# Filling Gap Weeks

Purpose: Proper implementation of **`Filling Gaps`** functionality (Code can be executed in project's dir).

## Setup

- After loading the data, we get variables of interest from the internal dataframe **`datWeeks`**.
  - The variable **`enrolled`** is the number of subjects enrolled in each week.
  - The variable **`cnt`** is the number of active recruitment days in each week.

```
load_all()
## ℹ Loading RCTRecruit
LoadData(gripsIM, ScreenDt, Enrolled)
## Variables Enrolled and ScreenDt were successfully loaded from data gripsIM
train <- the$datWeeks$enrolled
nonGap <- the$datWeeks$cnt != 0
nonGapVals <- train[nonGap]
gapIdx <- which(!nonGap)
the$datWeeks[gapIdx, ]
```

| | week | year | enrolled | holiday | cnt |
|---|---|---|---|---|---|
| 14 | 38 | 2019 | 0 | 0 | 0 |
| 15 | 39 | 2019 | 0 | 0 | 0 |
| 16 | 40 | 2019 | 0 | 0 | 0 |
| 17 | 41 | 2019 | 0 | 0 | 0 |
| 18 | 42 | 2019 | 0 | 1 | 0 |
| 20 | 44 | 2019 | 0 | 0 | 0 |
| 21 | 45 | 2019 | 0 | 0 | 0 |
| 22 | 46 | 2019 | 0 | 1 | 0 |
| 23 | 47 | 2019 | 0 | 0 | 0 |
| 24 | 48 | 2019 | 0 | 1 | 0 |
| 25 | 49 | 2019 | 0 | 0 | 0 |
| 28 | 52 | 2019 | 0 | 1 | 0 |
| 41 | 13 | 2020 | 0 | 0 | 0 |
| 42 | 14 | 2020 | 0 | 0 | 0 |
| 43 | 15 | 2020 | 0 | 1 | 0 |
| 44 | 16 | 2020 | 0 | 0 | 0 |
| 45 | 17 | 2020 | 0 | 0 | 0 |

We see there are 17 gap weeks. I will refer to each gap week by its order in the dataset (or rowname) and I wont' use the variable **`week`**. i.e. I will use **`Week 14`** insted of **`Week 38`**.

### Calculating Sampling Weights

- I used the binomial weights used in the rest of the package while
  - The weights for all gap weeks are set to zero.
  - Weights are normalized to sum to 1.

```
bnWt <- stats::dbinom(0L:51L, 51L, 0.5)
idWt <- \(t) ((0L:51L + 26L - t) %% 52L) + 1L
getWts <- \(t) (bnWt[idWt(t)] * nonGap) |> (\(x) x / sum(x))()
p <- lapply(gapIdx, getWts) |> setNames(paste("Week", gapIdx))
sapply(p, which.max)
```

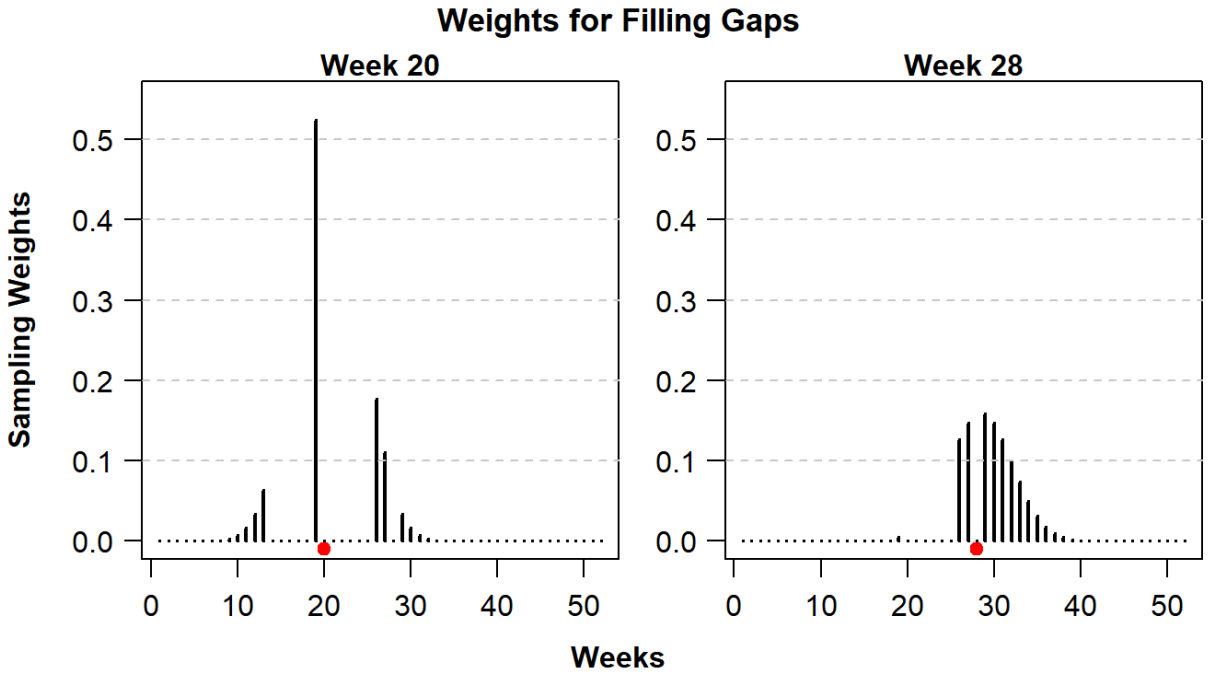
```
## Week 14 Week 15 Week 16 Week 17 Week 18 Week 20 Week 21 Week 22 Week 23 Week 24
##      13      13      19      19      19      19      19      19      26      26
## Week 25 Week 28 Week 41 Week 42 Week 43 Week 44 Week 45
##      26      29      40      40      46      46      46
```

