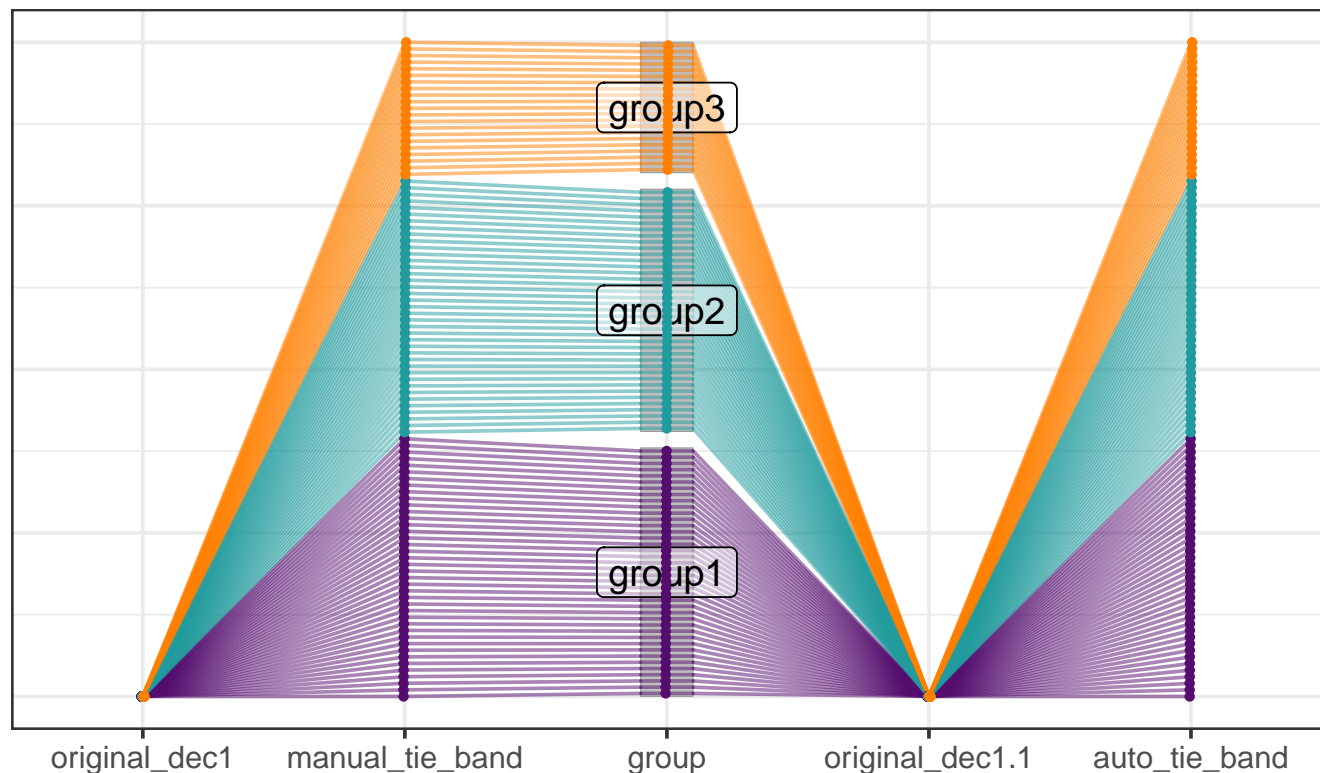


Simulating the bins: Analysis of Histogram Layout Differences Across Binwidths

Point Mass Distribution:

```
simulate_point_mass <- function(n, value) {  
  rep(value, n)  
}  
  
# Simulate data  
set.seed(42) # Set seed for reproducibility  
n <- 100  
  
point_mass_data <- simulate_point_mass(n, round(n/12, 0))  
  
create_simulated_data_test(point_mass_data, "Point Mass")
```

Parallel Coordinate Plot: Original vs Adjusted Data (Simulated Point



Exponential Distribution:

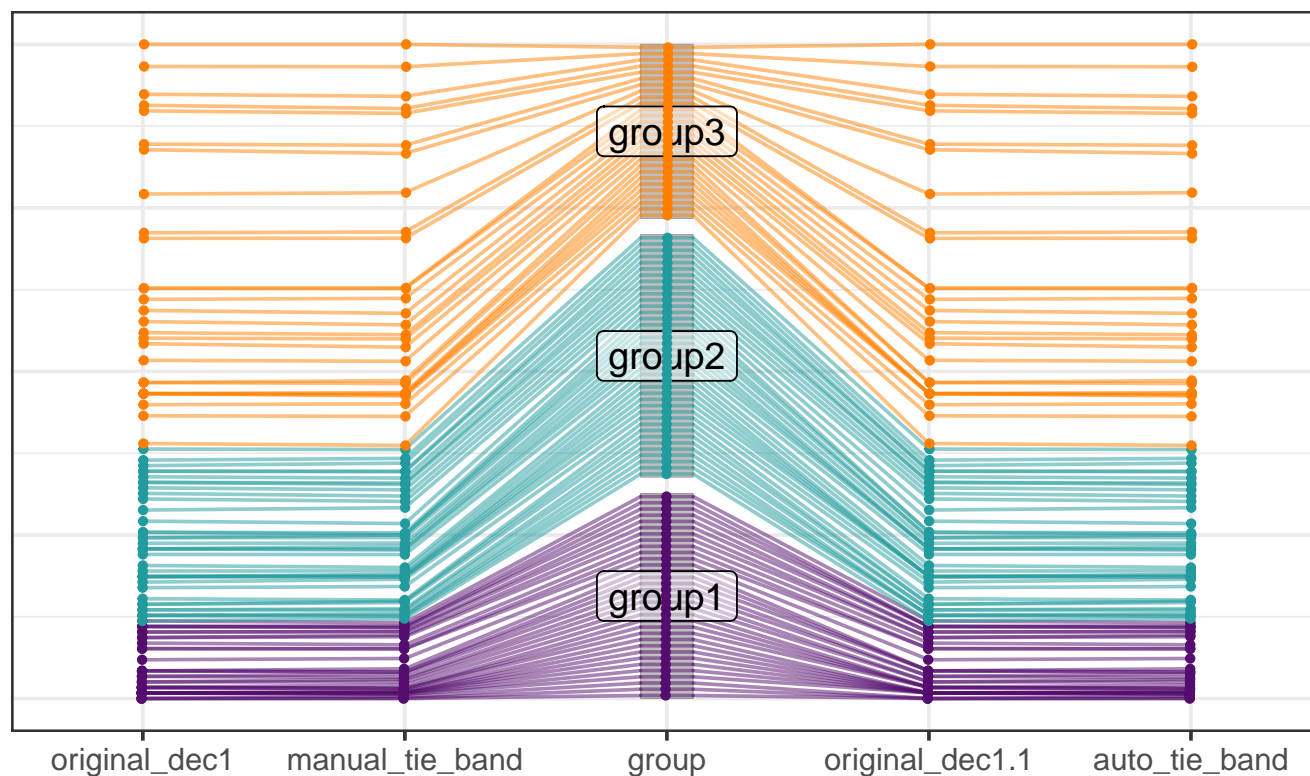
- Simulated data follows an exponential decay pattern.

```
# Function to simulate exponential distribution
simulate_exponential <- function(n, rate) {
  rexp(n, rate = rate)
}

exponential_data <- simulate_exponential(n, rate = .33)

create_simulated_data_test(exponential_data, "Exponential")
```

Parallel Coordinate Plot: Original vs Adjusted Data (Simulated Expo



Normal Distribution:

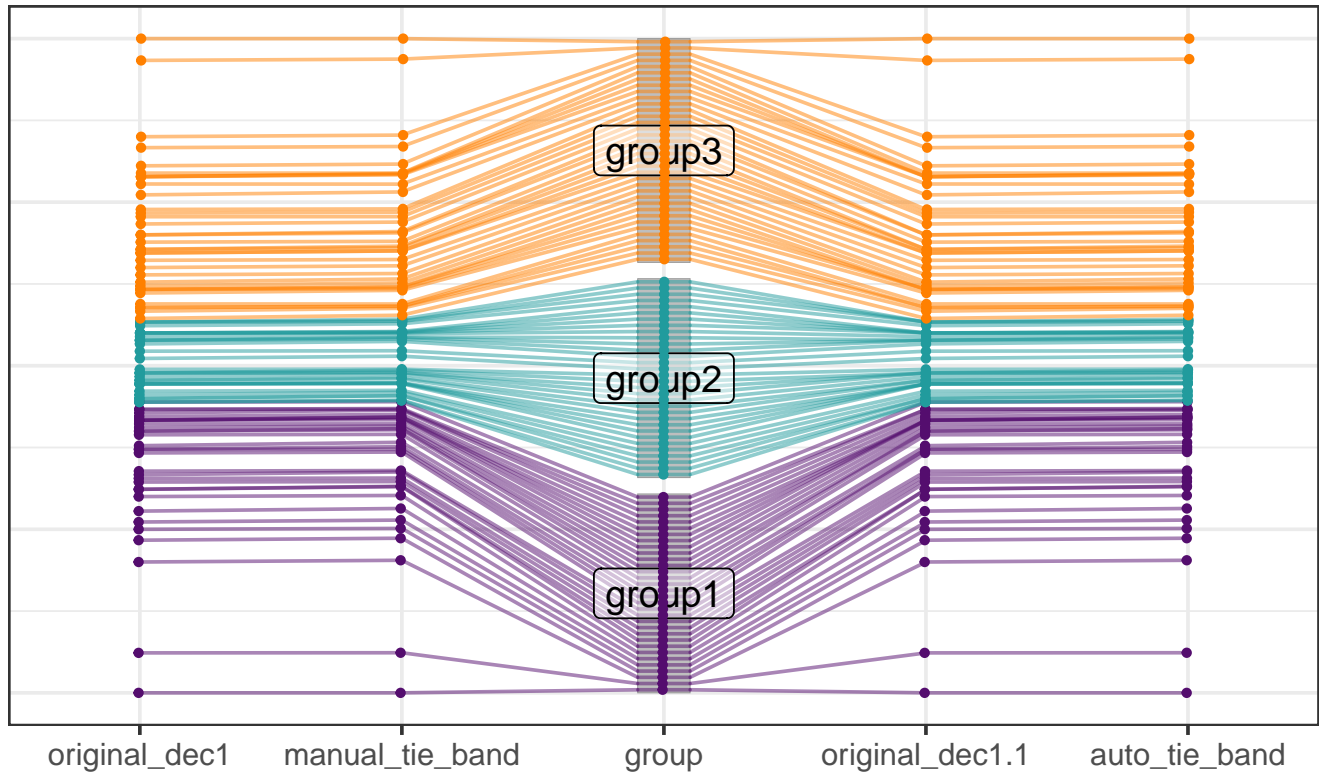
- Simulated data is centered around a mean (10) with a specified standard deviation (3).

```
# Function to simulate normal distribution
simulate_normal <- function(n, mean, sd) {
  rnorm(n, mean = mean, sd = sd)
}

normal_data <- simulate_normal(n, mean = 10, sd = 3)

create_simulated_data_test(normal_data, "Normal")
```

