Advent of Code 2021 - Day 2 Speed Edition

Gus Lipkin ~ github.com/guslipkin/AdventOfCode2021

Base R once again beats out data.table by just a little bit. The big difference here was moving from `colSums` to `sum`, although `readLines` instead of `fread` does play a part.

Fastest solution

```
dtM <- data.frame("V1" = readLines("input.txt"))
dtM <- tidyr::separate(dtM, "V1", c("d", "n"), sep = " ", convert = TRUE)

# part 1
dt <- dtM
dt$n <- ifelse(dt$d == "down", dt$n * -1, dt$n)
abs(sum(dt$n[dt$d == "forward"]) * sum(dt$n[dt$d != "forward"]))
```

```
## [1] 1635930
```

```
# part 2
dt <- dtM
dt$aim <- cumsum(ifelse(dt$d == "up", dt$n * -1,
                        ifelse(dt$d == "forward", 0, dt$n)))
sum(ifelse(dt$d == "forward", dt$n, 0), na.rm = TRUE) *
  sum(ifelse(dt$d == "forward", dt$aim * dt$n, 0), na.rm = TRUE)
```

```
## [1] 1781819478
```

Benchmark

```
rbenchmark::benchmark(
  "First try" = {
    library(data.table)
    # part 1
    dt <- data.table(read.table("input.txt", sep = "\n"))
    dt <- tidyr::separate(dt, "V1", c("d", "n"), sep = " ", convert = TRUE)
    dt$n <- ifelse(dt$d == "down", dt$n * -1, dt$n)

    abs(dt[d == "forward", .(sum = sum(n))] *
          as.numeric(dt[d != "forward", .(sum = sum(n))]))

    # part 2
    dt <- data.table(read.table("input.txt", sep = "\n"))
    dt <- tidyr::separate(dt, "V1", c("d", "n"), sep = " ", convert = TRUE)

    dt$aim <- ifelse(dt$d == "up" & dt$d != "forward", dt$n * -1, dt$n)
    dt$aim <- ifelse(dt$d == "forward", 0, dt$aim)
    dt$aim <- cumsum(dt$aim)

    dt$h <- ifelse(dt$d == "forward", dt$n, 0)
    dt$depth <- ifelse(dt$d == "forward", dt$aim * dt$n, 0)

    sum(dt$h, na.rm = TRUE) * sum(dt$depth, na.rm = TRUE)
    },
  "Base R" = {
    dtM <- data.frame("V1" = readLines("input.txt"))
    dtM$n <- na.omit(as.numeric(unlist(strsplit(dtM$V1, " "))))
    dtM$d <- as.character(unlist(strsplit(dtM$V1, " ")))[
      is.na(as.numeric(unlist(strsplit(dtM$V1, " "))))]

    # part 1
    dt <- dtM
    dt$n <- ifelse(dt$d == "down", dt$n * -1, dt$n)
    abs(sum(dt$n[dt$d == "forward"]) * sum(dt$n[dt$d != "forward"]))

    # part 2
    dt <- dtM
    dt$aim <- cumsum(ifelse(dt$d == "up", dt$n * -1,
                            ifelse(dt$d == "forward", 0, dt$n)))
    sum(ifelse(dt$d == "forward", dt$n, 0), na.rm = TRUE) *
      sum(ifelse(dt$d == "forward", dt$aim * dt$n, 0), na.rm = TRUE)
    },
  "data.table" = {
    library(data.table)
    dtM <- fread("input.txt", sep = "\n", header = FALSE)
    dtM <- tidyr::separate(dtM, "V1", c("d", "n"), sep = " ", convert = TRUE)

    # part 1
    dt <- dtM
    dt$n <- ifelse(dt$d == "down", dt$n * -1, dt$n)
    abs(colSums(dt[d == "forward", .(n)]) * colSums(dt[d != "forward", .(n)]))

    # part 2
    dt <- dtM
    dt$aim <- cumsum(ifelse(dt$d == "up", dt$n * -1,
                            ifelse(dt$d == "forward", 0, dt$n)))
    sum(ifelse(dt$d == "forward", dt$n, 0), na.rm = TRUE) *
      sum(ifelse(dt$d == "forward", dt$aim * dt$n, 0), na.rm = TRUE)
    },
  replications = 1000, columns = c(1:5), order = "user.self")
```

```
##         test replications user.self sys.self elapsed
## 2     Base R         1000     2.060    0.055   2.116
## 3 data.table         1000     5.814    0.450   6.266
## 1  First try         1000    11.711    0.973  12.696
```