Simulations, CLT and EVT

Sayani Gupta

25/08/2020

Discussion: Let the sample size be n and the number of iterations be k. By CLT (or EVT), when n is high enough, then the limiting distribution of sample mean (or max) goes to a standard normal (or Gumbel/Freschet/Weibull) distribution, irrespective of the underlying distribution of the sample. This means when n is high enough and we sample n iid random variables multiple times (high enough k), by CLT, the distribution of sample mean would look like standard normal. What if n is very small and k is large? Would the distribution of sample mean still look like standard normal? It would not. Similarly, EVT works when n is high enough. When we are taking the max of 2 or 3 rvs ("levels" in our terminology), $n = 2/3$, which are far too less. Hence, normalisation would be able to nullify the effect of n only for large enough n.

Simulation Study:

Samples are taken from Gamma(shape = 3, rate = 0.5) distribution and $k = 500$ in all the cases. Small values of n considered in the range $2, 3, \ldots, 10$ and large values of n considered within $25, 30, 35, \ldots, 45$. We study the distribution of:

1. sample means for small n (Obs: distribution doesn't look similar or normal)
2. sample means for large n (Obs: distribution looks similar and normal like with different scales)
3. normalised sample means for small n (Obs: distribution doesn't look similar)
4. normalised sample means for large n (Obs: distribution looks similar in both shape and scale with mean 0)
5. sample maximums for small n (Obs: distribution doesn't look too different except for tails, high scale)
6. sample maximums for large n (Obs: distribution looks similar and positively skewed with very high scales)
7. normalised sample maximums for small n (Obs: distribution doesn't look similar, scaled down, maybe similar after facet label 5)
8. normalised sample maximums for large n (Obs: distribution looks similar and it has been scaled down much more than maximum without normalisation. Can we use this fact to explore maximums more?)

Question: So we cannot normalise for the number of levels using limit theorems when we are talking about normalising for few levels? By normalisation, if we mean adjusting values measured on different scales to a common scale, then that could be done through limit theorems. If the intention is to align distributions to a normal like distribution, then we have to resort to something else like distribution transformation?

CLT

Suppose that $X_1, X_2, \ldots, X_n$ are i.i.d. random variables with expected values $E(X_i) = \mu < \infty$ and variance $Var(X_i) = \sigma^2 < \infty$. Suppose the sample mean is $\bar{X} = \frac{(X_1+X_2+\ldots+X_n)}{n}$, then by CLT the distribution of the normalised random variable: $Z_n = \frac{\bar{X}-\mu}{\sqrt{n}\sigma}$ converges in distribution to a standard normal distribution.

```
shape1 <- 3
rate1 <- 0.5
```

```
# create data for each levels

create_simdata <- function(nx = 2,
                           # number of rvs to generate
                           nsim = 500,
                           # number of simulations
                           sim_dist = distributional::dist_gamma(
                             shape = shape1,
                             rate = rate1
                           ),
                           # distribution of each rv
                           create_fun = mean
                           # function used on the rvs
) {
  mean_dist <- (1:nsim) %>%
    purrr::map(function(i) {
      x <- sim_dist %>%
        distributional::generate(nx) %>%
        unlist()
      x %>% create_fun()
    })
}
```

```
create_nlevels <- function(nlevels = seq(2, 10, 2),
                           nsim = 500,
                           sim_dist = distributional::dist_gamma(
                             shape = shape1,
                             rate = rate1
                           ),
                           # distr = "norm",
                           create_fun = mean) {
  data_orig <- nlevels %>%
    purrr::map_df(function(i) {
      create_datai <- create_simdata(nx = i, nsim, sim_dist, create_fun)
      tibble::tibble(ind = i, sim_data = create_datai)
    })
  data_orig %>% unnest(sim_data)
}
```

```
norm_mean <- function(x = NULL,
                      shape = shape1,
                      rate = rate1) {
  meanx <- shape / rate
  sdx <- shape / (rate^2)
```

```
  (mean(x) - meanx) / sdx
}
```