Parallel Coordinate Plot: Original vs Adjusted Data (Simulated Norr



Poisson Distribution:

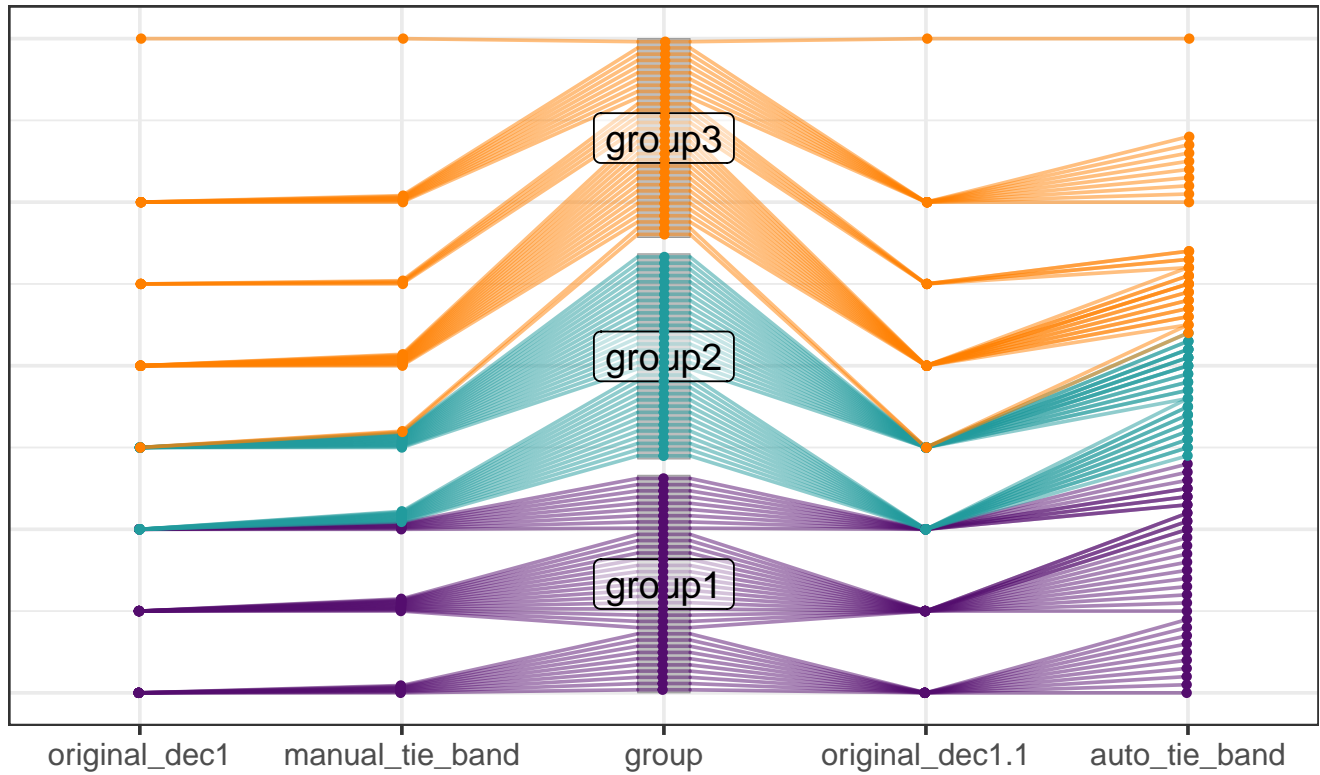
- Simulated count data with a mean occurrence ($\lambda = 3$).

```
# Function to simulate Poisson distribution
simulate_poisson <- function(n, lambda) {
  rpois(n, lambda = lambda)
}

poisson_data <- simulate_poisson(n, lambda = 3)

create_simulated_data_test(poisson_data, "Poisson")
```

