Novel Randomized M&V Examples

2024-05-23

Table of Contents

# 1 Objective

The purpose of this document is to provide open-source code for analysts who wish to apply the novel measurement and verification (M&V) procedure described in the associated paper to their own projects or datasets. By following the guidance below, users should be able to:

1. Replicate the novel M&V method proposed in the manuscript:
   * Design a randomized and balanced switchback control implementation schedule consisting of a baseline strategy and an intervention strategy,
   * Conduct sequential analysis to infer intervention effect on a target metric (e.g. energy consumption, carbon emissions, or thermal comfort).
2. Demonstrate the reliability of the novel randomized method and compare it with the conventional method under:
   * Scenario A: Discrete non-routine events
   * Scenario B: Continuous baseline change

By replicating the method proposed in the manuscript, we aim to demonstrate that the novel M&V approach can reach a conclusion faster than the standards requirements and provide a robust uncertainty quantification. Additionally, by evaluating savings estimations under various common scenarios, we showcase the reliability of this new method in overcoming the limitations of conventional approaches (e.g. the impact of changes in building performance that are wholely unrelated to the intervention). This example code follows the flowchart depicted in the manuscript and provides a minimal example for demonstration purposes. The analyst should make reasonable adjustments to address their specific use case.

require(pacman)  
  
# load packages using pacman  
pacman::p\_load(tidyverse, lubridate, here, rmarkdown,   
 scales, patchwork, magrittr, janitor, qpcR, knitr, # general  
 ggpmisc, # linear regression  
 ggpubr,  
 sjstats, pwr, # anova results  
 nmecr, # m&v modeling package for TOWT  
 slider, # moving averages  
 sprtt, # sequential testing  
 base, blocksdesign, # blocking  
 rstatix, #pipe friendly stats  
 overlapping, # distribution overlapping percentage  
 effsize, bootES, dabestr) #effect size  
  
# turn off scientific notation  
options(scipen = 999, digits = 15)  
  
# set directory  
here::i\_am("Example.rmd")  
  
# set default theme for ggplot  
theme\_set(theme\_minimal())  
  
# define base ggplot theme  
theme\_update(plot.title = element\_text(size = 14, colour = "grey20", face = "bold", hjust = 0.5),  
 plot.subtitle = element\_text(size = 10, colour = "grey20", face = "italic", hjust = 0.5, margin = margin(b = 10)),  
 plot.caption = element\_text(size = 10, colour = "grey20", face = "italic", hjust = 0.5),  
 plot.background = element\_rect(fill = "white", colour = NA),  
 panel.grid.minor = element\_blank(),  
 panel.grid.major = element\_blank(),  
 axis.text = element\_text(size = 10),  
 strip.text = element\_text(size = 10, color = "grey20", face = "bold"),  
 strip.background = element\_blank())  
  
# define global color theme (from palatte "Set2")  
ls\_colours <- c("Baseline" = "#99d8c9",  
 "Measured baseline" = "#99d8c9",   
 "Adjusted baseline" = "#99d8c9",  
 "Projected baseline\n(no change)" = "#1b9e77",  
 "Intervention" = "#fdbb84",  
 "Measured interv" = "#fdbb84")  
  
# Get case study example data  
readfile\_path <- "./readfiles/"  
  
# Load defined functions  
function\_path <- "./functions/"

# 2 M&V use case

We apply the same use case described in the manuscript where a software-as-a-service control company offers a more efficient chiller control algorithm. Here, the analyst aims the determine the amount of HVAC energy saved by this control intervention. The overall M&V protocol regarding sampling ratio, carryover effect, blocking design and stopping criteria remain the same as the manuscript. For demonstration purposes, the M&V procedure here does not include an annual saving estimation normalized on typical weather using fitted energy models. Instead, we use the average treatment effect (ATE) by calculating the mean difference between baseline measurements and intervention measurements to estimate the energy-saved by the control algorithm (and normalize using the mean of the baseline measurements to obtain fractional savings).

# 3 Use case measurement dataset

For this demonstration, we simply create a hypothetical baseline and intervention measurement set by considering a linear function including outdoor temperature (), peak and off-peak hour indicator (, ) as independent variables:

The term accounts for random noise sampled from a normal distribution . The created measurement set for baseline and intervention controls is shown in figure 3.1. If the analyst follows the conventional M&V method described in ASHRAE Guideline 14 (G14) or the International Performance Measurement and Verification Protocol (IPMVP), then only baseline measurements are measurable/observable during the pre-retrofit period and only intervention measurements are measurable during the post-retrofit period. In other words, the control strategy switches between baseline and intervention one time in the entire study period.