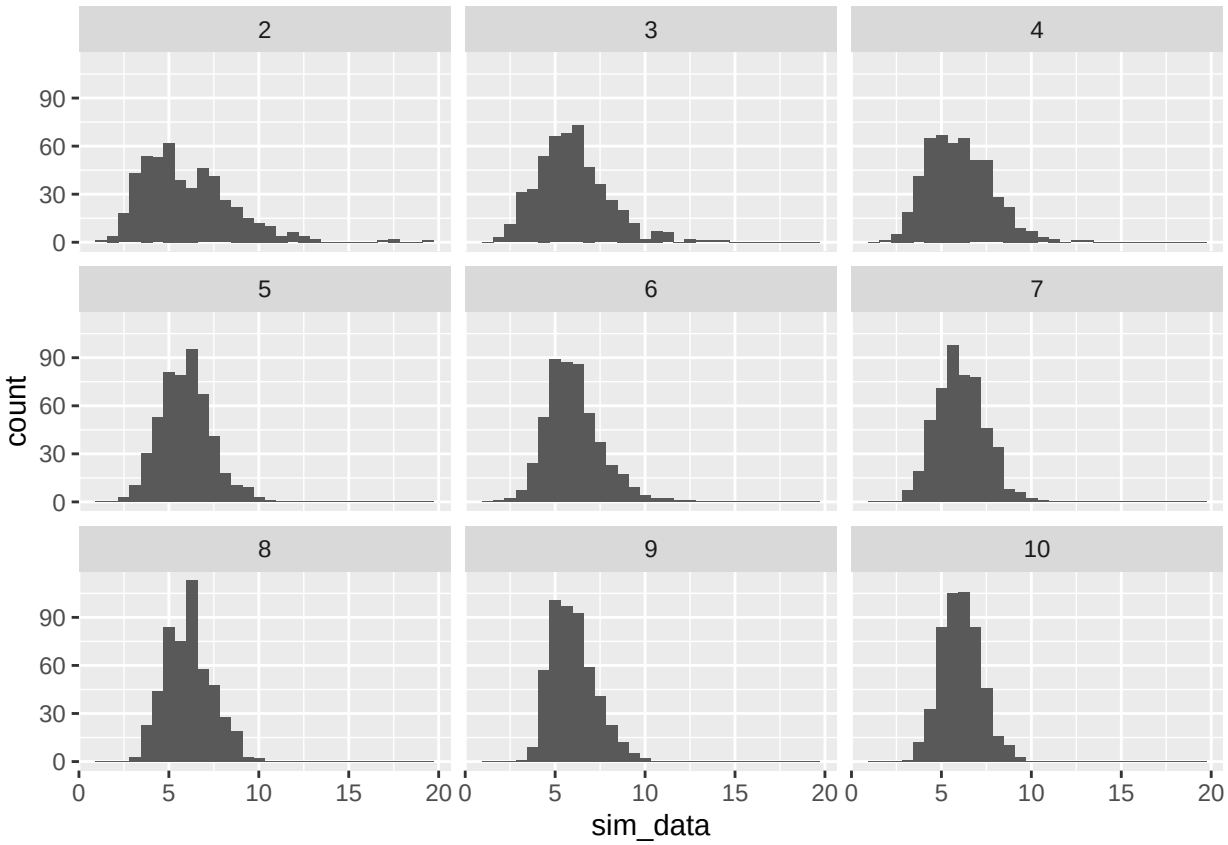
```
# normalised maximum of a vector
norm_max <- function(x = NULL,
                     shape = shape1,
                     rate = rate1) {
  nx <- length(x)
  p <- 1 - (1 / nx)
  meanx <- shape / rate
  sdx <- shape / (rate^2)
  # meanx = 5
  # sdx = 10
  b <- stats::qnorm(p, meanx, sdx)
  a <- 1 / (nx * stats::dnorm(b, meanx, sdx))
  # k = stats::quantile(x, p, type = 8, na.rm = TRUE)
  (max(x) - a) / b
  # max(x)/k
}
```

```
set.seed(12345)
data_all_mean <- create_nlevels(nlevels = seq(2, 50, 1), create_fun = mean)

set.seed(12345)
data_all_max <- create_nlevels(nlevels = seq(2, 50, 1),
                               create_fun = max)

set.seed(12345)
data_all_norm_mean <- create_nlevels(nlevels = seq(2, 50, 1),
                               create_fun = norm_mean)

set.seed(12345)
data_all_norm_max <- create_nlevels(nlevels = seq(2, 50, 1),
                               create_fun = norm_max)
```

