

# Comparison between the new method and the old method

2024-04-24

## Load Dataset

```
##this file tests the proteomics example
library("IHWpaper")
```

```
## Loading required package: IHW
```

```
##
```

```
## Attaching package: 'IHW'
```

```
## The following object is masked _by_ '.GlobalEnv':
```

```
##
```

```
##      ihw
```

```
library("IHW")
```

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'stringr' was built under R version 4.2.3
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
## v dplyr      1.1.4      v readr      2.1.5
```

```
## v forcats   1.0.0      v stringr   1.5.1
```

```
## v ggplot2   3.5.0      v tibble    3.2.1
```

```
## v lubridate 1.9.3      v tidyr     1.3.1
```

```
## v purrr     1.0.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x ggplot2::alpha() masks IHW::alpha()
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::id() masks CVXR::id()
```

```
## x purrr::is_vector() masks CVXR::is_vector()
```

```
## x dplyr::lag() masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
proteomics_file <- system.file("extdata/real_data",
                                "science_signaling.csv", package = "IHWpaper")
```

```
proteomics_df <- read.csv(proteomics_file, stringsAsFactors = F)
```

```
proteomics_df$pvalue <- rank(proteomics_df$p1, ties.method="first")*proteomics_df$p1/nrow(proteomics_df)
```

## Run the three methods

```
num_bins <- seq(from = 10, to = 50, by = 10)
N <- length(num_bins)

time.new <- rep(NA, N)
time.old <- rep(NA, N)
time.cvxr <- rep(NA, N)

rej_new <- rep(NA, N)
rej_old <- rep(NA, N)
rej_cvxr <- rep(NA, N)

for(i in 1:N){
  time.new[i] <- system.time(ihw_res_new <- ihw(proteomics_df$pvalue, proteomics_df$X..peptides, alpha = 0.05))
  rej_new[i] <- IHW::rejections(ihw_res_new)

  time.old[i] <- system.time(ihw_res_old <- IHWold::ihwOld(proteomics_df$pvalue, proteomics_df$X..peptides, alpha = 0.05))
  rej_old[i] <- IHW::rejections(ihw_res_old)

  print(paste0("Ending iteration ", i))
}

## We recommend that you supply (many) more than 1000 p-values for meaningful data-driven hypothesis weighting
## [1] "Ending iteration 1"
## We recommend that you supply (many) more than 200 p-values for meaningful data-driven hypothesis weighting
## We recommend that you supply (many) more than 1000 p-values for meaningful data-driven hypothesis weighting
## [1] "Ending iteration 2"
## We recommend that you supply (many) more than 200 p-values for meaningful data-driven hypothesis weighting
## We recommend that you supply (many) more than 1000 p-values for meaningful data-driven hypothesis weighting
## [1] "Ending iteration 3"
## We recommend that you supply (many) more than 200 p-values for meaningful data-driven hypothesis weighting
## We recommend that you supply (many) more than 1000 p-values for meaningful data-driven hypothesis weighting
## [1] "Ending iteration 4"
## We recommend that you supply (many) more than 200 p-values for meaningful data-driven hypothesis weighting
## We recommend that you supply (many) more than 1000 p-values for meaningful data-driven hypothesis weighting
## [1] "Ending iteration 5"
```

## Plot the results

```
load("~/Library/Mobile Documents/com~apple~CloudDocs/Desktop/UchicagoStudy/Nikolaos Research/IHW_New/CVXR")
load("~/Library/Mobile Documents/com~apple~CloudDocs/Desktop/UchicagoStudy/Nikolaos Research/IHW_New/CVXR")
load("~/Library/Mobile Documents/com~apple~CloudDocs/Desktop/UchicagoStudy/Nikolaos Research/IHW_New/CVXR")

data <- data.frame(num_bins = num_bins, rejection_new = rej_new[1:5], rejection_admm = rej_admm[1:5], rejection_cvxr = rej_cvxr[1:5])

