Advent of Code 2021 - Day 7 Speed Edition

Gus Lipkin ~ github.com/guslipkin/AdventOfCode2021

Please don't look at this. It's embarrassing.

I only did one rep because my first try is too slow.

Fastest solution

```
library(stringr)
library(tidyverse)
library(data.table)


dt <- as.numeric(data.table::fread("input.txt", header = FALSE)[1])
# part 1
s <- max(dt) ^ length(dt)
for (i in-ceiling(max(dt) / 2):ceiling(max(dt) / 2)) {
  dt2 <-
    data.table("start" = dt)[, mean := floor(mean(start))][, diff := abs(start - mean - i)]
  dtDiff <- sum(dt2$diff)
  s <- data.table::fifelse(dtDiff < s, dtDiff, s)
}
s <- c(s)

# part 2
cSum <- function(y) {
  eval(parse(text = y))
}
dt2 <- data.table("start" = dt)
dt2[, mean := mean(dt2$start, na.rm = TRUE)]
dt2$abs <- abs(dt2$start - dt2$mean)
dt2$meanC <- ceiling(dt2$mean)
dt2$meanF <- floor(dt2$mean)
dt2$absC <- abs(dt2$start - dt2$meanC)
dt2$absF <- abs(dt2$start - dt2$meanF)
dt2$diffC <- sapply(paste0("sum(1:", dt2$absC, ")"), cSum)
dt2$diffF <- sapply(paste0("sum(1:", dt2$absF, ")"), cSum)
s <- min(sum(dt2$diffC), sum(dt2$diffF))
```

Benchmark

```r
bench <- rbenchmark::benchmark(
  "first" = {
    library(stringr)
    library(tidyverse)
    library(data.table)
    dt <-
      as.numeric(data.table::fread("input.txt", header = FALSE)[1])
    # part 1
    dt2 <- data.table("start" = dt)
    s <- c()
    for (i in-max(dt):max(dt)) {
      dt2 <- data.table("start" = dt)
      dt2[, mean := floor(mean(start))][, diff := abs(start - mean + i)]
      s <- append(s, sum(dt2$diff))
    }
    s <- min(s)

    # part 2
    dt2 <- data.table("start" = dt)
    s <- c()

    for (i in-max(dt):max(dt)) {
      dt2 <- data.table("start" = dt)
      dt2[, mean := floor(mean(start))]
      for (r in 1:nrow(dt2))
        dt2$diff[r] <- sum(1:(abs(dt2$start[r] - dt2$mean[r] - i)))
      s <- append(s, sum(dt2$diff))
    }
    s <- min(s)
  },
  "second" = {
    library(stringr)
    library(tidyverse)
    library(data.table)
    dt <-
      as.numeric(data.table::fread("input.txt", header = FALSE)[1])
    # part 1
    s <- max(dt) ^ length(dt)
    for (i in-ceiling(max(dt) / 2):ceiling(max(dt) / 2)) {
      dt2 <-
        data.table("start" = dt)[, mean := floor(mean(start))][, diff := abs(start - mean - i)]
      dtDiff <- sum(dt2$diff)
      s <- data.table::fifelse(dtDiff < s, dtDiff, s)
    }
    s <- c(s)

    cSum <- function(y) {
      eval(parse(text = y))
    }
    dt2 <- data.table("start" = dt)
    dt2[, mean := mean(dt2$start, na.rm = TRUE)]
    dt2$abs <- abs(dt2$start - dt2$mean)
    dt2$meanC <- ceiling(dt2$mean)
    dt2$meanF <- floor(dt2$mean)
    dt2$absC <- abs(dt2$start - dt2$meanC)
    dt2$absF <- abs(dt2$start - dt2$meanF)
    dt2$diffC <- sapply(paste0("sum(1:", dt2$absC, ")"), cSum)
    dt2$diffF <- sapply(paste0("sum(1:", dt2$absF, ")"), cSum)
    min(sum(dt2$diffC), sum(dt2$diffF))
  },
  replications = 1,
  columns = 1:5,
  order = "user.self"
)

bench$per <- bench$user.self / bench$replications
bench
```

```
##     test replications user.self sys.self elapsed    per
## 2 second            1     0.551    0.014   0.565  0.551
## 1  first            1    70.408    6.040  76.479 70.408
```