# load holidays  
list\_holidays <- read\_csv(paste0(readfile\_path, "us\_holidays\_2022.csv"),  
 col\_names = c("date", "weekday", "holiday")) %>%  
 mutate(date = mdy(date))  
  
# load tmy weather file  
df\_tmy <- read\_csv(paste0(readfile\_path, "USA\_IL\_Chicago.Midway.Intl.AP.725340\_TMY3.epw"),  
 skip = 8, col\_types = "ddddd-d---------------------------------",  
 col\_names = c("year", "month", "day", "hour", "min", "tmy")) %>%  
 mutate(year = 2022,  
 time = ymd\_h(paste(paste(year, month, day, sep = "-"), hour, sep = " ")),  
 temp = tmy) %>%  
 dplyr::select(time, temp)  
  
# load temperature from local weather station  
df\_weather <- list.files(path = readfile\_path, pattern = str\_glue("weather\_\*"), full.name = TRUE) %>%  
 map\_dfr(read\_rds) %>%  
 mutate(datetime = with\_tz(datetime\_UTC, tz = "America/Chicago")) %>%   
 select(c(datetime, t\_out)) %>%   
 mutate(across(t\_out, ~ zoo::na.approx(., na.rm = FALSE))) %>%  
 filter(datetime >= as.Date("2021-01-01")) %>%   
 filter(datetime < as.Date("2023-01-01")) %>%   
 unique() %>%   
 arrange(datetime)  
  
# Create a sequence of hourly timestamps for one year  
timestamps <- seq(from = as.POSIXct("2021-01-01 00:00:00"),  
 to = as.POSIXct("2022-12-31 23:00:00"),  
 by = "hour")  
  
# Define peak hours (e.g., 12 PM to 6 PM)  
peak\_hours <- as.integer(format(timestamps, "%H") %in% c("12", "13", "14", "15", "16", "17"))  
  
# Calculate non-peak hours  
non\_peak\_hours <- 1 - peak\_hours  
  
# Baseline model parameters  
intercept <- 45  
beta\_peak <- 20  
beta\_temp <- 0.9  
beta\_non\_peak <- 5  
  
# Compute the energy consumption  
energy\_consumption <- intercept +  
 beta\_temp \* df\_weather$t\_out +  
 beta\_peak \* peak\_hours +  
 beta\_non\_peak \* non\_peak\_hours +  
 rnorm(length(timestamps), mean = 0, sd = sqrt(100))  
  
# Create a data frame  
df\_base <- data.frame(datetime = timestamps,  
 t\_out = df\_weather$t\_out ,  
 power = energy\_consumption)   
  
# Output the first few rows of the dataset  
head(df\_base)

## datetime t\_out power  
## 1 2021-01-01 00:00:00 -1.66666666666667 56.6160321203583  
## 2 2021-01-01 01:00:00 -1.54000000000000 54.7324532275127  
## 3 2021-01-01 02:00:00 -1.41333333333333 45.7907255177579  
## 4 2021-01-01 03:00:00 -1.28666666666667 47.4134782446421  
## 5 2021-01-01 04:00:00 -1.16000000000000 64.6905910495159  
## 6 2021-01-01 05:00:00 -1.03333333333333 36.9628379973113

# Intervention model parameters  
intercept <- 40  
beta\_temp <- 0.5  
beta\_peak <- 12  
beta\_non\_peak <- 10  
  
# Compute the energy consumption  
energy\_consumption <- intercept +  
 beta\_temp \* df\_weather$t\_out +  
 beta\_peak \* peak\_hours +  
 beta\_non\_peak \* non\_peak\_hours +  
 rnorm(length(timestamps), mean = 0, sd = sqrt(100))  
  
# Create a data frame  
df\_interv <- data.frame(datetime = timestamps,  
 t\_out = df\_weather$t\_out,  
 power = energy\_consumption)

df\_base %>%   
 rename(Baseline = power,   
 base\_t\_out = t\_out) %>%   
 left\_join(df\_interv %>% rename(Intervention = power), by = "datetime") %>%   
 pivot\_longer(-c(datetime, base\_t\_out, t\_out), names\_to = "parameter", values\_to = "value") %>%   
 ggplot(aes(x = t\_out, y = value, color = parameter)) +  
 geom\_point(size = 0.1, alpha = 0.1) +  
 geom\_smooth() +  
 scale\_x\_continuous(expand = c(0, 0),   
 limits = c(-15, 35),  
 breaks = breaks\_pretty(n = 5),  
 labels = number\_format(suffix = " °C")) +  
 scale\_y\_continuous(expand = c(0, 0),   
 labels = number\_format(suffix = " kW")) +  
 scale\_color\_manual(values = ls\_colours) +  
 labs(x = NULL,   
 y = NULL,  
 color = NULL,   
 title = "Created baseline and intervention measurements") +  
 theme(panel.grid.major.y = element\_line(colour = "grey80", size = 0.25),  
 legend.direction = "horizontal",  
 legend.position = "bottom",  
 plot.margin = margin(t = 2, r = 7, b = 2, l = 2, unit = "mm"))