Small levels and many iterations of means

```
data_all_mean %>%
  filter(ind %in% seq(2, 10, 1)) %>%
  ggplot(aes(x = sim_data)) +
  geom_histogram() +
  facet_wrap(~ind)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

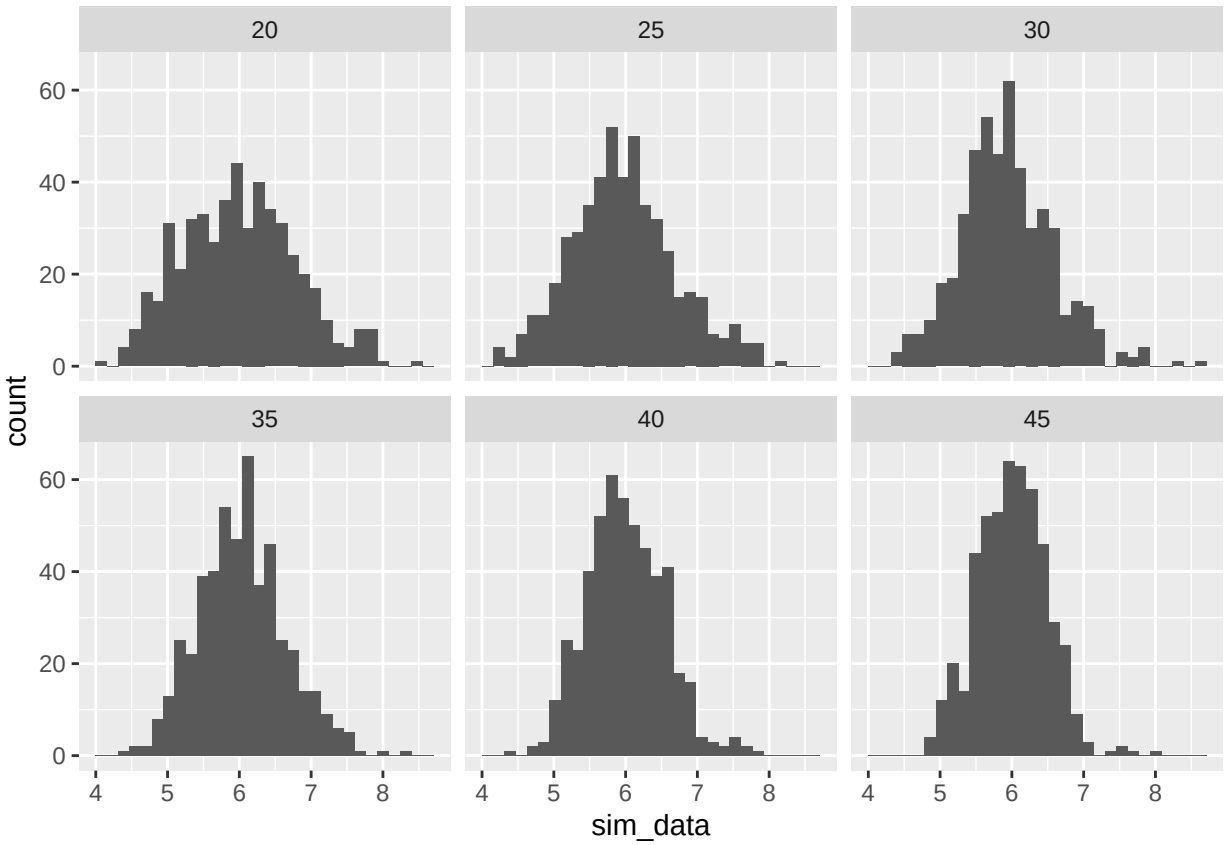


In facet with label 2, two random numbers are generated and their mean is computed. We repeat this process 500 times and draw the histogram of the computed means. Similarly, for facet with label 3, 3 random numbers are generated and histogram is drawn for the computed means of 3 random numbers.

Many levels and many iterations of means

```
data_all_mean %>%
  filter(ind %in% seq(20, 45, 5)) %>%
  ggplot(aes(x = sim_data)) +
  geom_histogram() +
  facet_wrap(~ind)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