#we first plot runtime versus bins
# Create the line plot
```

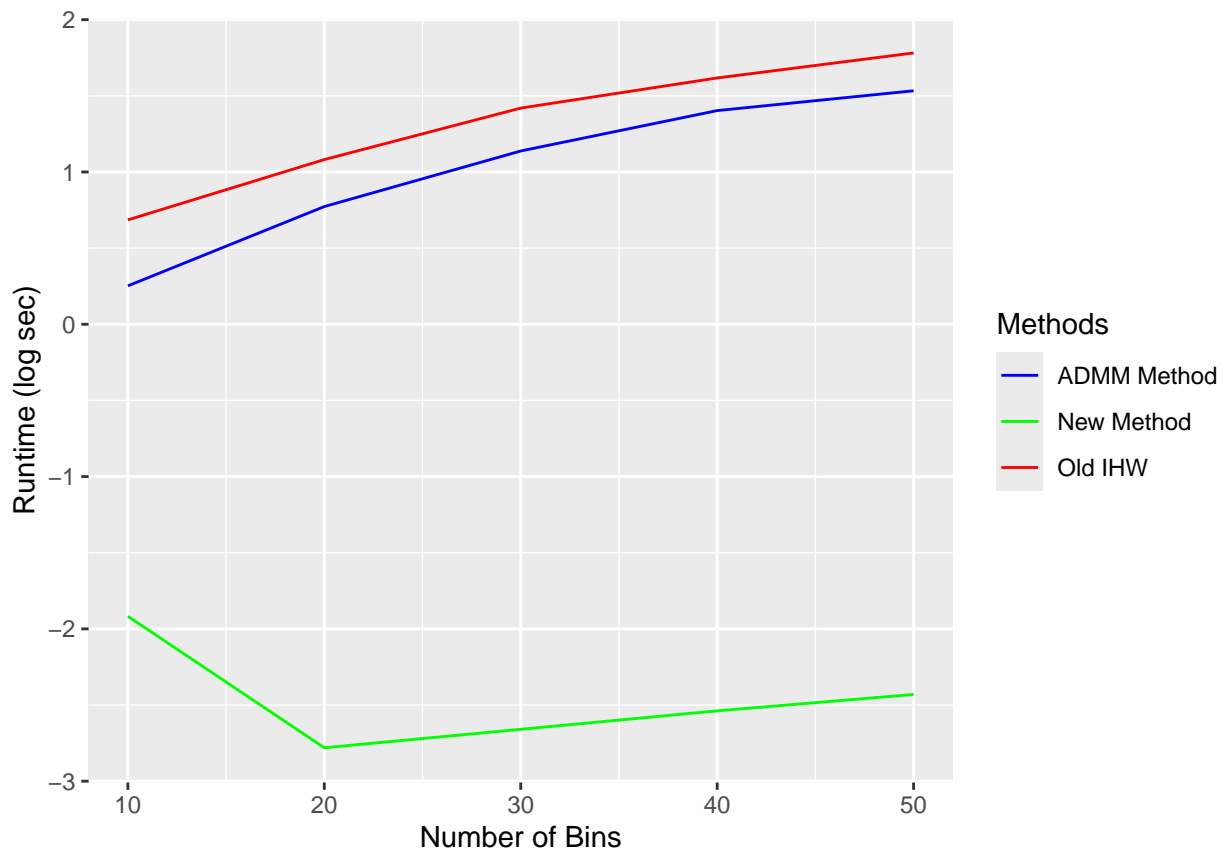
```
p1 <- ggplot(data) +
  geom_line(aes(x = num_bins, y = log(time_admm), col = "ADMM Method", linetype = "ADMM Method")) +
  geom_line(aes(x = num_bins, y = log(time_old), col = "Old IHW", linetype = "Old IHW")) +
  geom_line(aes(x = num_bins, y = log(time_new), col = "New Method", linetype = "New Method")) +
  labs(x = "Number of Bins", y = "Runtime (log sec)") +
  scale_color_manual(values = c("ADMM Method" = "blue", "Old IHW" = "red", "New Method" = "green"), name = "Methods") +
  scale_linetype_manual(values = c("ADMM Method" = "solid", "Old IHW" = "solid", "New Method" = "solid"))
```

*#we then plot rejections versus bins*

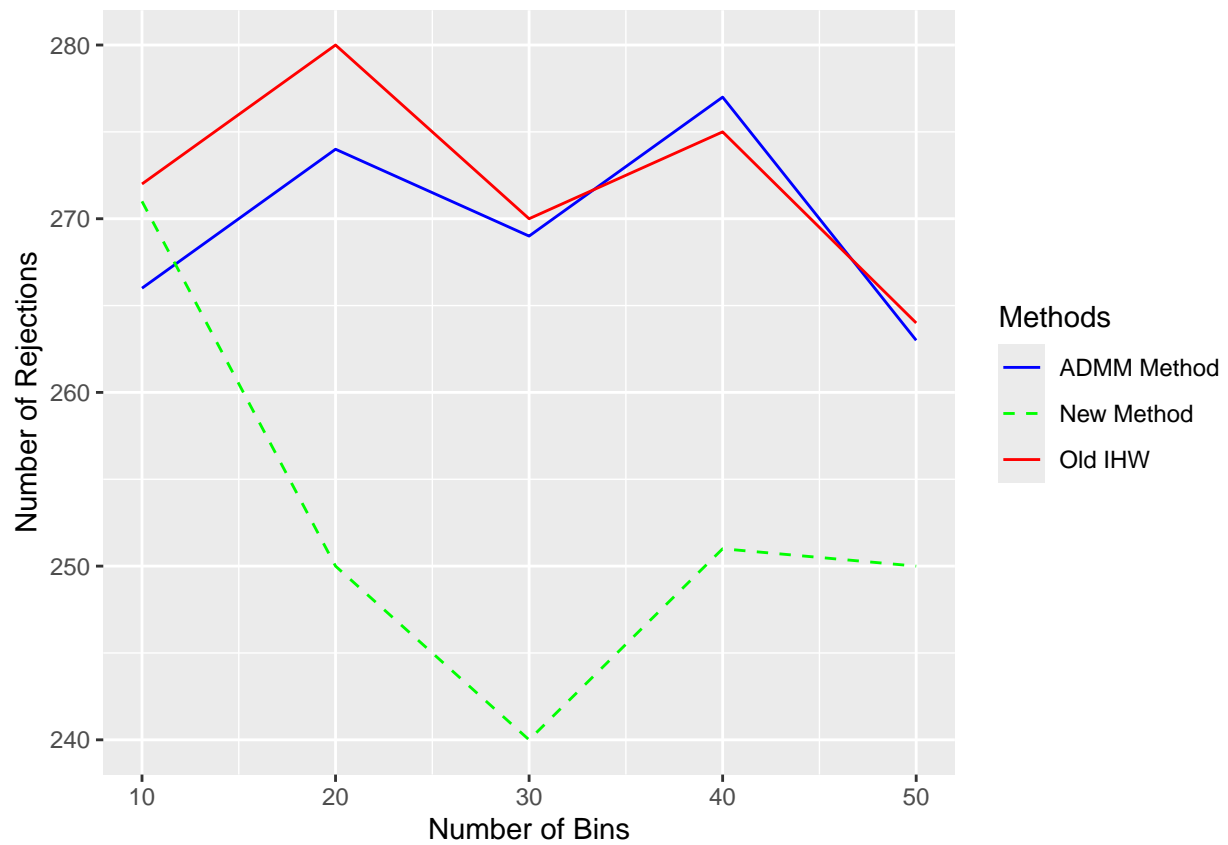
*# Create the line plot*