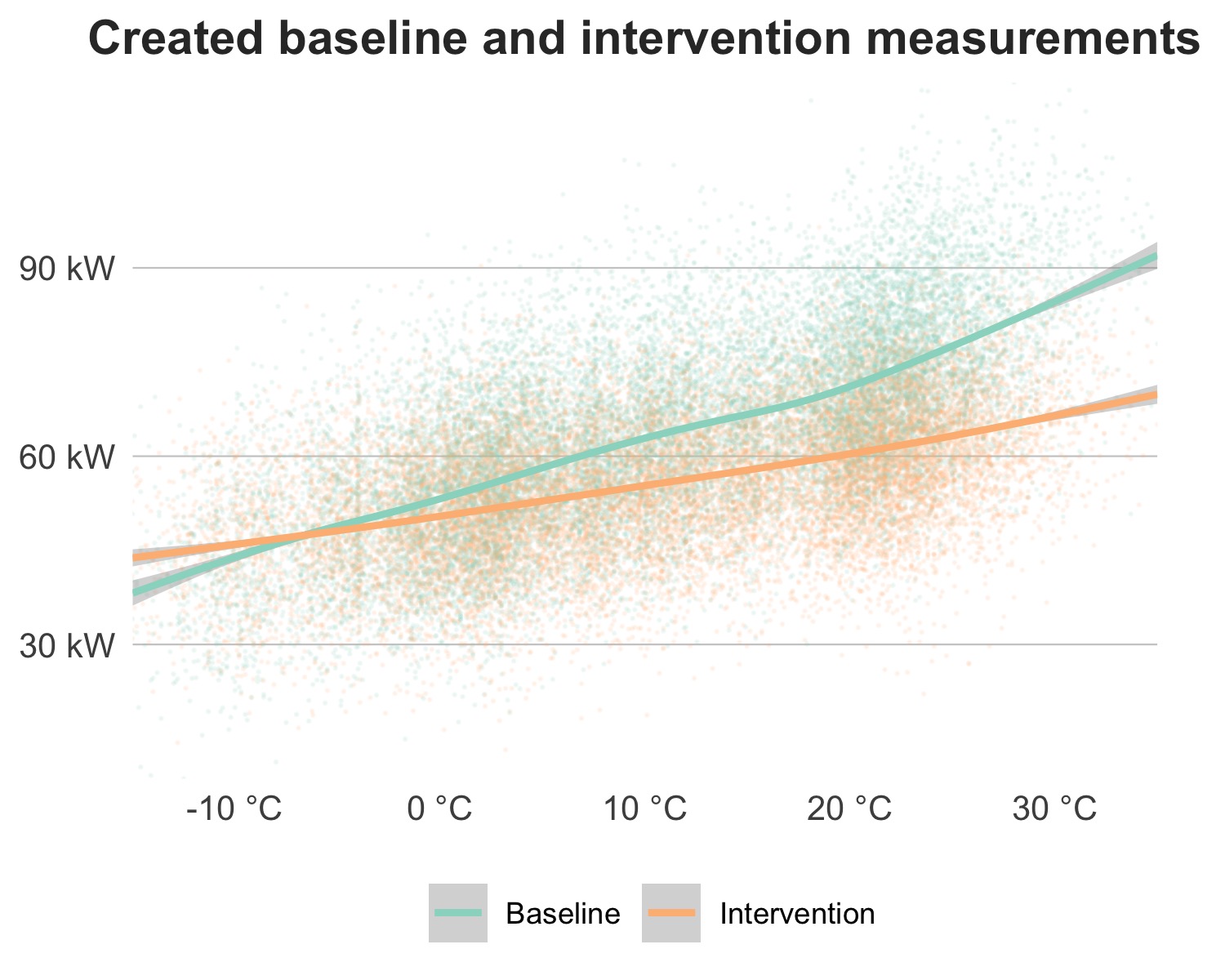


Figure 3.1: Generated hypothetical baseline and intervention measurements using a linear function (data points show hourly energy consumption and line fitted using the loess function).

# 4 Novel randomized M&V analysis

## 4.1 Switchback experimental design

As most control intervention strategies can be implemented interchangeably with the building baseline control, we argue in the paper that randomizing the implementation sequence between intervention and baseline control strategies can balance the impact of confounding variables when estimating the intervention effect, find a result faster, and be more robust to the occurrence of changes in performance that are unrelated to the intervention. A simple way to get started is using a built-in function ’*sample()’* in R package *‘base’* to randomly select at each sampling interval which strategy to implement.

# Define strategies and number of days  
strategies <- c("Base", "Intervention")  
num\_weeks <- 12  
days\_per\_week <- 7  
total\_days <- num\_weeks \* days\_per\_week  
  
# Randomly assign strategies to each day  
assigned\_strategies <- sample(strategies, total\_days, replace = TRUE)  
  
# Convert the linear structure into a weekly structure for easier interpretation  
weekly\_assignment <- matrix(assigned\_strategies, nrow = num\_weeks, ncol = days\_per\_week)  
  
print(weekly\_assignment[1, ])

## [1] "Base" "Base" "Base" "Intervention" "Intervention"  
## [6] "Intervention" "Base"

In statistics, average treatment effect (ATE) estimation depends on the statistical power of unbiased randomization and balanced sample size. This means to accurately estimate the energy-saving effect of the intervention control strategy, we should design an unbiased randomization that has an equal amount of measurements for each level of confounding variables. Therefore, a simple complete randomization can still be biased, for instance, getting a poor randomization where a the baseline, by chance, is sampled more often on weekends and vice versa. This can be avoided by designing a block randomization, which is a type of experimental design that first divides the testing units into blocks and then randomly assigns which strategy to implement for each unit within each block. Thus in practice, each block type corresponds to one confounding variable, and since strategies are sampled at the block level, this means each confounding variable is covered by all testing strategies. For example, building energy consumption is known to be affected by occupancy, time, and outdoor weather conditions. If the control service company wishes to bill the customer based on the savings solely from the control retrofit itself, the M&V analyst should decompose the measured energy savings and adjust for the influence from mild weather or decreased occupancy. For the example in this document, we assume just two strategies are being tested, over a blocking period of 12 weeks.