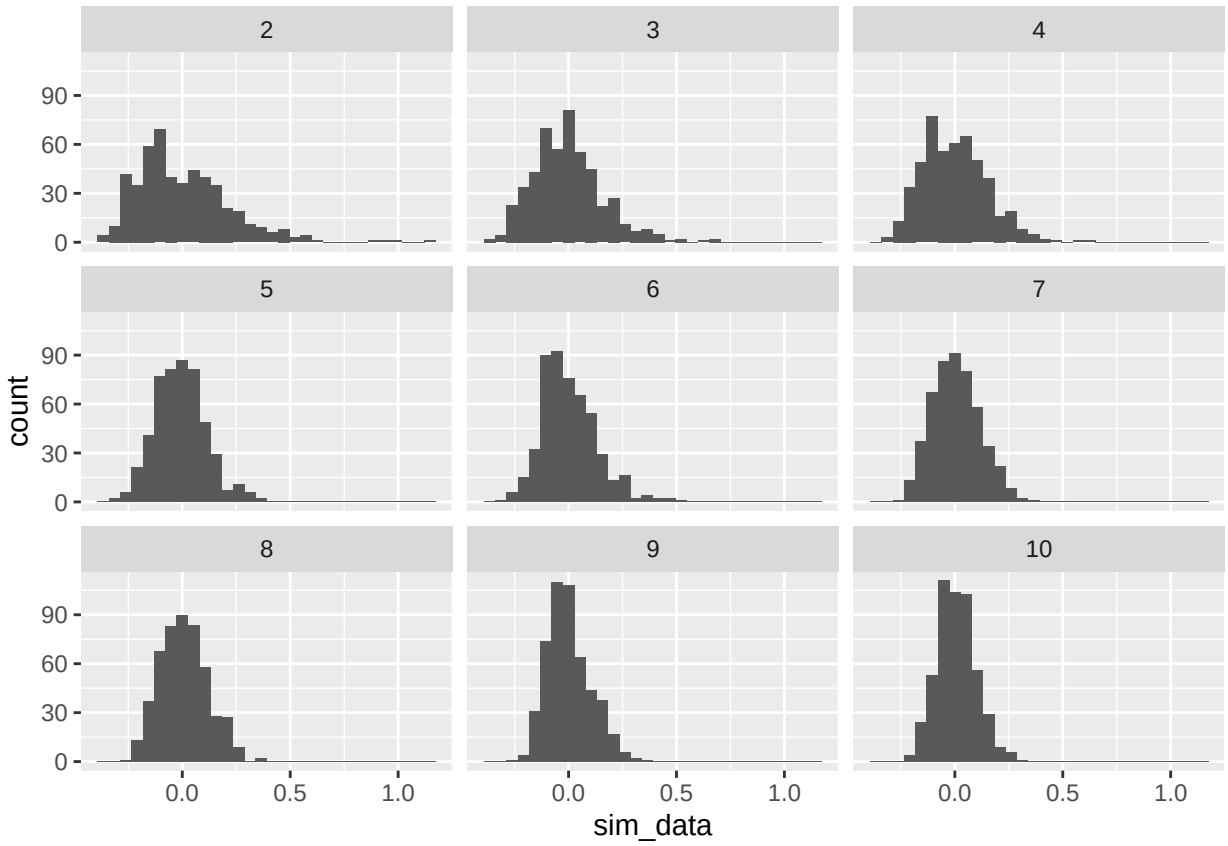


The histograms look like a normal distribution in this case, because number of levels are higher. For smaller number of levels, even with many repetitions the histograms do not look like normal.

Small levels and many iterations of normalised means

```
data_all_norm_mean %>%
  filter(ind %in% seq(2, 10, 1)) %>%
  ggplot(aes(x = sim_data)) +
  geom_histogram() +
  facet_wrap(~ind)
```

