Compression Strategy Along Time Series

ScMaSigPro Supplementary Material-I

Priyansh Srivastava, … Ana Conesa

2023-08-16

Required Library

Define some custom functions

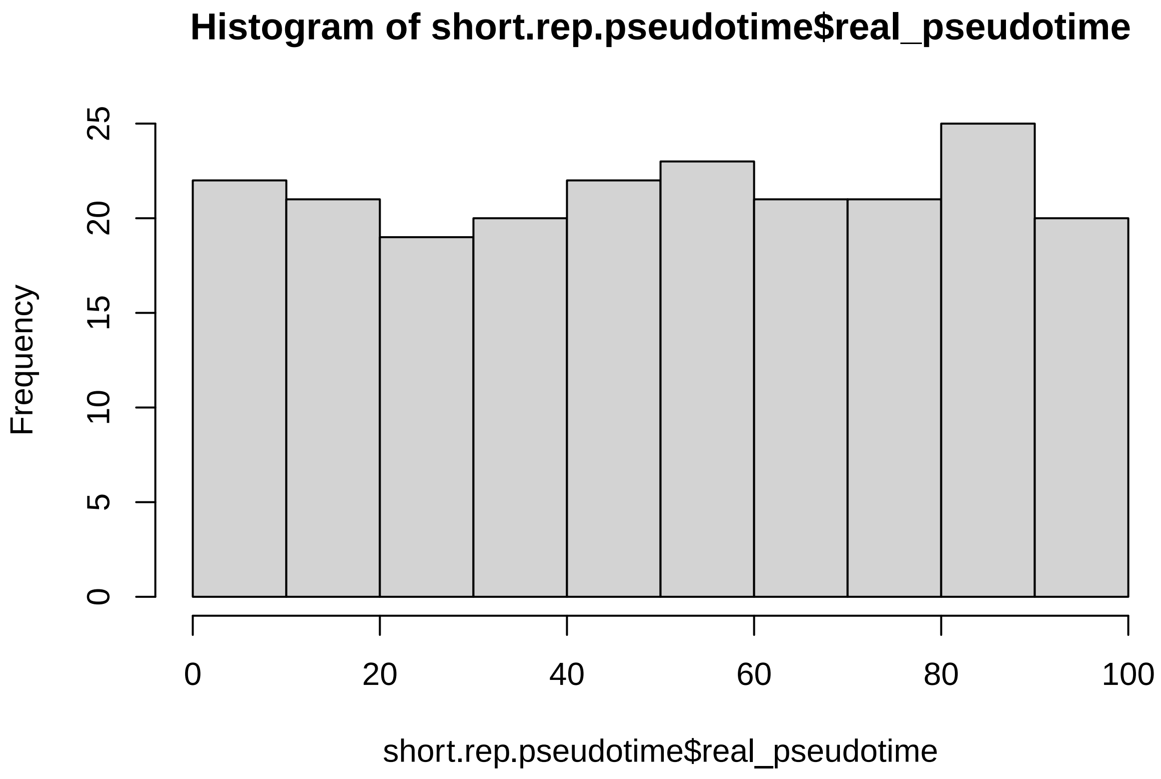
```r
# Define a function 'create_random_repeated_vector'
create_random_repeated_vector <- function(start, end, min_repetitions, max_repetitions) {
  repetitions <- sample(min_repetitions:max_repetitions,
                        size = end - start + 1, replace = TRUE)
  result_vector <- rep(seq(from = start, to = end, by = 1),
                       times = repetitions)
  return(result_vector)
}

# Define a function 'discretize'
discretize <- function(x, numBins, r = range(x)) {
  b <- seq(from = r[1], to = r[2], length.out = numBins + 1)
  cut_x <- cut(x, breaks = b, include.lowest = TRUE)
  y <- table(cut_x)
  return(y)
}

# Define a function 'create_range'
create_range <- function(x) {
  y <- as.character(x[["bin"]])
  y <- y %>% stringr::str_remove_all(pattern = "\\[|\\]|\\(|\\)")
  y1 <- as.numeric(sapply(strsplit(y, ","), "[", 1))
  y2 <- as.numeric(sapply(strsplit(y, ","), "[", 2))
  rangeVec <- c(y1, y2, x[["bin_size"]], x[["customTime"]])
  return(as.numeric(rangeVec))
}
```

Create Sample Data

```r
# Short Pseudotime with repetitions
short.rep.pseudotime <- data.frame(
  real_pseudotime = create_random_repeated_vector(1, 100, 1, 3)
)
hist(short.rep.pseudotime$real_pseudotime)
```



Sturges Binning

Step-0: Get the time-series

```
time_series <- short.rep.pseudotime$real_pseudotime
head(time_series)
```

```
## [1] 1 1 1 2 2 3
```

Step-1: Calculate the number of time-points

```
time_points <- length(time_series)
time_points
```

```
## [1] 214
```

Step-2: Take the log2 of the length

```
estBins <- log2(time_points) + 1
estBins
```

```
## [1] 8.741467
```

Step-3: Multiply with drop-factor to further reduce the bin-size

```
estBins <- estBins * 0.7 # drop.fac
estBins
```

```
## [1] 6.119027
```

Step-4: Calculate Bin intervals

```
bin_intervals <- as.data.frame(discretize(time_series,
  numBins = estBins,
  r = range(time_series)
))
kable(bin_intervals)
```

cut_x	Freq
[1,15.1]	30
(15.1,29.3]	31
(29.3,43.4]	27
(43.4,57.6]	32
(57.6,71.7]	30
(71.7,85.9]	30
(85.9,100]	34

Step-5: Clean the table before merge

```
colnames(bin_intervals) <- c("bin", "bin_size")
bin_intervals$customTime <- rownames(bin_intervals)
kable(bin_intervals)
```

bin	bin_size	customTime
[1,15.1]	30	1
(15.1,29.3]	31	2
(29.3,43.4]	27	3
(43.4,57.6]	32	4
(57.6,71.7]	30	5
(71.7,85.9]	30	6
(85.9,100]	34	7

Step-6: Create the bin table

```
bin_table <- as.data.frame(t(as.data.frame(apply(bin_intervals, 1, create_range))))
colnames(bin_table) <- c("from", "to", "bin_size", "binnedTime")
kable(bin_table)
```

	from	to	bin_size	binnedTime
V1	1.0	15.1	30	1
V2	15.1	29.3	31	2
V3	29.3	43.4	27	3
V4	43.4	57.6	32	4
V5	57.6	71.7	30	5
V6	71.7	85.9	30	6
V7	85.9	100.0	34	7

Step-7: Merge with Original time-series

```
short.rep.pseudotime.pooled <- as.data.frame(
  left_join(
    short.rep.pseudotime, bin_table,
    by = join_by(
      closest(real_pseudotime >= from),
      closest(real_pseudotime <= to)
    )
  )
)
kable(short.rep.pseudotime.pooled[c(c(1:3), c(29:33), c(55:58)),])
```

	real_pseudotime	from	to	bin_size	binnedTime
1	1	1.0	15.1	30	1
2	1	1.0	15.1	30	1
3	1	1.0	15.1	30	1
29	14	1.0	15.1	30	1
30	15	1.0	15.1	30	1
31	16	15.1	29.3	31	2
32	16	15.1	29.3	31	2
33	16	15.1	29.3	31	2
55	25	15.1	29.3	31	2
56	25	15.1	29.3	31	2
57	26	15.1	29.3	31	2
58	26	15.1	29.3	31	2