Parallel Coordinate Plot: Original vs Adjusted Data (Simulated Poiss



Uniform Distribution:

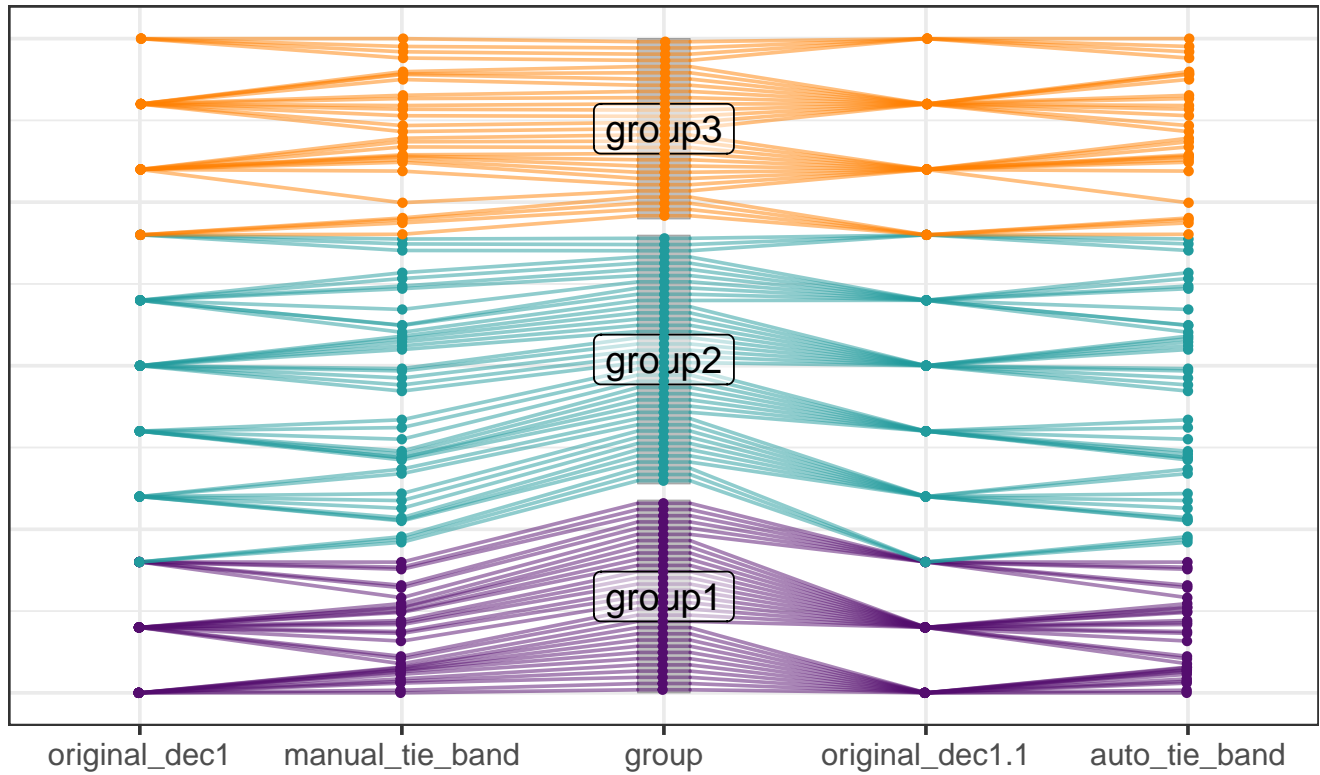
- Simulated data is evenly distributed between a minimum (0) and maximum (1).

```
# Function to simulate Uniform distribution
simulate_uniform <- function(n, min, max) {
  runif(n, min = min, max = max)
}

uniform_data <- simulate_uniform(n, min = 0, max = 1)

create_simulated_data_test(uniform_data, "Uniform")
```

Parallel Coordinate Plot: Original vs Adjusted Data (Simulated Unifo



Binomial Distribution:

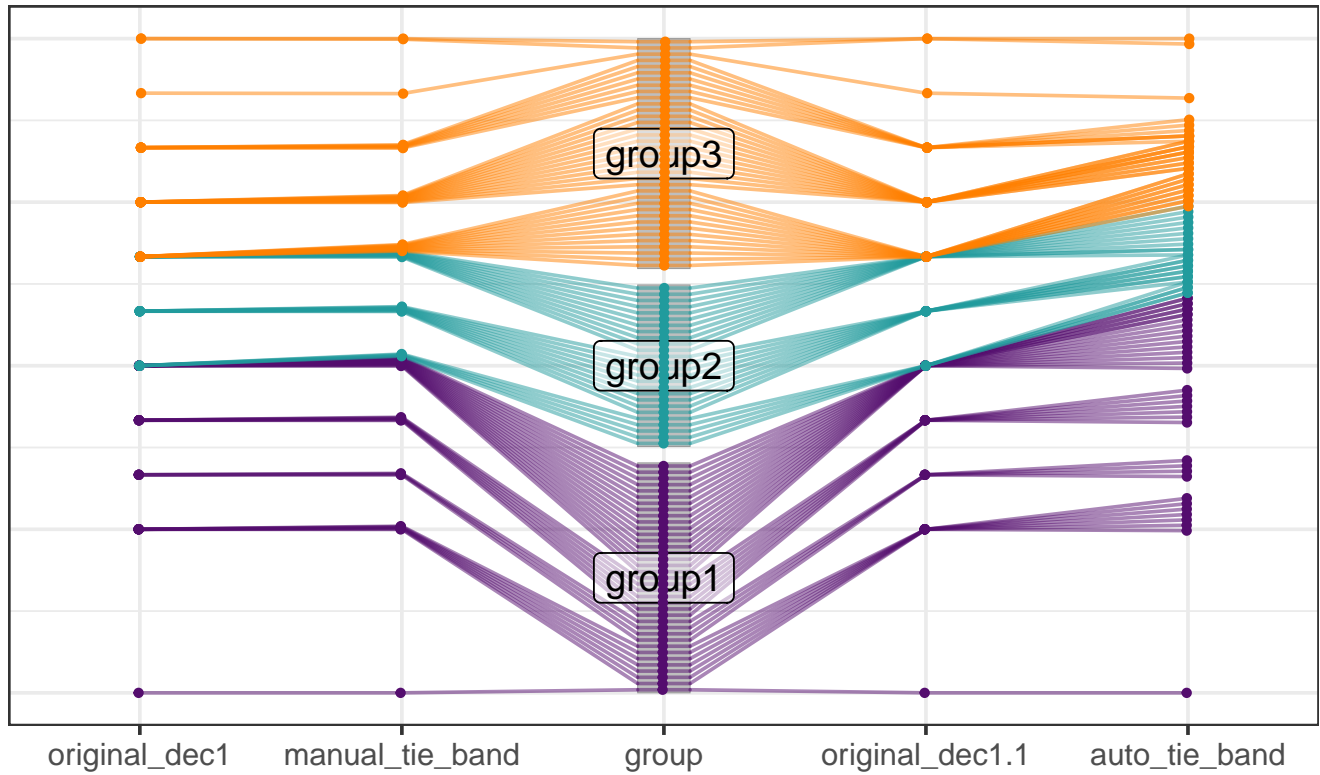
- Models the number of successes in a fixed number of trials ($\text{size} = 10$, $\text{prob} = 0.5$).

```
# Function to simulate binomial distribution
simulate_binomial <- function(n, size, prob) {
  rbinom(n, size = size, prob = prob)
}

binomial_data <- simulate_binomial(n, size = 20, prob = 0.5)

create_simulated_data_test(binomial_data, "Binomial")
```

Parallel Coordinate Plot: Original vs Adjusted Data (Simulated Binor



Beta Distribution:

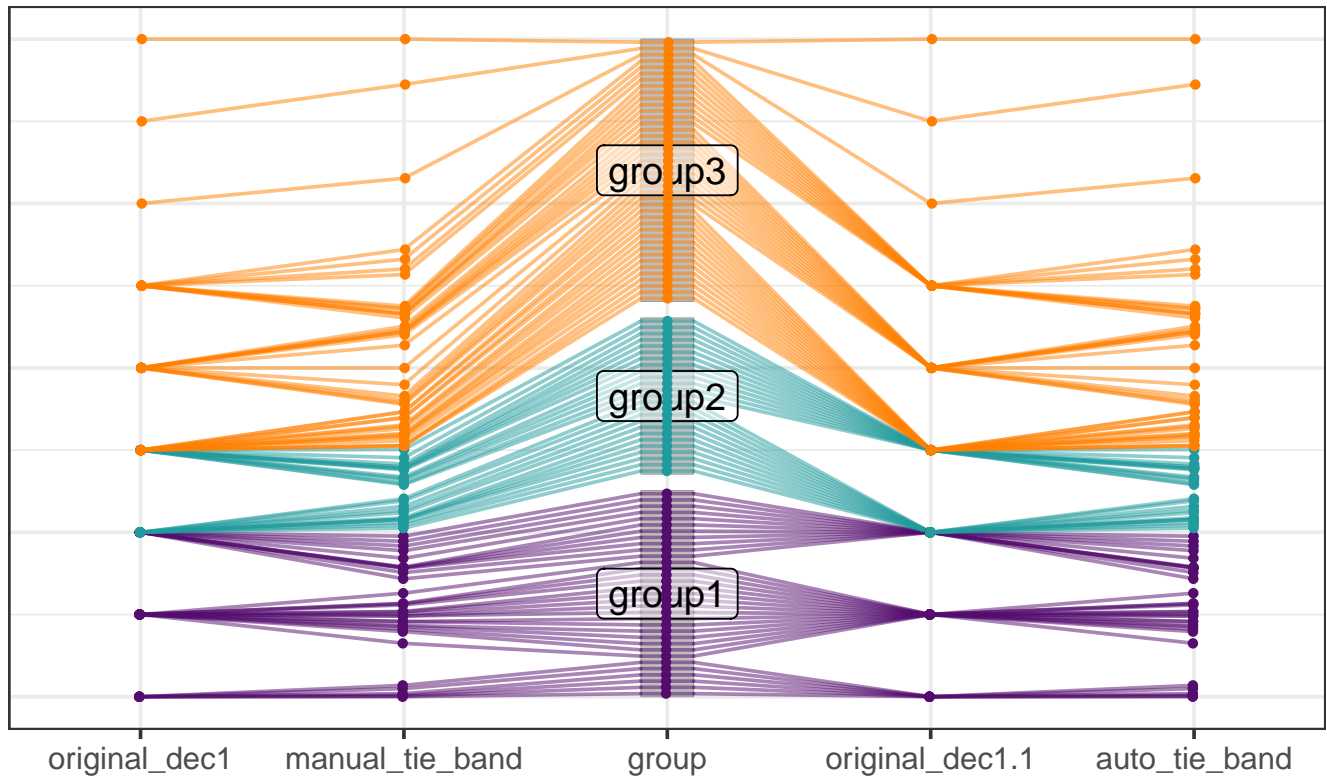
- Continuous data bounded between 0 and 1, modeled with shape parameters (shape1 = 2, shape2 = 5).

```
# Function to simulate beta distribution
simulate_beta <- function(n, shape1, shape2) {
  rbeta(n, shape1 = shape1, shape2 = shape2)
}

beta_data <- simulate_beta(n, shape1 = 2, shape2 = 5)

create_simulated_data_test(beta_data, "Beta")
```

Parallel Coordinate Plot: Original vs Adjusted Data (Simulated Beta



Gamma Distribution:

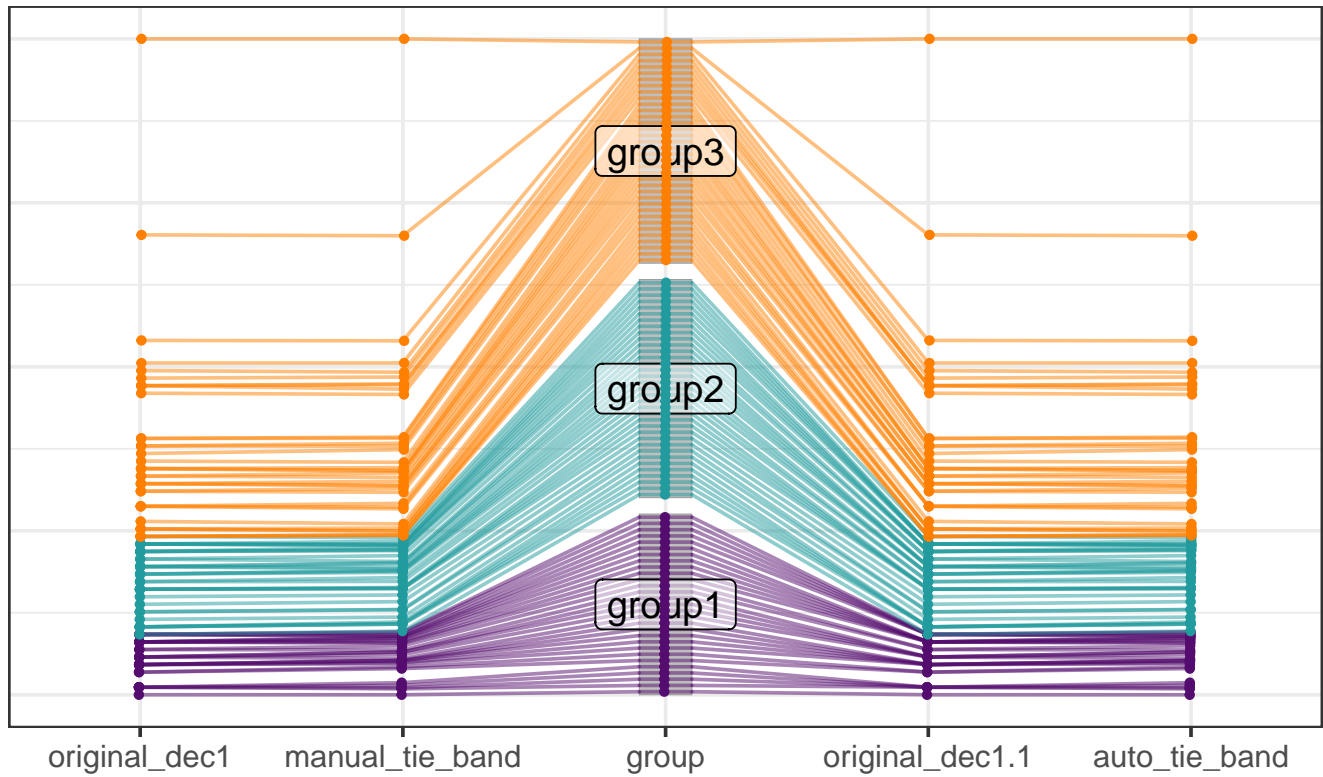
- Models waiting times with shape and rate parameters (shape = 2, rate = 1).

```
# Function to simulate gamma distribution
simulate_gamma <- function(n, shape, rate) {
  rgamma(n, shape = shape, rate = rate)
}

gamma_data <- simulate_gamma(n, shape = 2, rate = 1)

create_simulated_data_test(gamma_data, "Gamma")
```