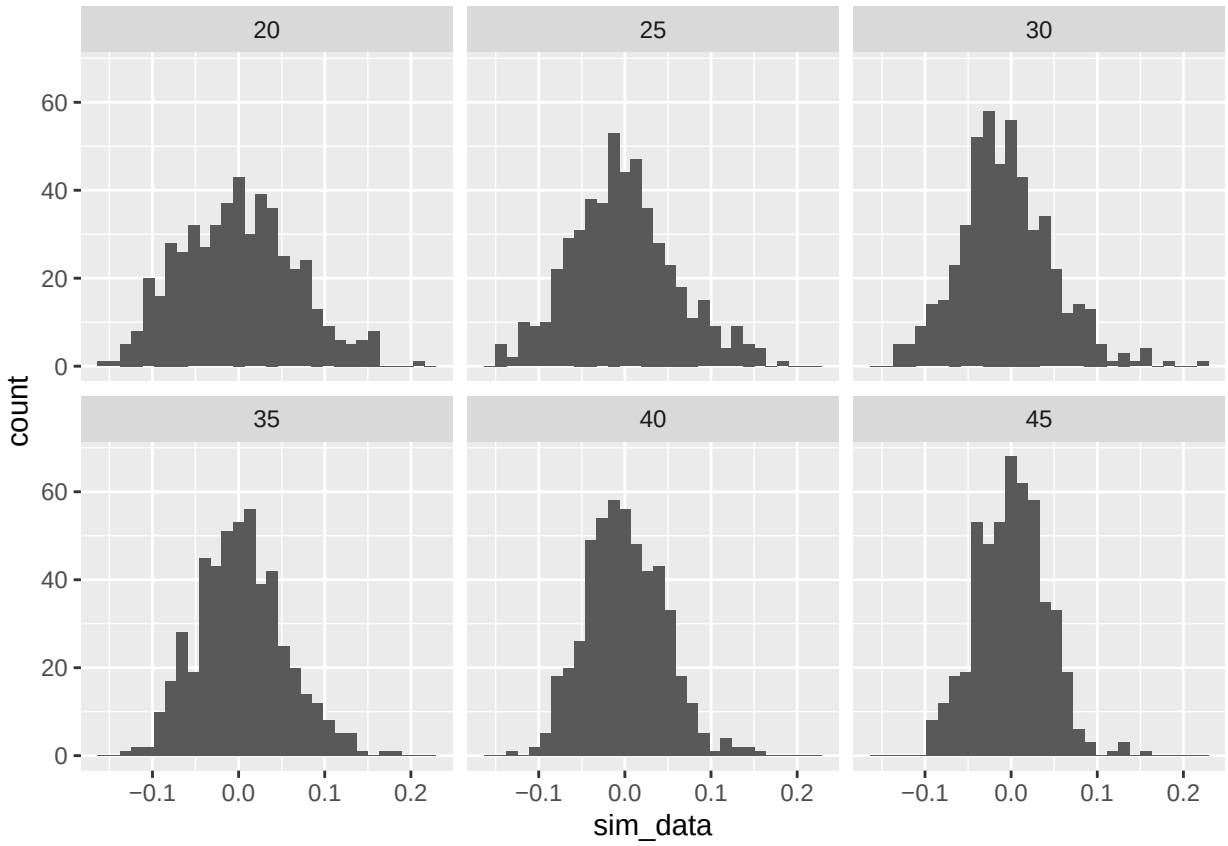
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Many levels and many iterations of normalised means

```
data_all_norm_mean %>%
  filter(ind %in% seq(20, 45, 5)) %>%
  ggplot(aes(x = sim_data)) +
  geom_histogram() +
  facet_wrap(~ind)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

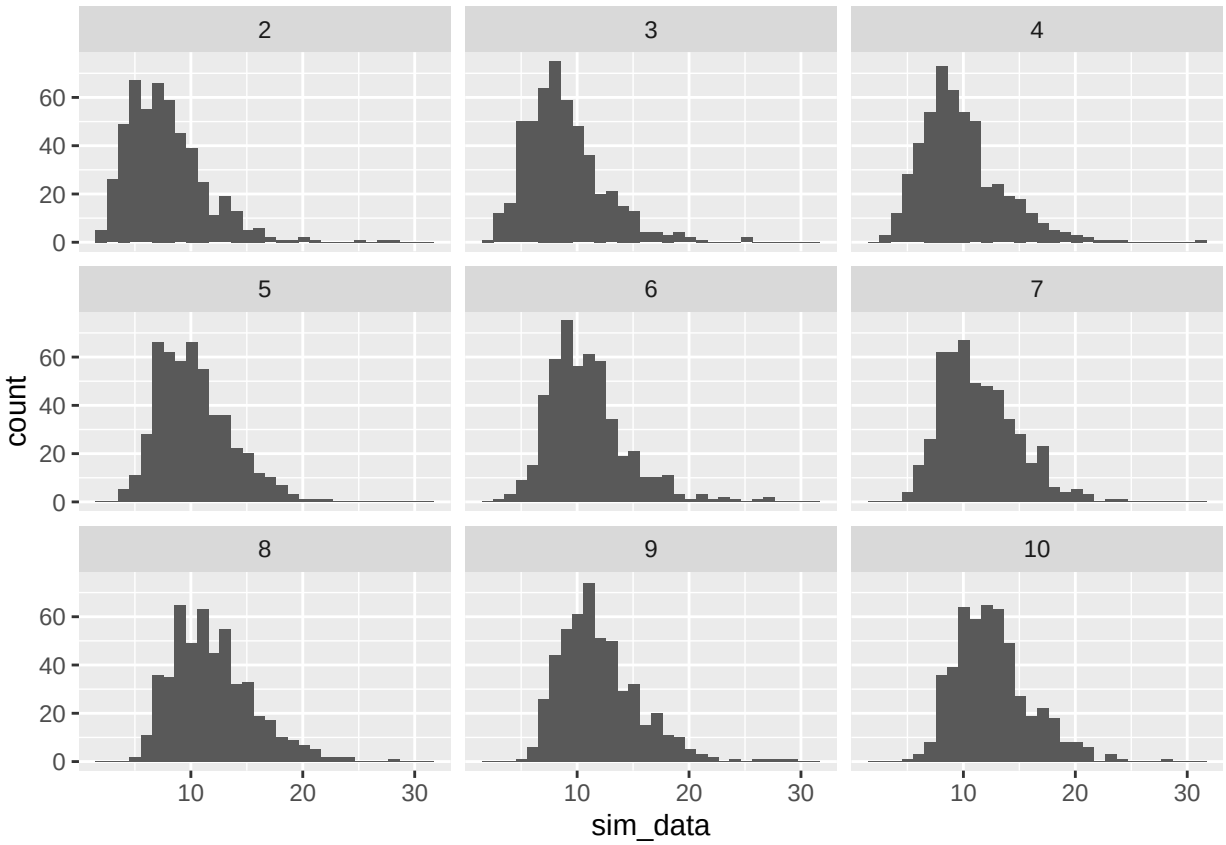


EVT

Small levels and many iterations of max