```
p2 <- ggplot(data) +
  geom_line(aes(x = num_bins, y = rejection_admm, col = "ADMM Method", linetype = "ADMM Method")) +
  geom_line(aes(x = num_bins, y = rejection_old, col = "Old IHW", linetype = "Old IHW")) +
  geom_line(aes(x = num_bins, y = rejection_new, col = "New Method", linetype = "New Method")) +
  labs(x = "Number of Bins", y = "Number of Rejections") +
  scale_color_manual(values = c("ADMM Method" = "blue", "Old IHW" = "red", "New Method" = "green"), name = "Methods") +
  scale_linetype_manual(values = c("ADMM Method" = "solid", "Old IHW" = "solid", "New Method" = "dashed"))
```

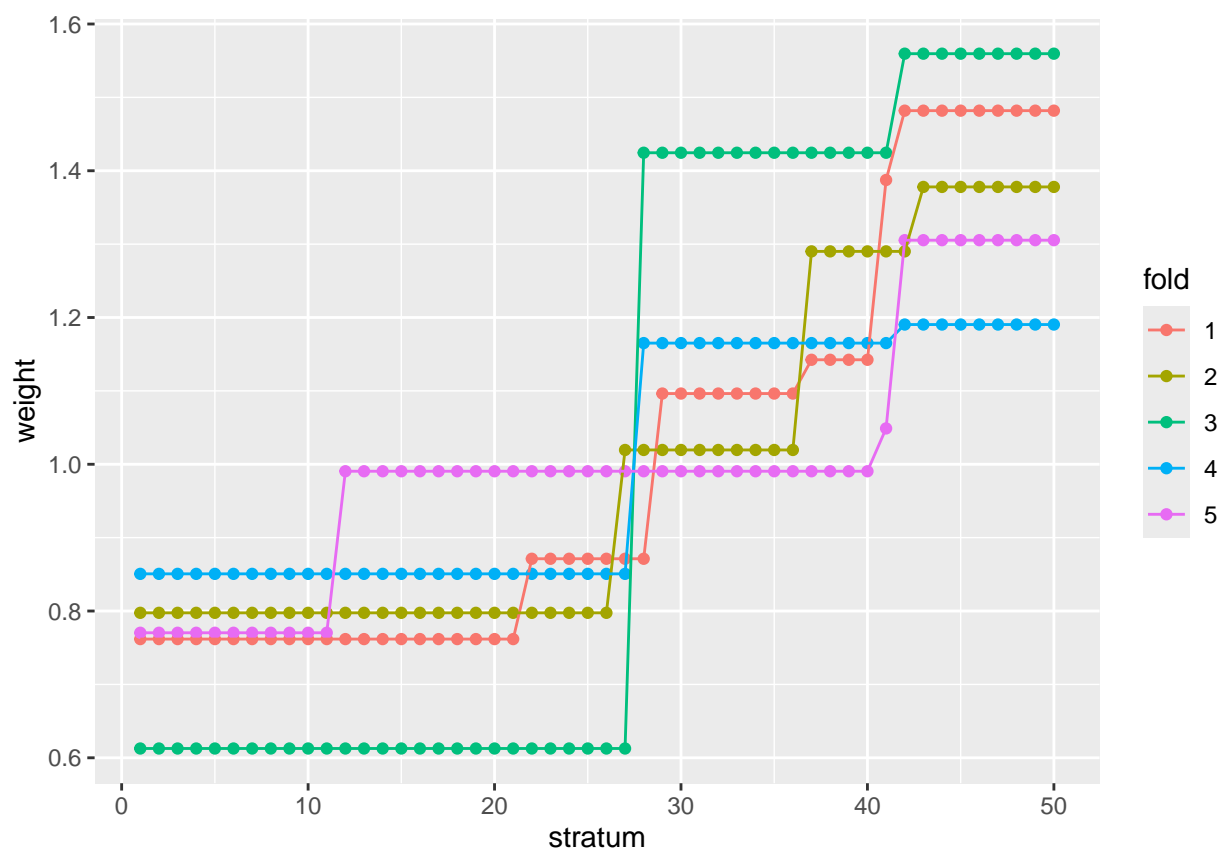
p1



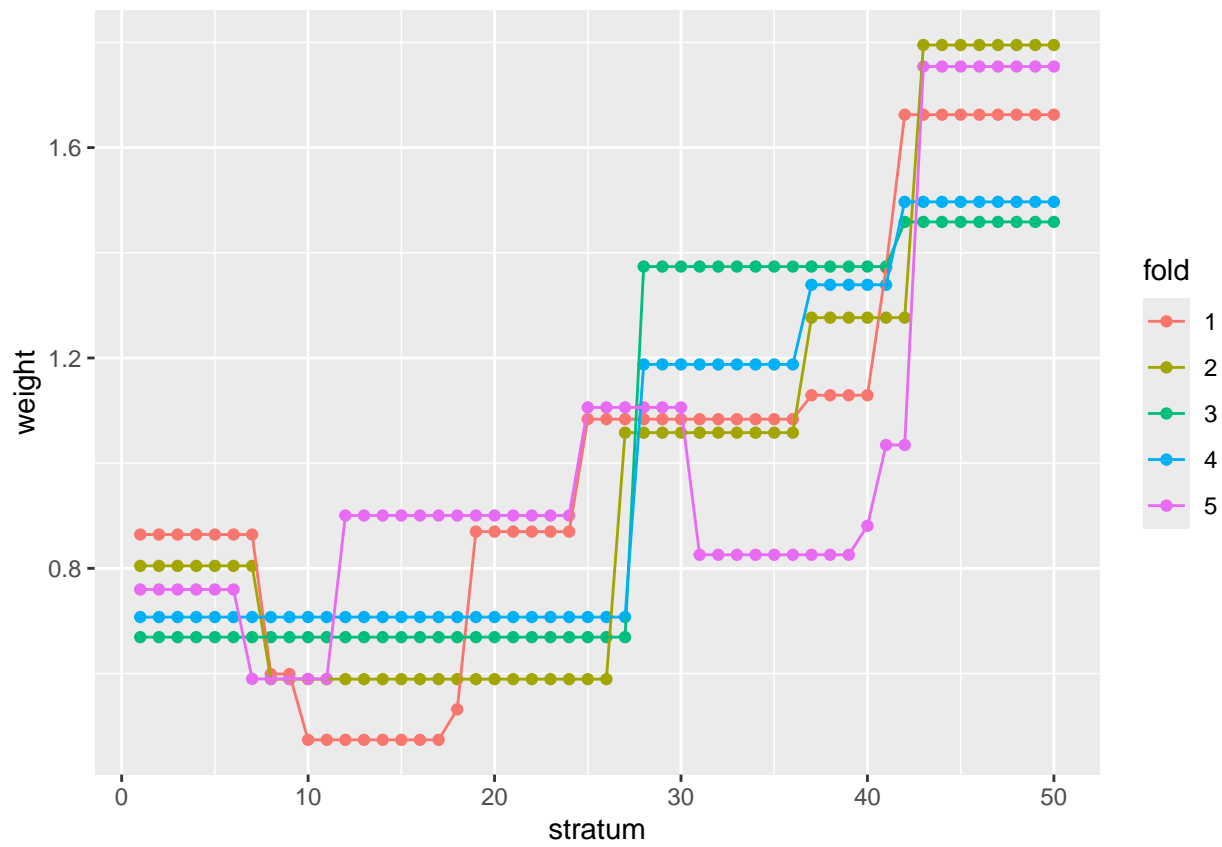
p2



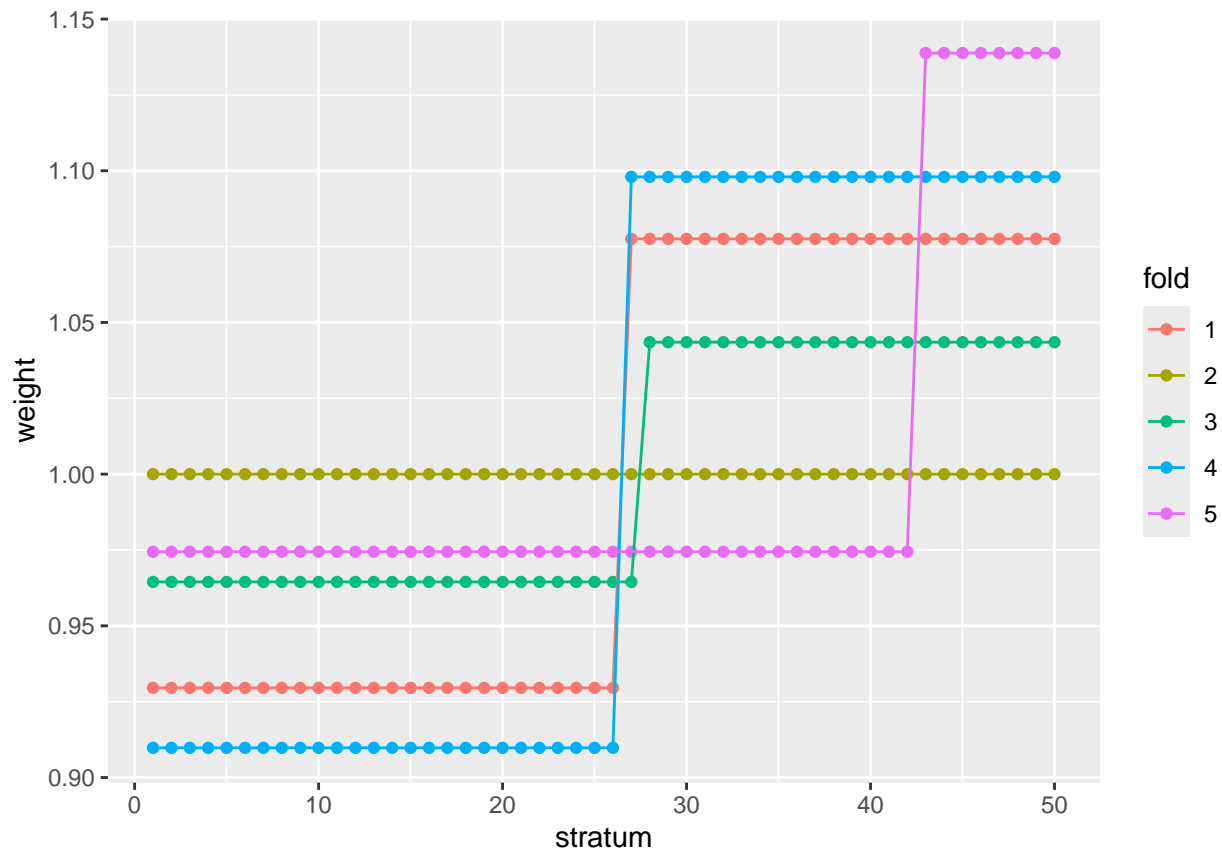
```
#plot weights: num bins = 50  
IHW:::plot(ihw_res_admm)
```