To do block design, we usde a function in R called *‘blocks()’* from the *‘blocksdesign’* package. We further bonded other functions and inputs to make the package more versatile and one example is shown below. Once successfully called, the function will return a list consisting of schedule output and schedule summaries. The detailed schedule can be viewed by calling the schedule key name.

source(paste0(function\_path, "blocking.R")) # block design schedule generation function  
schedule <- blocking(start\_date = "2021-01-01",  
 n\_weeks = 48,   
 n\_seasons = 4,   
 seed = 390,   
 searches = 20,   
 jumps = 20,   
 treatments = 2,   
 consec = 1)  
  
kable(head(schedule$schedule, n = 14))

| season | date | weekday | strategy |
| --- | --- | --- | --- |
| 1 | 2021-01-01 | Fri | 1 |
| 1 | 2021-01-02 | Sat | 2 |
| 1 | 2021-01-03 | Sun | 1 |
| 1 | 2021-01-04 | Mon | 2 |
| 1 | 2021-01-05 | Tue | 1 |
| 1 | 2021-01-06 | Wed | 1 |
| 1 | 2021-01-07 | Thu | 1 |
| 1 | 2021-01-08 | Fri | 2 |
| 1 | 2021-01-09 | Sat | 1 |
| 1 | 2021-01-10 | Sun | 1 |
| 1 | 2021-01-11 | Mon | 1 |
| 1 | 2021-01-12 | Tue | 1 |
| 1 | 2021-01-13 | Wed | 1 |
| 1 | 2021-01-14 | Thu | 2 |

To check whether the blocking experimental design is balanced, we can call three built-in summary table outputs:

1. Weekday summary: calculates how many days are sampled for each season weekday and strategy combination.
2. Consecutive days summary: calculates how many consecutive days are sampled for each strategy

kable(head(schedule$weekday\_summary))

| season | weekday | strategy | n |
| --- | --- | --- | --- |
| 1 | 1 | 1 | 6 |
| 1 | 1 | 2 | 6 |
| 1 | 2 | 1 | 6 |
| 1 | 2 | 2 | 6 |
| 1 | 3 | 1 | 6 |
| 1 | 3 | 2 | 6 |

kable(head(schedule$consec\_summary))

| strategy | consecutive |
| --- | --- |
| 1 | 80 |
| 2 | 81 |

## 4.2 Sequential analysis of the intervention effect

This section replicates the sequential analysis described in the manuscript using the newly created measurement dataset. The results are shown in figure 4.1.

# source(paste0(function\_path, "saving\_pred.R")) # annual saving estimation function  
source(paste0(function\_path, "ol\_est.R")) # independent variable overlapping calculation  
source(paste0(function\_path, "saving\_pred.R")) # annual saving estimation function  
# Combine random sampling  
df\_schedule <- data.frame(strategy = schedule$schedule$strategy,  
 datetime = schedule$schedule$date)  
df\_base\_sch <- df\_base %>%   
 left\_join(df\_schedule, by = "datetime") %>%   
 fill(strategy, .direction = "down") %>%   
 filter(strategy == 1)  
  
df\_interv\_sch <- df\_interv %>%   
 left\_join(df\_schedule, by = "datetime") %>%   
 fill(strategy, .direction = "down") %>%   
 filter(strategy == 2)  
  
df\_hourly <- bind\_rows(df\_base\_sch, df\_interv\_sch) %>%   
 arrange(datetime)  
  
# make daily totals  
df\_daily <- df\_hourly %>%  
 group\_by(datetime = floor\_date(datetime, unit = "day")) %>%  
 summarise(strategy = unique(strategy),  
 power\_ave = mean(power, na.rm = TRUE),  
 power\_peak = max(power, na.rm = TRUE),  
 t\_out = mean(t\_out, na.rm = TRUE)) %>%  
 ungroup()

# prepare sequential test dataset  
sprt\_hourly <- df\_hourly %>%  
 mutate(week = interval(min(datetime), datetime) %>% as.numeric('weeks') %>% floor())  
  
# set analysis parameters  
param\_sprt <- list(baseline = "Baseline",  
 strategy = "Intervention",  
 parameter = "power\_ave", # choice: power\_ave; mmoer; co2  
 label = "power", # choice: power; emissions  
 n\_weeks = 48)  
  
# prepare dataframe for analysis  
df\_sprt <- df\_daily %>%  
 mutate(strategy = as.factor(strategy),  
 strategy = recode\_factor(strategy, "1" = "Baseline", "2" = "Intervention")) %>%  
 filter(strategy %in% c(param\_sprt$baseline, param\_sprt$strategy)) %>%  
 pivot\_longer(cols = -c(datetime, strategy),   
 names\_to = "parameter",   
 values\_to = "value") %>%  
 mutate(week = interval(min(datetime), datetime) %>% as.numeric('weeks') %>% floor(),  
 value = value) %>%  
 filter(str\_detect(parameter, param\_sprt$parameter)) %>%  
 droplevels()  
  
  
# define list to store weekly means  
df\_means <- list()  
  
# calculate weekly means  
for (i in 1:param\_sprt$n\_weeks) {  
   
 # subset data by week  
 df\_means[[i]] <- df\_sprt %>%  
 filter(week <= i) %>%  
 group\_by(strategy,   
 parameter) %>%  
 summarise(week = i,  
 value\_ave = mean(value, na.rm = TRUE),  
 value\_sd = sd(value, na.rm = TRUE),   
 .groups = "keep") %>%  
 ungroup()  
  