The variable **p** is a list of 17 vectors, with each vector containing the weights for the corresponding gap week. The output above shows which **non-gap** week has the highest weight for each gap week.

### Visual Inspection of sampling weights

For visual inspection, I plot the weights for the gap weeks 20 and 28.

```
pPar <- list(ylim = c(0, 0.55), type = "h", xlab = "", lwd = 2)
gPar <- list(las = 1, font.lab = 2, cex.main = 1, cex.lab = 1, cex.axis = 1,
  fig = c(0, .5, 0, 1), mar = c(1, 4, 1, 0), mgp = c(3, .8, 0), pty = "s"
)
plotWeek <- \(x, ylab = "", y = paste("Week", x)) {
  pPar[c("x", "ylab", "main")] <-  list(p[[y]], ylab, y)
  do.call(par, gPar); do.call(plot, pPar);
  points(x, -.01, pch = 19, col = "red")
  abline(h = c(1:5) / 10, lty = 2, col = "gray80")
}
oldPar <- par(no.readonly = TRUE)
plotWeek(20, "Sampling Weights")
gPar[c("fig", "mar",  "new")] = list(c(.5, 1, 0, 1), mar = c(1, 3, 1, 1), TRUE)
plotWeek(28)
mtext("Weights for Filling Gaps", 3, -2, TRUE, cex = 1.1, font = 2)
mtext("Weeks", 1, -1, TRUE, font = 2)
```




```
do.call(par, oldPar)
```

We observe that there are gaps in what would have been a bell-shaped curve. Also, in the case of **Week 20**, the weight of week 19 is very high as there are not other non-gap weeks in the vicinity.

## Simulation Results

I will refer to the weights created above simply as **`weights`**. I run 10,000 simulations, for each of 5 different sampling methods and I present the generated sample's mean and CV [MEAN (SD/MEAN)] for each gap week. The five sampling method are:

- **Bootstrap**: Sample 1 value from the non-gap weeks (equal weights for all).
- **Binomial**: Sample 1 value using the weights.
- **BinomialMu3**, **BinomialMu6** and **BinomialMu9**: Mean of 3, 6, or 9 sampled values using the weights.

The results of each method are presented as a column in the table below. The first column is the dot product of the real sample with the corresponding weights of each gap week (Expected Enrollment Value). The last row is resulting total enrollment as [Total (SD of Total)] (reminder: init N = 18)

```r
len <- length(gapIdx)
wk <- seq_len(len) |> setNames(names(p))

btstrp <- \() vapply(wk, \(x) sample(nonGapVals, 1), 0L)
bnm <- \(n = 1) vapply(wk, \(x) mean(sample(train, n, FALSE, p[[x]])), .0)
list2df <- \(x) do.call(rbind, x) |> as.data.frame()

nSim <- seq_len(1e4)
dfs <- list()
dfs[["Bootstrap"]] <- lapply(nSim, \(x) btstrp()) |> list2df()
dfs[["Binom"]]     <- lapply(nSim, \(x) bnm(1L))  |> list2df()
dfs[["BinomMu3"]]  <- lapply(nSim, \(x) bnm(3L))  |> list2df()
dfs[["BinomMu6"]]  <- lapply(nSim, \(x) bnm(6L))  |> list2df()
dfs[["BinomMu9"]]  <- lapply(nSim, \(x) bnm(9L))  |> list2df()


reportMeanCVUnit <- \(x) sprintf("%5.2f (%3.1f)", mean(x), sd(x) / mean(x))
reportMeanSdUnit <- \(x) sprintf("%5.2f (%3.1f)", mean(x), sd(x))
out <- numeric(len) |> setNames(names(p))
reportMeanCV <- function(df) {
  out <- lapply(df, reportMeanCVUnit) |> cbind()
  N <- ((apply(df, 1, sum)) + 18) |> reportMeanSdUnit()
  rbind(out, N)
}
for (i in seq_len(len)) out[i] <- sum(train * p[[i]])
WtMean <- c(out, 18 + sum(out)) |> sprintf(fmt = "%5.2f", ... = _)
MeanCV <- lapply(dfs, reportMeanCV) |> as.data.frame()

cbind(WtMean, MeanCV)
```