```
IHW: plot(ihw_res_old)
```



IHW: `plot(ihw_res_new)`



## 2nd example

```
library("IHWpaper")
library("IHW")
bottomly <- analyze_dataset("bottomly")
```

```
## Note: levels of factors in the design contain characters other than
## letters, numbers, '_' and '.'. It is recommended (but not required) to use
## only letters, numbers, and delimiters '_' or '.', as these are safe characters
## for column names in R. [This is a message, not a warning or an error]
```

```
## estimating size factors
```

```
## Note: levels of factors in the design contain characters other than
## letters, numbers, '_' and '.'. It is recommended (but not required) to use
## only letters, numbers, and delimiters '_' or '.', as these are safe characters
## for column names in R. [This is a message, not a warning or an error]
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## Note: levels of factors in the design contain characters other than
## letters, numbers, '_' and '.'. It is recommended (but not required) to use
## only letters, numbers, and delimiters '_' or '.', as these are safe characters
## for column names in R. [This is a message, not a warning or an error]
```

```
## final dispersion estimates
```

```
## Note: levels of factors in the design contain characters other than
## letters, numbers, '_' and '.'. It is recommended (but not required) to use
## only letters, numbers, and delimiters '_' or '.', as these are safe characters
## for column names in R. [This is a message, not a warning or an error]

## fitting model and testing

## Note: levels of factors in the design contain characters other than
## letters, numbers, '_' and '.'. It is recommended (but not required) to use
## only letters, numbers, and delimiters '_' or '.', as these are safe characters
## for column names in R. [This is a message, not a warning or an error]

## -- replacing outliers and refitting for 4 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## Note: levels of factors in the design contain characters other than
## letters, numbers, '_' and '.'. It is recommended (but not required) to use
## only letters, numbers, and delimiters '_' or '.', as these are safe characters
## for column names in R. [This is a message, not a warning or an error]

## fitting model and testing

## Note: levels of factors in the design contain characters other than
## letters, numbers, '_' and '.'. It is recommended (but not required) to use
## only letters, numbers, and delimiters '_' or '.', as these are safe characters
## for column names in R. [This is a message, not a warning or an error]

num_bins <- seq(from = 10, to = 50, by = 10)
N <- length(num_bins)
```

## Run the three methods

```
time.new <- rep(NA, N)
time.old <- rep(NA, N)
#time.cvxr <- rep(NA, N)

rej_new <- rep(NA, N)
rej_old <- rep(NA, N)
#rej_cvxr <- rep(NA, N)

for(i in 1:N){
  time.new[i] <- system.time(ihw_res_new <- ihw(bottomly$pvalue, bottomly$baseMean, 0.1, nbins = num_bins))
  rej_new[i] <- IHW::rejections(ihw_res_new)

  time.old[i] <- system.time(ihw_res_old <- IHWold::ihwOld(bottomly$pvalue, bottomly$baseMean, 0.1, nbins = num_bins))
  rej_old[i] <- IHW::rejections(ihw_res_old)

  #time.cvxr[i] <- system.time(ihw_res_cvxr <- IHW::ihw(bottomly$pvalue, bottomly$baseMean, 0.1, nbins = num_bins))
  #rej_cvxr[i] <- IHW::rejections(ihw_res_cvxr)

  print(paste0("Ending iteration ", i))
}
```



```

## [1] "Ending iteration 1"
## We recommend that you supply (many) more than 1000 p-values for meaningful data-driven hypothesis we
## [1] "Ending iteration 2"
## We recommend that you supply (many) more than 1000 p-values for meaningful data-driven hypothesis we
## [1] "Ending iteration 3"
## We recommend that you supply (many) more than 1000 p-values for meaningful data-driven hypothesis we
## [1] "Ending iteration 4"
## We recommend that you supply (many) more than 1000 p-values for meaningful data-driven hypothesis we
## [1] "Ending iteration 5"

load("~/Library/Mobile Documents/com~apple~CloudDocs/Desktop/UchicagoStudy/Nikolaos Research/IHW_New/CV)
load("~/Library/Mobile Documents/com~apple~CloudDocs/Desktop/UchicagoStudy/Nikolaos Research/IHW_New/CV)
load("~/Library/Mobile Documents/com~apple~CloudDocs/Desktop/UchicagoStudy/Nikolaos Research/IHW_New/CV)