}  
  
# combine list into df  
df\_means <- bind\_rows(df\_means)  
  
# define lists to store stopping criteria results  
## SPRT results  
sprt\_res <- list()  
  
## 80% independent variable  
overlap\_base <- list()  
overlap\_s2 <- list()  
quantile\_tmy <- quantile(df\_tmy$temp, na.rm = TRUE, c(0, 0.99))  
  
## annual saving estimation  
annual\_saving <- list()  
  
# do sample  
for (i in 2:param\_sprt$n\_weeks) {  
   
 # subset data by week  
 df\_seq <- df\_sprt %>%  
 filter(week <= i) %>%  
 droplevels()  
   
 # Calculate overlapping temperature range  
 overlap\_base[[i]] <- tibble("n\_weeks" = i,   
 overlap\_base = sprt\_hourly %>%  
 filter(week <= i & strategy == 1) %>%   
 ol\_est(., quantile\_tmy))  
   
 overlap\_s2[[i]] <- tibble("n\_weeks" = i,   
 overlap\_s2 = sprt\_hourly %>%  
 filter(week <= i & strategy == 2) %>%   
 ol\_est(., quantile\_tmy))  
   
 # start estimation after 2 months for complete ftow profile  
 if (i/4 > 2) {  
   
 # Update user on the week of update  
 # print(paste0("updating TOWT model in week ", i))  
 annual\_saving[[i]] <- tibble("n\_weeks" = i,  
 annual\_saving = saving\_pred(sprt\_hourly %>% filter(week <= i)))  
  
 }  
   
 # do test  
 results\_seq <- seq\_ttest(x = value ~ strategy,   
 data = df\_seq,  
 d = 0.5,   
 power = 0.9,   
 alternative = "less", # greater  
 paired = FALSE,  
 verbose = TRUE)  
   
 # calculate effect size  
 results\_ci <- effsize::cohen.d(d = df\_seq$value,  
 f = df\_seq$strategy,  
 conf.level = 0.90,  
 paired = FALSE,  
 na.rm = TRUE)  
   
 # update user  
 # print(paste0("Calculating effect size for week ", i))  
   
 # bootstrap effect size  
 results\_bs <- bootES::bootES(data = df\_seq, R = 1000,   
 contrast = c(param\_sprt$baseline, param\_sprt$strategy),  
 data.col = "value", group.col = "strategy")  
   
 # extract test statistic  
 sprt\_res[[i]] <- tibble("n\_weeks" = i,   
 "threshold\_lower" = exp(results\_seq@B\_boundary\_log),  
 "threshold\_upper" = exp(results\_seq@A\_boundary\_log),  
 "statistic" = results\_seq@likelihood\_ratio,  
 "decision" = results\_seq@decision,  
 "cohens\_d" = round(results\_ci$estimate, digits = 2),  
 "ci\_low" = round(results\_ci$conf.int[[1]], digits = 2),  
 "ci\_high" = round(results\_ci$conf.int[[2]], digits = 2),  
 "ns\_stat" = round(results\_bs$t0, digits = 2),  
 "ns\_ci\_low" = round(results\_bs$bounds[[1]], digits = 2),  
 "ns\_ci\_high" = round(results\_bs$bounds[[2]], digits = 2))  
   
}   
  
# combine list and vectors into df  
annual\_saving <- bind\_rows(annual\_saving)  
  
# work out when threshold is reached  
sprt\_res <- bind\_rows(sprt\_res)  
  
sprt\_res <- sprt\_res %>%  
 left\_join(., annual\_saving, by = "n\_weeks") %>%  
 mutate(flag = ifelse(str\_detect(decision, "accept"), 1, 0))  
  
sprt\_overlap\_base <- bind\_rows(overlap\_base) %>%   
 left\_join(., annual\_saving, by = "n\_weeks") %>%   
 mutate(flag = ifelse(overlap\_base >= 0.8, 1, 0),  
 flag = ifelse(flag == lag(flag, 1), 0, flag))  
  
sprt\_overlap\_s2 <- bind\_rows(overlap\_s2) %>%   
 left\_join(., annual\_saving, by = "n\_weeks") %>%   
 mutate(flag = ifelse(overlap\_s2 >= 0.8, 1, 0),  
 flag = ifelse(flag == lag(flag, 1), 0, flag))

# Baseline prediction  
# Specifying model input information  
model\_input\_options <- nmecr::assign\_model\_inputs(regression\_type = "TOWT", occupancy\_threshold = 0.65)  
  
baseline\_train <- df\_base %>%  
 filter(datetime < as.Date("2022-01-01")) %>%   
 rename(time = datetime,   
 eload = power,   
 temp = t\_out)  
  
# do baseline model to estimate occupancy  
towt\_base <- nmecr::model\_with\_TOWT(training\_data = baseline\_train,  
 model\_input\_options = model\_input\_options)  
  
# Add occupancy info to model specification  
base\_model\_input\_options <- map(model\_input\_options, ~.) %>%   
 append(list(find\_occ\_unocc(training\_data = baseline\_train, model\_input\_options = towt\_base$model\_input\_options))) %>%   
 set\_names(., c(names(model\_input\_options), "occupancy\_info"))  
  