```
data_all_max %>%
  filter(ind %in% seq(2, 10, 1)) %>%
  ggplot(aes(x = sim_data)) +
  geom_histogram() +
  facet_wrap(~ind)
```

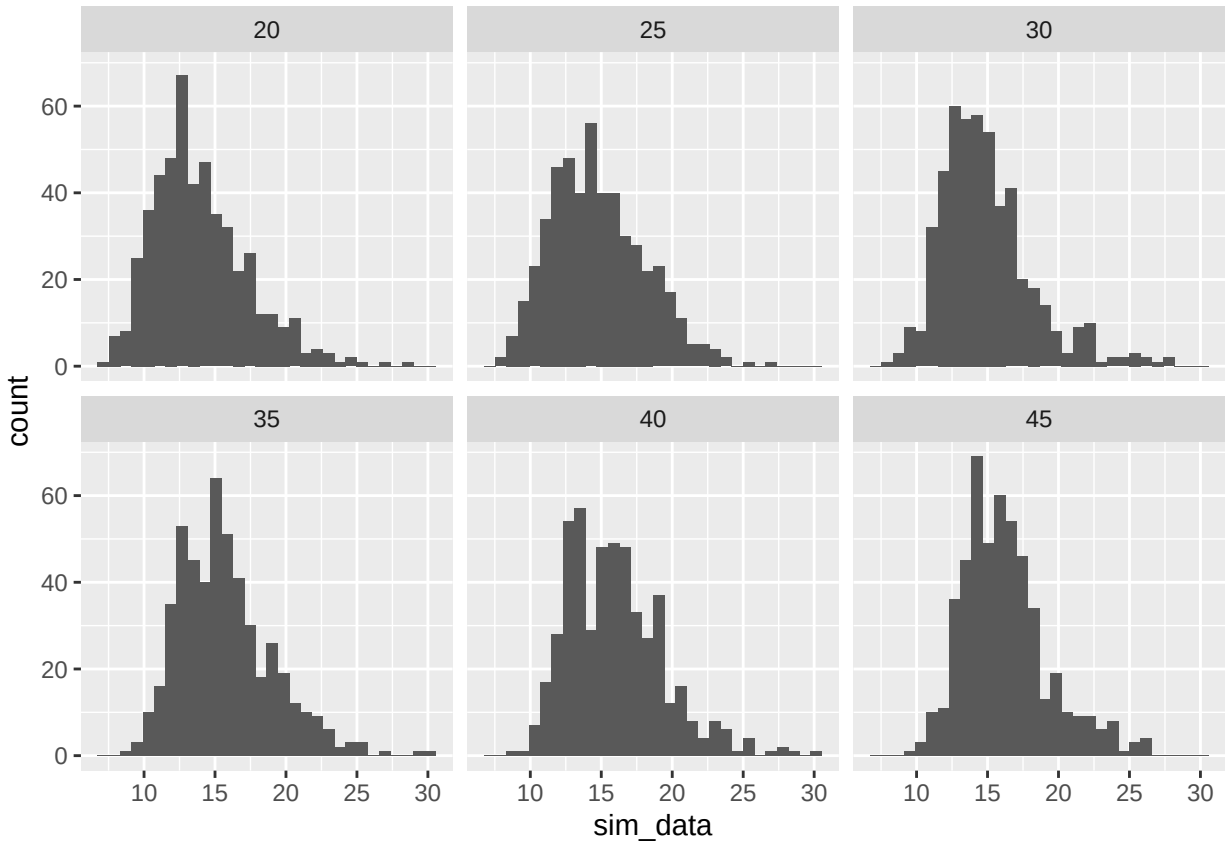
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Many levels and many iterations of max