| | WtMean | Bootstrap | Binom | BinomMu3 | BinomMu6 | BinomMu9 |
|---|---|---|---|---|---|---|
| Week 14 | 0.14 | 0.52 (1.5) | 0.13 (2.6) | 0.15 (1.3) | 0.17 (0.6) | 0.21 (0.4) |
| Week 15 | 0.21 | 0.52 (1.5) | 0.21 (2.0) | 0.21 (0.9) | 0.19 (0.5) | 0.24 (0.4) |
| Week 16 | 0.33 | 0.51 (1.5) | 0.33 (1.5) | 0.28 (0.7) | 0.24 (0.6) | 0.29 (0.4) |
| Week 17 | 0.47 | 0.51 (1.5) | 0.49 (1.1) | 0.37 (0.6) | 0.32 (0.5) | 0.33 (0.2) |
| Week 18 | 0.64 | 0.52 (1.5) | 0.64 (0.9) | 0.50 (0.6) | 0.41 (0.4) | 0.34 (0.1) |
| Week 20 | 0.88 | 0.51 (1.5) | 0.89 (0.8) | 0.75 (0.4) | 0.50 (0.1) | 0.35 (0.2) |
| Week 21 | 0.91 | 0.51 (1.5) | 0.92 (0.8) | 0.80 (0.4) | 0.52 (0.2) | 0.39 (0.3) |
| Week 22 | 0.88 | 0.49 (1.5) | 0.88 (1.0) | 0.78 (0.4) | 0.55 (0.2) | 0.46 (0.3) |
| Week 23 | 0.81 | 0.53 (1.5) | 0.80 (1.1) | 0.71 (0.5) | 0.57 (0.3) | 0.55 (0.2) |
| Week 24 | 0.72 | 0.50 (1.5) | 0.72 (1.3) | 0.65 (0.5) | 0.57 (0.3) | 0.62 (0.2) |
| Week 25 | 0.64 | 0.53 (1.5) | 0.64 (1.4) | 0.59 (0.6) | 0.56 (0.3) | 0.64 (0.2) |
| Week 28 | 0.52 | 0.51 (1.5) | 0.51 (1.7) | 0.54 (0.8) | 0.58 (0.4) | 0.69 (0.2) |
| Week 41 | 0.63 | 0.51 (1.5) | 0.64 (1.2) | 0.66 (0.6) | 0.72 (0.3) | 0.77 (0.2) |
| Week 42 | 0.58 | 0.51 (1.5) | 0.58 (1.3) | 0.61 (0.7) | 0.65 (0.4) | 0.70 (0.2) |
| Week 43 | 0.56 | 0.51 (1.5) | 0.56 (1.4) | 0.57 (0.7) | 0.59 (0.4) | 0.62 (0.2) |
| Week 44 | 0.56 | 0.51 (1.5) | 0.56 (1.4) | 0.55 (0.7) | 0.55 (0.4) | 0.54 (0.2) |
| Week 45 | 0.55 | 0.52 (1.5) | 0.55 (1.4) | 0.54 (0.7) | 0.51 (0.4) | 0.50 (0.3) |
| N | 28.01 | 26.72 (3.2) | 28.03 (3.0) | 27.24 (1.4) | 26.19 (0.7) | 26.23 (0.5) |

The main takeaways from the table above are:

- **`Bootstrap`**: assigns the same value (0.5 subjects) on all weeks regardless of position, and variation is high (CV of 1.5).
- **`Binomial`**: Weekly means are close to the expected values, but the variability is even higher in some cases.
- **Mean of 3,6,9 sampled values**: As we take the mean of more values, the variability decreases but the mean progressively deviates from the Expected Value.

Finally, the table below shows the 95% confidence interval for the total of resulting enrollment (rounded to the closest integer) for each method.

```
lapply(dfs, \(x) { round(rowSums(x) + 18) |>  quantile(c(.025, .5, .975))}) |>
  do.call(rbind, args = _) |>
  as.data.frame()
```

| | 2.5% | 50% | 97.5% |
|---|---|---|---|
| Bootstrap | 21 | 27 | 33 |
| Binom | 22 | 28 | 34 |
| BinomMu3 | 25 | 27 | 30 |
| BinomMu6 | 25 | 26 | 28 |
| BinomMu9 | 25 | 26 | 27 |

Again, we observed that sampling only one value using the binomial weights is probably not the best approach because of the high variability.