# re-estimate baseline with occupancy  
towt\_base <- nmecr::model\_with\_TOWT(training\_data = baseline\_train,  
 model\_input\_options = base\_model\_input\_options)  
df\_towt <- df\_weather %>%   
 filter(datetime >= as.Date("2022-01-01")) %>%   
 rename(time = datetime,   
 temp = t\_out) %>%   
 mutate(eload = NA)  
  
baseline\_results <- model\_pred(amod = towt\_base$model\_occupied,   
 bmod = towt\_base$model\_unoccupied,  
 time = df\_towt$time,   
 temp = df\_towt$temp,  
 temp\_knots = towt\_base$model\_input\_options$calculated\_temp\_knots,   
 dataframe = df\_towt,  
 occ\_info = towt\_base$model\_input\_options$occupancy\_info,  
 interval\_minutes = towt\_base$model\_input\_options$interval\_minutes) %>%  
 rename("datetime" = "time") %>%   
 mutate("datetime" = force\_tz(datetime, tzone = "America/Chicago"))  
  
conv\_saving <- baseline\_results %>%   
 left\_join(df\_interv %>% filter(datetime >= as.Date("2022-01-01")), by = "datetime") %>%   
 mutate(saving = power - towt) %>%   
 mutate(n\_weeks = interval(min(datetime), datetime) %>% as.numeric('weeks') %>% floor()) %>%  
 filter(n\_weeks <= 48)

p2 <- ggplot(sprt\_res, aes(x = n\_weeks, y = ns\_stat)) +  
 geom\_ribbon(aes(ymin = ns\_ci\_low, ymax = ns\_ci\_high), alpha = 0.5, fill = "#D0E3F1") +  
 geom\_line(size = 1.25, colour = "grey20", alpha = 0.4) +  
 geom\_hline(yintercept = mean(conv\_saving %>% .$saving, na.rm = T),   
 linetype = "dashed",   
 color = "black") +  
 annotate(geom = "text",  
 x = 5,  
 y = -11,   
 size = 3,  
 label = "conventional method\nestimation\n(2 years)") +  
 geom\_text(data = sprt\_res[2, ],   
 aes(x = n\_weeks, y = ns\_ci\_high, label = "Upper 90% CI"),  
 position = position\_nudge(x = 1),   
 colour = "grey20",   
 size = 2.5,   
 fontface = "italic") +  
 geom\_text(data = sprt\_res[2, ],   
 aes(x = n\_weeks, y = ns\_ci\_low, label = "Lower 90% CI"),  
 position = position\_nudge(x = 1),   
 colour = "grey20",   
 size = 2.5,   
 fontface = "italic") +  
 geom\_point(data = sprt\_res %>% filter(n\_weeks%%12 == 0),   
 size = 8,   
 shape = 16,   
 colour = "#347EB3",   
 alpha = 0.5) +  
 geom\_point(data = sprt\_res %>% filter(n\_weeks%%12 == 0),   
 size = 4,   
 shape = 16,   
 colour = "#347EB3",   
 alpha = 0.9) +  
 geom\_text(data = sprt\_res %>% filter(n\_weeks%%12 == 0),   
 aes(x = n\_weeks, y = ns\_stat, label = paste0(round(ns\_stat, 1L), " kW")),   
 position = position\_nudge(x = 2.5),   
 colour = "grey20",   
 size = 3.0,   
 fontface = "italic") +  
 geom\_errorbar(data = sprt\_res %>% filter(n\_weeks%%12 == 0),   
 aes(x = n\_weeks, ymin = ns\_ci\_low, ymax = ns\_ci\_high),   
 width = 1,   
 colour = "#347EB3",   
 alpha = 0.5,   
 size = 0.8) +  
 geom\_text(data = sprt\_res %>% filter(n\_weeks%%12 == 0),   
 aes(x = n\_weeks, y = ns\_ci\_high, label = paste0(round(ns\_ci\_high, 1L), " kW")),   
 position = position\_nudge(x = 2.5),   
 colour = "grey20",   
 size = 2.5,   
 fontface = "italic") +  
 geom\_text(data = sprt\_res %>% filter(n\_weeks%%12 == 0),   
 aes(x = n\_weeks, y = ns\_ci\_low, label = paste0(round(ns\_ci\_low, 1L), " kW")),   
 position = position\_nudge(x = 2.5),   
 colour = "grey20",   
 size = 2.5,   
 fontface = "italic") +  
 geom\_text(data = sprt\_res %>% filter(n\_weeks%%12 == 0 & flag == 1),  
 aes(x = n\_weeks, y = -15, label = "Difference\nfound"),   
 colour = "grey20",   
 size = 3.0,   
 fontface = "italic") +  
 geom\_text(data = slice\_tail(sprt\_res, n = 1),   
 aes(x = n\_weeks, y = ns\_stat, label = paste0(round(ns\_stat, 1L), " kW")),   
 position = position\_nudge(x = 0.5),   
 colour = "grey20",   
 size = 3.0,   
 check\_overlap = TRUE,   
 hjust = 0) +  
 geom\_text(data = slice\_tail(sprt\_res, n = 1),   
 aes(x = ifelse(is.na(first(flag)), 0, n\_weeks),   
 y = ns\_ci\_high,   
 label = ifelse(is.na(first(flag)), NULL, paste0(round(ns\_ci\_high, 1L), " kW"))),   
 position = position\_nudge(x = 0.5),   
 colour = "grey20",   
 size = 2.5,   
 check\_overlap = TRUE,   
 hjust = 0) +  
 geom\_text(data = slice\_tail(sprt\_res, n = 1),   
 aes(x = ifelse(is.na(first(flag)), 0, n\_weeks),   
 y = ns\_ci\_low,   
 label = ifelse(is.na(first(flag)), NULL, paste0(round(ns\_ci\_low, 1L), " kW"))),   
 position = position\_nudge(x = 0.5),   
 colour = "grey20",   
 size = 2.5,   
 check\_overlap = TRUE,   
 hjust = 0) +  
 geom\_vline(xintercept = seq(0, 50, by = 12),   
 linetype = "dashed",   
 color = "grey20",   
 alpha = 0.3,   
 size = 0.5) +  
 annotate(geom = "text",   
 x = seq(5, 50, by = 12),   
 y = 2.5,   
 label = paste0("12-week block"),   
 alpha = 0.5,   
 size = 3) +  
 scale\_x\_continuous(expand = c(0, 0),   
 limits = c(1, param\_sprt$n\_weeks + 0.5),  
 labels = number\_format(accuracy = 1L, suffix = "\nweeks")) +  
 scale\_y\_continuous(expand = c(0, 0),   
 limits = c(-15, 2.5),  
 breaks = pretty\_breaks(n = 3),  
 labels = number\_format(suffix = " kW")) +  
 labs(title = NULL,   
 subtitle = "SPRT results and estimated difference in power consumption without weather normalization",  
 x = NULL,   
 y = NULL) +  
 guides(alpha = "none") +  
 coord\_cartesian(clip = "off") +  
 theme(panel.grid.major.y = element\_line(colour = "grey80", size = 0.25),  
 axis.text.x = element\_blank(),  
 plot.margin = margin(3, 15, 3, 3, unit = "mm"))  
  