Parallel Coordinate Plot: Original vs Adjusted Data (Simulated Gam



Chi-Square Distribution:

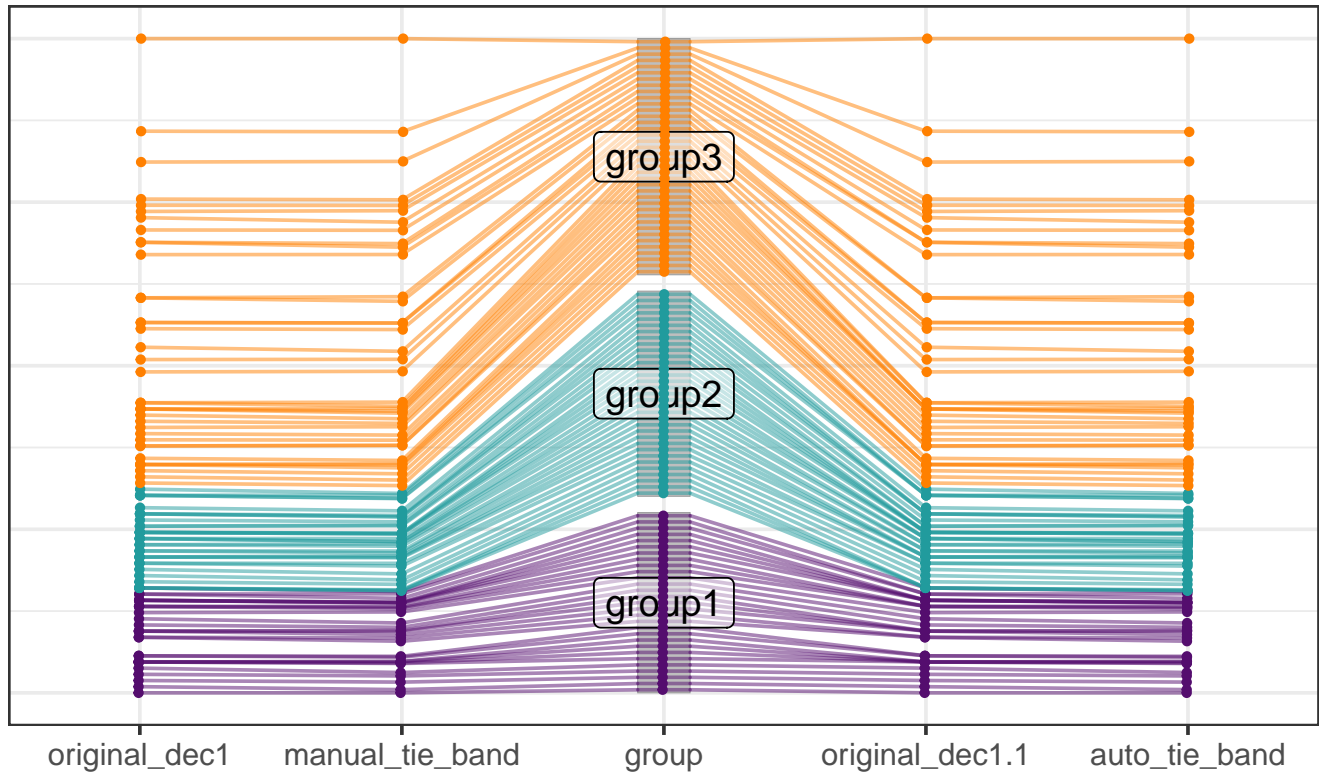
- Data follows a chi-square distribution with degrees of freedom ($df = 3$).

```
# Function to simulate chi-square distribution
simulate_chisquare <- function(n, df) {
  rchisq(n, df = df)
}

chisquare_data <- simulate_chisquare(n, df = 3)

create_simulated_data_test(chisquare_data, "Chi-Square")
```