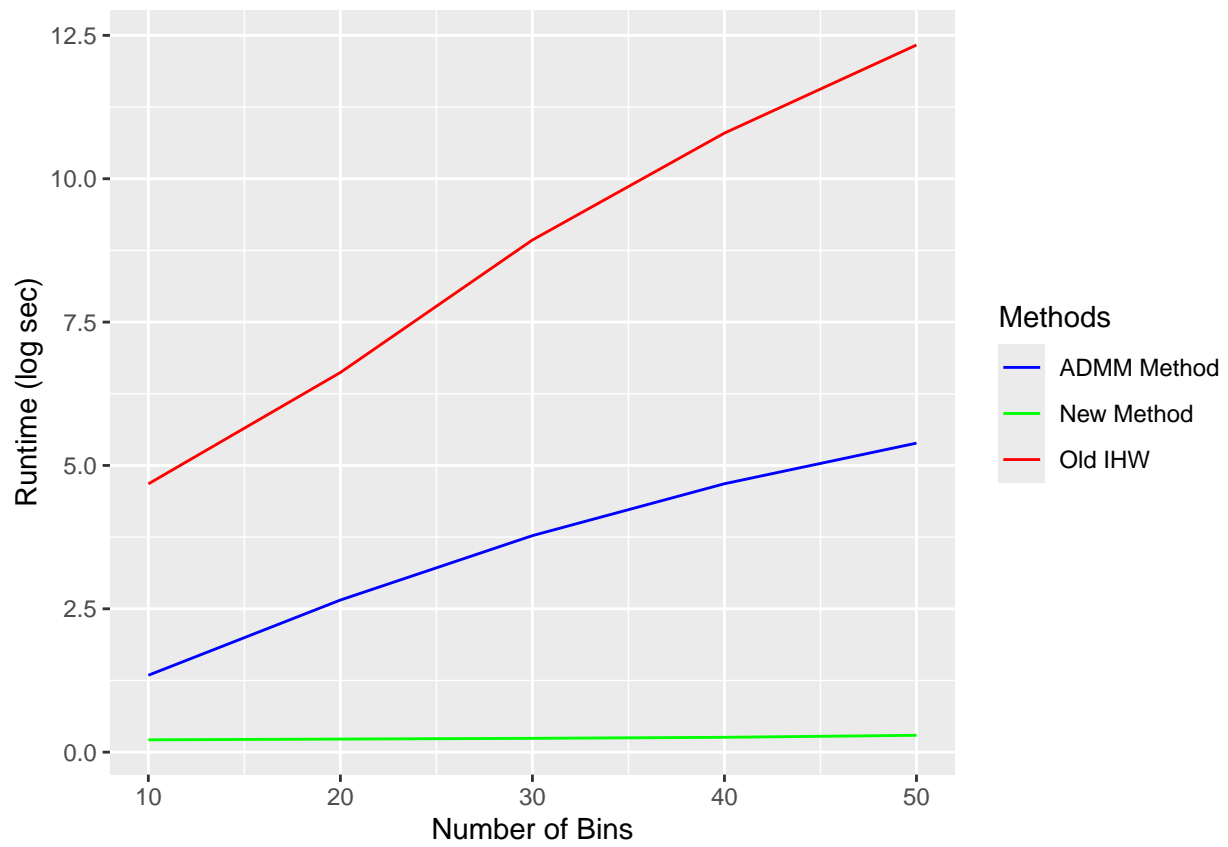
data <- data.frame(num_bins = num_bins, rejection_new = rej_new[1:5], rejection_admm = rej_admm[1:5], re

library(tidyverse)
#we first plot runtime versus bins
# Create the line plot
p1 <- ggplot(data) +
  geom_line(aes(x = num_bins, y = time_admm, col = "ADMM Method", linetype = "ADMM Method")) +
  geom_line(aes(x = num_bins, y = time_old, col = "Old IHW", linetype = "Old IHW")) +
  geom_line(aes(x = num_bins, y = time_new, col = "New Method", linetype = "New Method")) +
  labs(x = "Number of Bins", y = "Runtime (log sec)") +
scale_color_manual(values = c("ADMM Method" = "blue", "Old IHW" = "red", "New Method" = "green"), name =
  scale_linetype_manual(values = c("ADMM Method" = "solid", "Old IHW" = "solid", "New Method" = "solid")

