="black", linetype=2, alpha=.5)
```



Question

Assume that you have chosen B bins.

- What is the distribution of the the number of sample points in a bin?
- What is the average number of points in a bin, what is its variance?
- Provide an upper bound on the expected maximum number of points in a bin.

Question

Assume that you have chosen B bins.
Compare the *empirical* distribution of the number of points in a bin with the theoretical distribution of the number of points in a bin.

```
B <- 30

df_counts <- df |>
  mutate(across(everything(), \(x) cut(x,breaks)))
```

```
df_counts$V1 |>
  table() |>
  as.numeric() |>
  table()
```

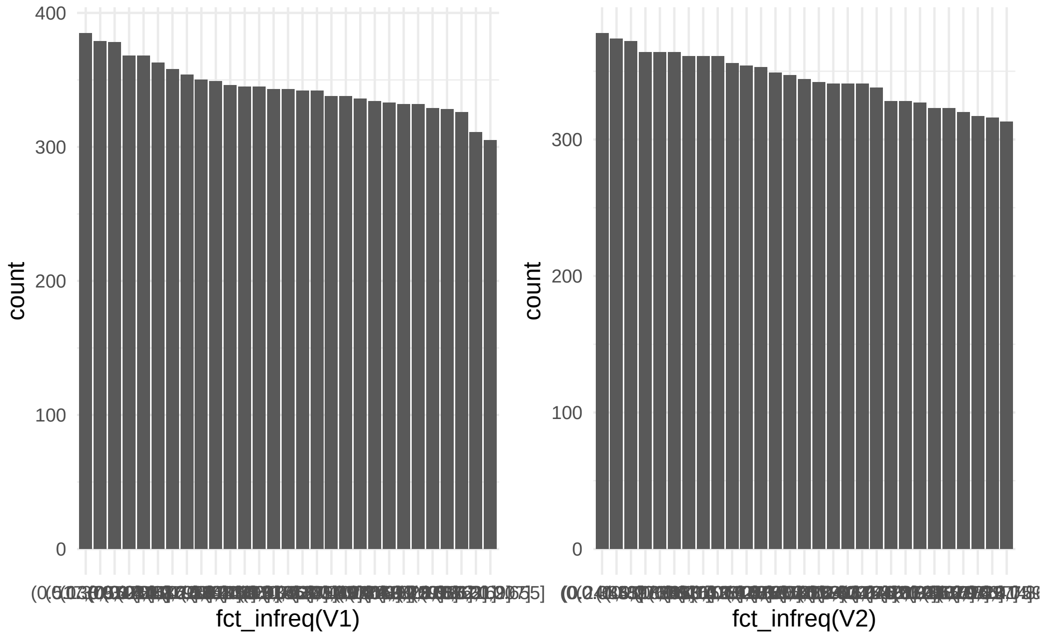
```
305 311 326 328 329 332 333 334 336 338 342 343 345 346 349 350 354 358 363 368
  1   1   1   1   1   2   1   1   1   2   2   2   2   1   1   1   1   1   1   2
378 379 385
  1   1   1
```

```
df_profiles <- df_counts |>
  summarise(across(everything(), \(x) list(table(table(x)))))
```

```
my_bar <- function(df, col) {
  df_counts |>
    ggplot() +
    aes(x=fct_infreq({{col}})) +
```

```
    geom_bar()
}
```

```
p1 <- my_bar(df_counts, V1)
p2 <- my_bar(df_counts, V2)

p1 + p2
```




```
max(names(df_profiles$V1[[1]]))
```

```
[1] "385"
```