```
data_all_max %>%
  filter(ind %in% seq(20, 45, 5)) %>%
  ggplot(aes(x = sim_data)) +
  geom_histogram() +
  facet_wrap(~ind)
```

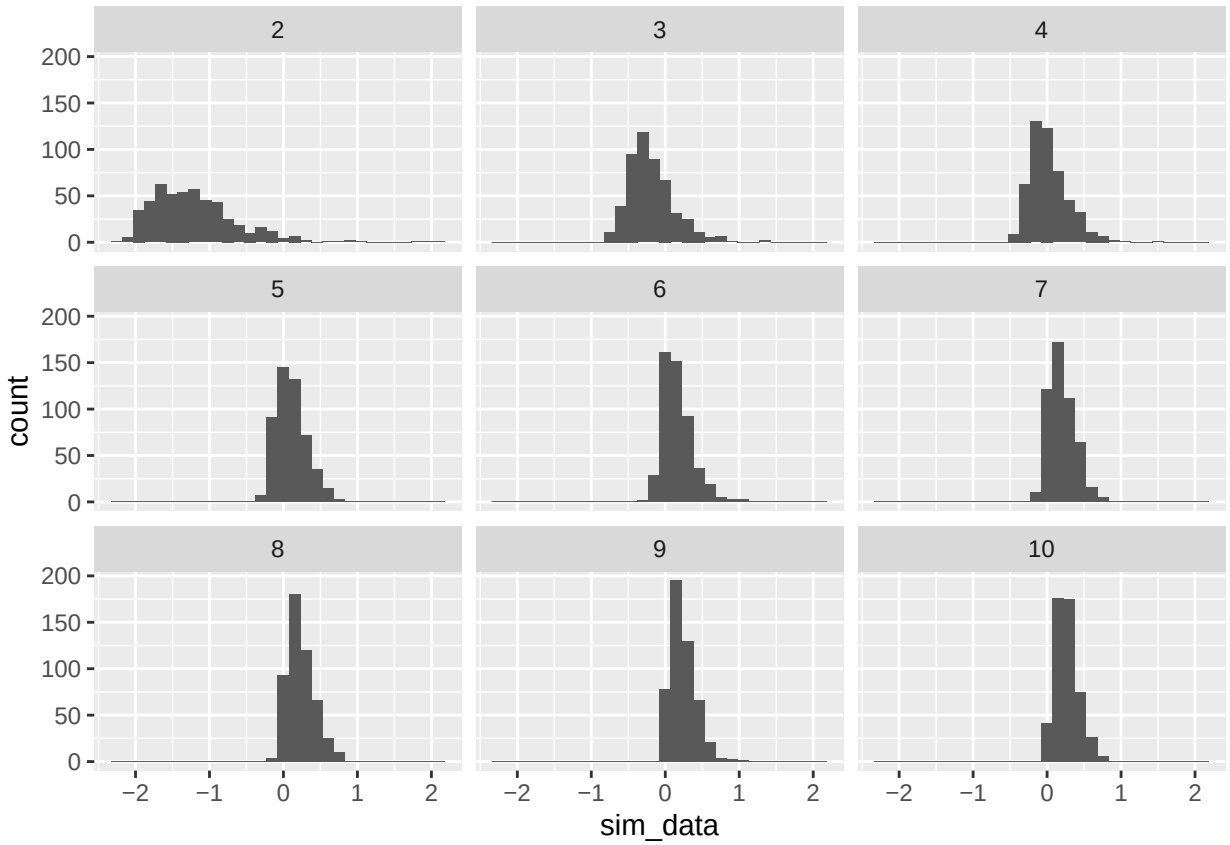
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Small levels and many iterations of norm max