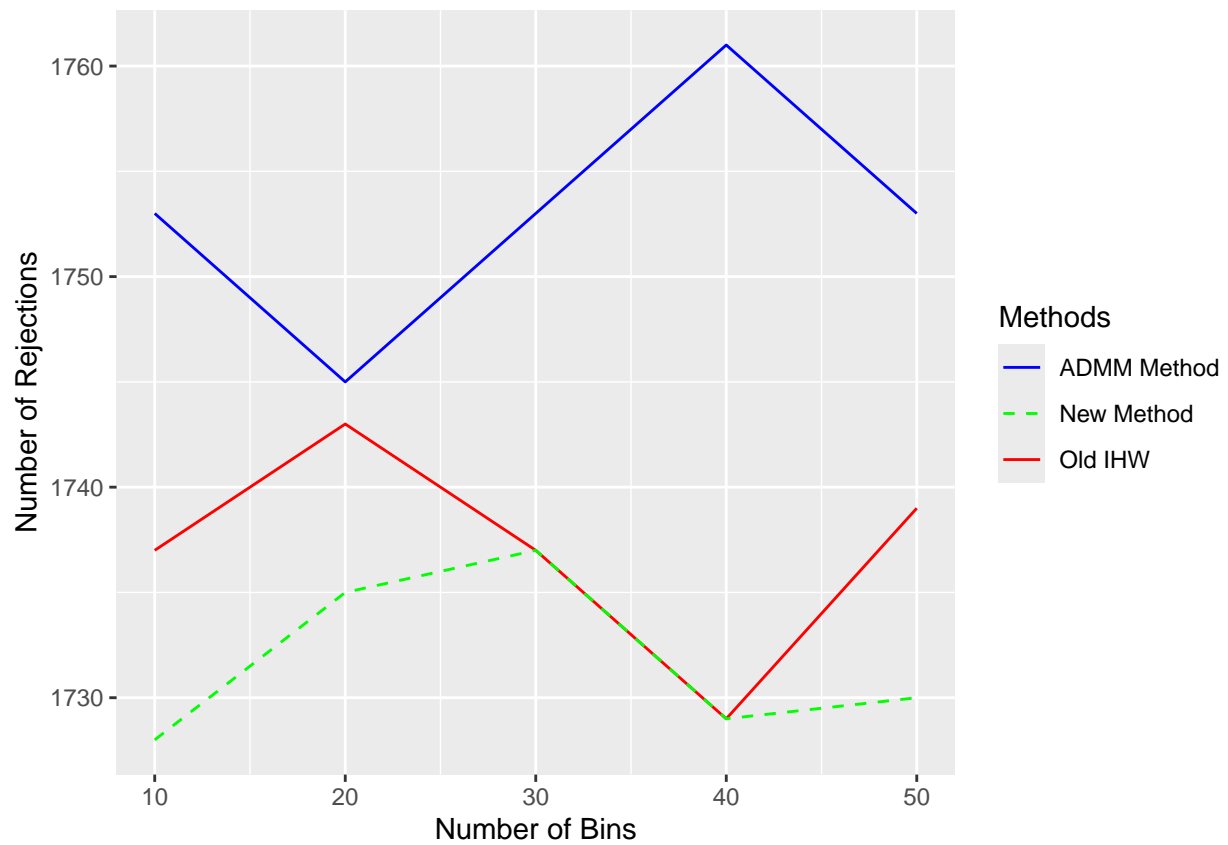
#we then plot rejections versus bins
# Create the line plot
p2 <- ggplot(data) +
  geom_line(aes(x = num_bins, y = rejection_admm, col = "ADMM Method", linetype = "ADMM Method")) +
  geom_line(aes(x = num_bins, y = rejection_old, col = "Old IHW", linetype = "Old IHW")) +
  geom_line(aes(x = num_bins, y = rejection_new, col = "New Method", linetype = "New Method")) +
  labs(x = "Number of Bins", y = "Number of Rejections") +
scale_color_manual(values = c("ADMM Method" = "blue", "Old IHW" = "red", "New Method" = "green"), name =
  scale_linetype_manual(values = c("ADMM Method" = "solid", "Old IHW" = "solid", "New Method" = "dashed")