p4 <- ggplot(sprt\_overlap\_base) +  
 geom\_line(aes(x = n\_weeks, y = overlap\_base, color = "Baseline"),   
 size = 1.25) +  
 geom\_line(data = sprt\_overlap\_s2,   
 aes(x = n\_weeks, y = overlap\_s2, color = "Intervention"),  
 size = 1.25) +  
 geom\_text(data = sprt\_overlap\_base[nrow(sprt\_overlap\_base), ],  
 aes(x = n\_weeks, y = 0.8, label = "80% of\nTMY range\nthreshold"),  
 position = position\_nudge(x = 0.5),   
 color = "red",   
 size = 3.0,  
 check\_overlap = TRUE,  
 hjust = 0) +  
 geom\_hline(yintercept = 0.8, linetype = "dashed", color = "red") +  
 scale\_color\_manual(values = ls\_colours) +  
 scale\_x\_continuous(expand = c(0, 0),   
 limits = c(1, param\_sprt$n\_weeks + 0.5),  
 labels = number\_format(accuracy = 1L, suffix = "\nweeks")) +  
 scale\_y\_continuous(breaks = seq(0.2, 1.0, by = 0.2),   
 labels = c("20%", "40%", "60%", "80%", "100%")) +  
 labs(title = NULL,   
 subtitle = "Confounding variable (outdoor drybulb temperature) range coverage",  
 x = NULL,   
 y = NULL) +  
 coord\_cartesian(clip = "off") +  
 theme(panel.grid.major.y = element\_line(),  
 legend.position = "none",  
 plot.margin = margin(3, 15, 3, 3, unit = "mm"))  
  
# do running mean plot  
p1 <- ggplot(df\_means, aes(x = week, y = value\_ave)) +  
 geom\_line(aes(colour = strategy), size = 1.25) +  
 geom\_text(data = filter(df\_means,   
 week == param\_sprt$n\_weeks,   
 parameter == param\_sprt$parameter),   
 aes(x = param\_sprt$n\_weeks,   
 y = value\_ave,   
 colour = strategy,   
 label = strategy),   
 position = position\_nudge(x = 0.5),   
 size = 3.0,   
 check\_overlap = TRUE,   
 hjust = 0) +  
 scale\_x\_continuous(expand = c(0, 0),   
 limits = c(1, param\_sprt$n\_weeks + 0.5)) +  
 scale\_y\_continuous(expand = c(0, 0),   
 breaks = breaks\_pretty(n = 4),  
 labels = number\_format(suffix = " kW")) +  
 scale\_color\_manual(values = ls\_colours) +  
 labs(title = NULL,   
 subtitle = "Running average power consumption of the case study building",  
 x = NULL,   
 y = NULL) +  
 guides(alpha = "none", colour = "none") +  
 coord\_cartesian(clip = "off") +  
 theme(panel.grid.major.y = element\_line(colour = "grey80", size = 0.25),  
 axis.text.x = element\_blank(),  
 plot.margin = margin(3, 15, 3, 3, unit = "mm"))  
  