```
data_all_norm_max %>%
  filter(ind %in% seq(2, 10, 1)) %>%
  ggplot(aes(x = sim_data)) +
  geom_histogram() +
  facet_wrap(~ind)
```

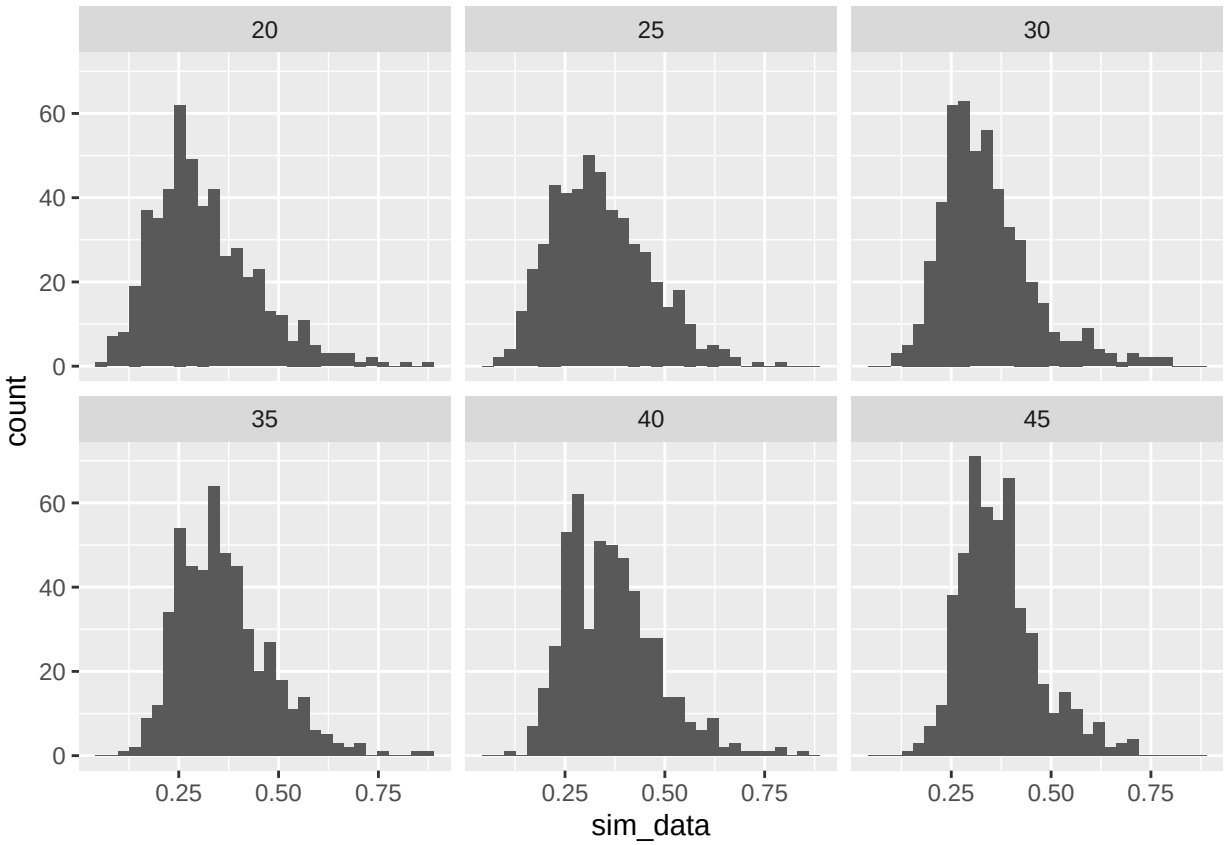
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Many levels and many iterations of norm max

```
data_all_norm_max %>%
  filter(ind %in% seq(20, 45, 5)) %>%
  ggplot(aes(x = sim_data)) +
  geom_histogram() +
  facet_wrap(~ind)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


20
25
30
35
40
45
count
sim_data
0
20
40
60
0.25
0.50
0.75