Parallel Coordinate Plot: Original vs Adjusted Data (Simulated Chi–)



Negative Binomial Distribution:

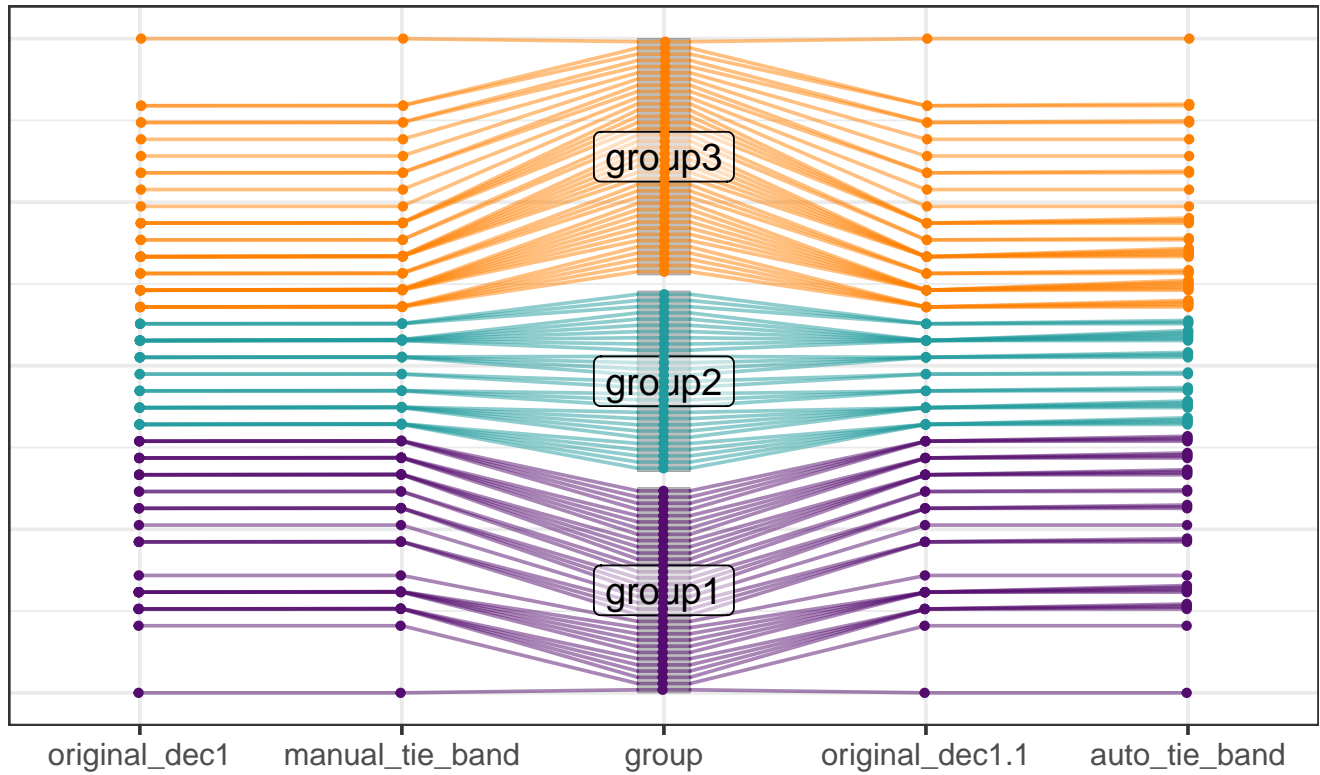
- Models the number of failures before achieving a set number of successes.

```
# Function to simulate negative binomial distribution
simulate_negative_binomial <- function(n, size, prob) {
  rnbinom(n, size = size, prob = prob)
}

negative_binomial_data <- simulate_negative_binomial(n, size = 10, prob = 0.3)

create_simulated_data_test(negative_binomial_data, "Negative Binomial")
```

Parallel Coordinate Plot: Original vs Adjusted Data (Simulated Negative)



Log-Normal Distribution:

- Data is distributed such that the logarithm of the values follows a normal distribution ($\text{meanlog} = 0$, $\text{sdlog} = 1$).

```
# Function to simulate log-normal distribution
simulate_lognormal <- function(n, meanlog, sdlog) {
  rlnorm(n, meanlog = meanlog, sdlog = sdlog)
}

lognormal_data <- simulate_lognormal(n, meanlog = 0, sdlog = 1)

create_simulated_data_test(lognormal_data, "Log-normal")
```

Parallel Coordinate Plot: Original vs Adjusted Data (Simulated Log-