p1

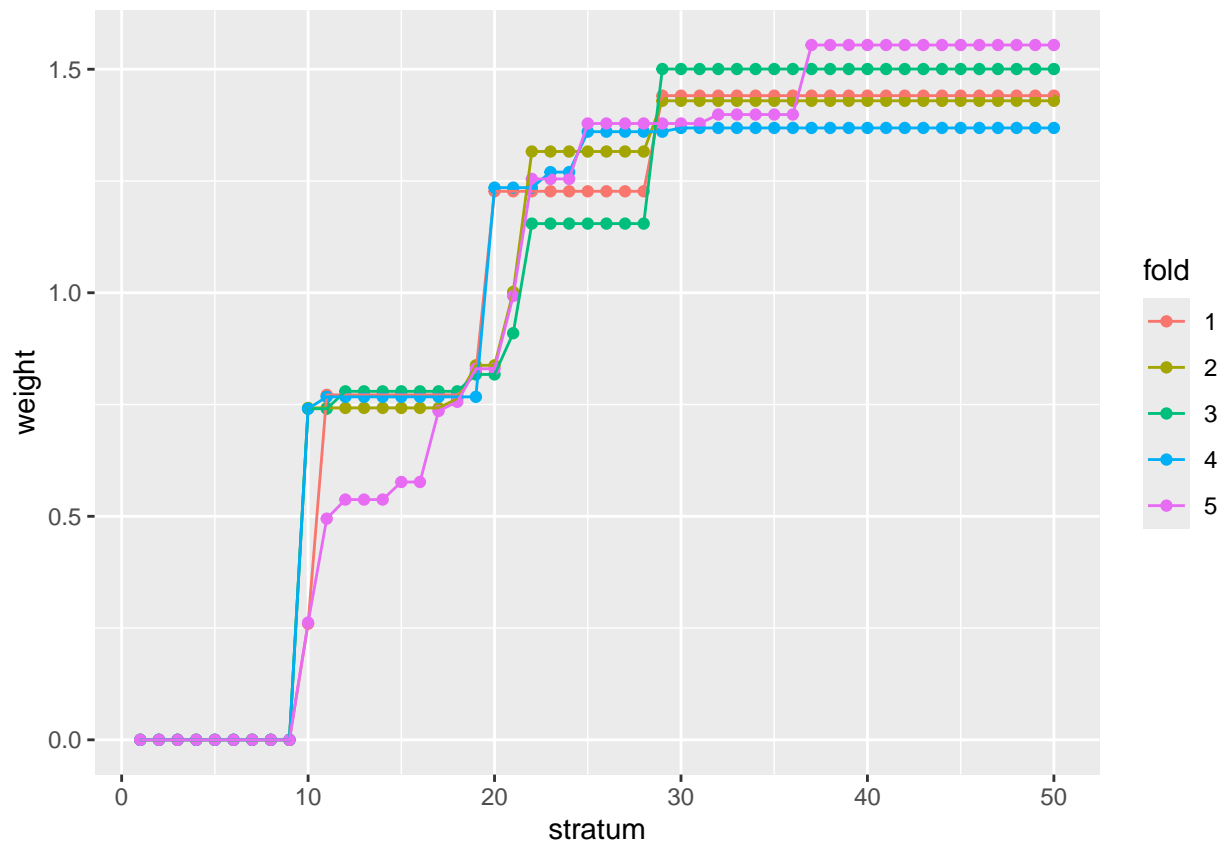
```



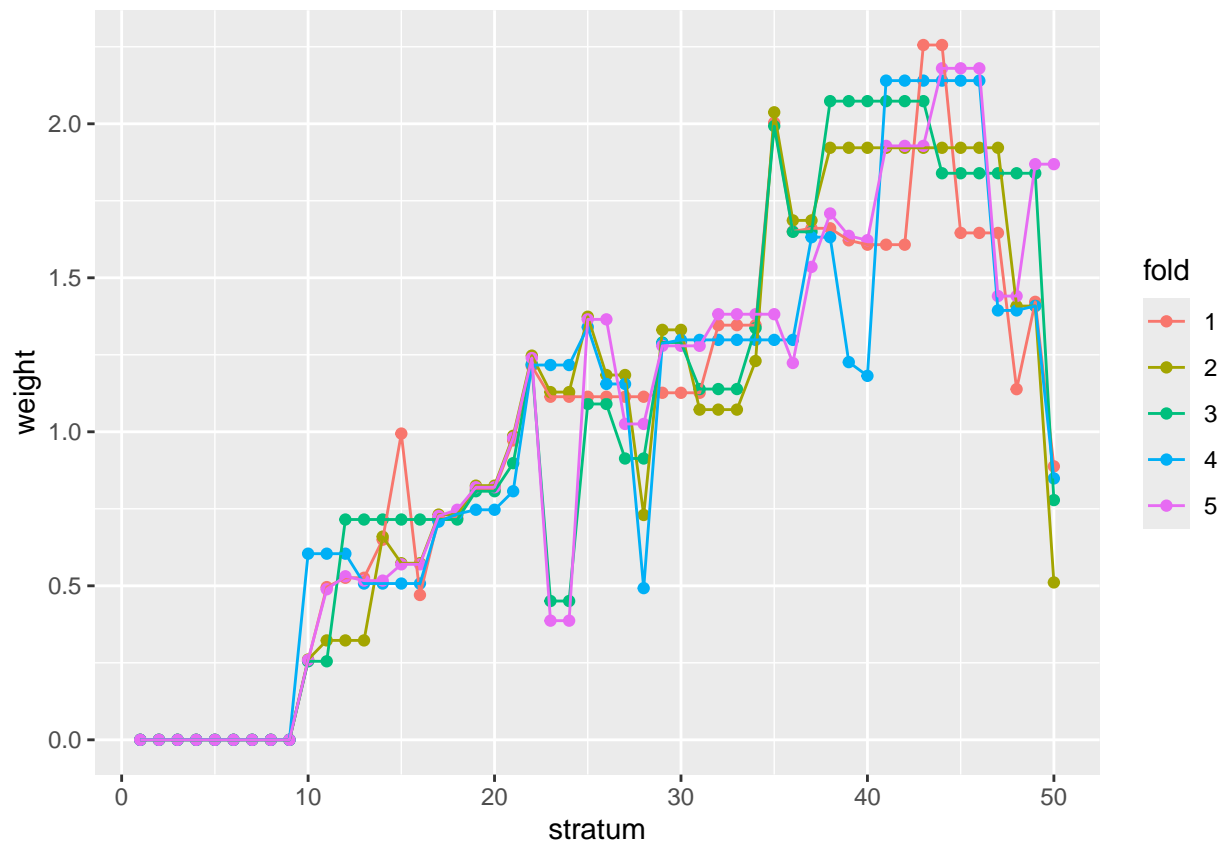
p2



```
#plot weights: num bins = 50  
IHW:::plot(ihw_res_admm)
```



```
IHW::plot(ihw_res_old)
```



```
IHW::plot(ihw_res_new)
```