p3 <- ggplot(annual\_saving) +  
 geom\_line(aes(x = n\_weeks, y = annual\_saving),   
 size = 1.25, color="darkgrey") +  
 geom\_point(data = annual\_saving[1, ],   
 aes(x = n\_weeks, y = annual\_saving),   
 size = 4,  
 alpha = 0.8,   
 shape = 16,   
 colour = "#347EB3") +  
 geom\_text(data = annual\_saving[1, ],   
 aes(x = n\_weeks - 3,   
 y = annual\_saving,   
 label = paste0("Estimation start\nweek ", n\_weeks)),   
 colour = "grey20", size = 2.5, fontface = "italic") +  
 geom\_point(data = sprt\_res %>% filter(flag == 1),   
 aes(x = ifelse(is.null(first(flag)), max(sprt\_res$n\_weeks), n\_weeks),   
 y = ifelse(is.null(first(flag)), max(sprt\_res$annual\_saving), annual\_saving)),   
 size = 4,  
 alpha = 0.8,   
 shape = 16,   
 colour = "#347EB3") +  
 geom\_point(data = sprt\_overlap\_base %>% filter(flag == 1),   
 aes(x = ifelse(is.null(first(flag)), max(sprt\_overlap\_base$n\_weeks), n\_weeks),   
 y = ifelse(is.null(first(flag)), max(sprt\_overlap\_base$annual\_saving), annual\_saving)),   
 size = 4,  
 alpha = 0.8,   
 shape = 16,   
 colour = "#347EB3") +  
 geom\_text(data = sprt\_overlap\_base %>% filter(flag == 1),   
 aes(x = ifelse(is.null(first(flag)), max(sprt\_overlap\_base$n\_weeks), n\_weeks),   
 y = ifelse(is.null(first(flag)), max(sprt\_overlap\_base$annual\_saving), annual\_saving) + 4,   
 label = paste0("Temperature check\n(Baseline)\nweek ", n\_weeks)),   
 colour = "grey20", size = 2.5, fontface = "italic") +  
 geom\_point(data = sprt\_overlap\_s2 %>% filter(flag == 1),   
 aes(x = ifelse(is.null(first(flag)), max(sprt\_overlap\_s2$n\_weeks), n\_weeks),   
 y = ifelse(is.null(first(flag)), max(sprt\_overlap\_s2$annual\_saving), annual\_saving)),   
 size = 4,  
 alpha = 0.8,   
 shape = 16,   
 colour = "#347EB3") +  
 geom\_text(data = sprt\_overlap\_s2 %>% filter(flag == 1),   
 aes(x = ifelse(is.null(first(flag)), max(sprt\_overlap\_s2$n\_weeks), n\_weeks),   
 y = ifelse(is.null(first(flag)), max(sprt\_overlap\_s2$annual\_saving), annual\_saving) - 2,   
 label = paste0("Temperature check\n(Intervention)\nweek ", n\_weeks)),   
 colour = "grey20", size = 2.5, fontface = "italic") +  
 geom\_point(data = annual\_saving %>% filter(n\_weeks == 12),   
 aes(x = n\_weeks, y = annual\_saving),   
 size = 4,  
 alpha = 0.8,   
 shape = 16,   
 colour = "#347EB3") +  
 geom\_text(data = annual\_saving %>% filter(n\_weeks == 12),  
 aes(x = n\_weeks, y = annual\_saving + 3,label = paste0("Minimum\ntest length check\nweek ", n\_weeks)),  
 colour = "grey20",   
 size = 2.5,   
 fontface = "italic")+  
 geom\_point(data = annual\_saving %>% filter(n\_weeks == 24),   
 aes(x = n\_weeks, y = annual\_saving),   
 size = 4,  
 alpha = 0.8,   
 shape = 16,   
 colour = "#347EB3") +  
 geom\_text(data = annual\_saving %>% filter(n\_weeks == 24),  
 aes(x = n\_weeks, y = annual\_saving + 3, label = paste0("End of block period\nall stopping criteria met\nweek ", n\_weeks)),  
 colour = "grey20",   
 size = 2.5,   
 fontface = "italic")+  
 geom\_text(data = sprt\_res %>% filter(n\_weeks%%12 == 0 & flag == 1) %>% slice(1),  
 aes(x = n\_weeks, y = annual\_saving - 2, label = paste0("SPRT check\nweek ", n\_weeks)),   
 colour = "grey20",   
 size = 2.5,   
 fontface = "italic") +  
 geom\_vline(xintercept = seq(0, 50, by = 12),   
 linetype = "dashed",   
 color = "grey20",   
 alpha = 0.3,   
 size = 0.5) +  
 annotate(geom = "text",   
 x = seq(3, 50, by = 12),   
 y = 22,   
 label = paste0("12-week block"),   
 alpha = 0.5,   
 size = 2.5) +  
 geom\_text(data = annual\_saving[nrow(annual\_saving), ],   
 aes(x = n\_weeks, y = annual\_saving, label = paste0(round(annual\_saving, 0), "%")),   
 position = position\_nudge(x = 0.5),   
 colour = "grey20",   
 size = 3.0,   
 check\_overlap = TRUE,   
 hjust = 0) +  
 scale\_x\_continuous(expand = c(0, 0),   
 limits = c(1, param\_sprt$n\_weeks + 0.5),  
 labels = number\_format(accuracy = 1L, suffix = "\nweeks")) +  
 scale\_y\_continuous(breaks = seq(0, 20, by = 5),   
 labels = c("0%", "5%", "10%", "15%", "20%")) +  
 labs(title = NULL,   
 subtitle = "Normalized annual energy savings estimate using TOWT model predictions for a typical meteorological year",  
 x = NULL,   
 y = NULL) +  
 coord\_cartesian(clip = "off") +  
 theme(panel.grid.major.y = element\_line(),  
 plot.margin = margin(3, 15, 3, 3, unit = "mm"))  
  
# Overall result  
p1 / p2 / p4 / p3 +  
 plot\_annotation(title = "Overall sequential evaluation results using pre-defined stopping criteria") +  
 plot\_annotation(tag\_levels = c('a'), tag\_suffix = ')') &  
 theme(plot.tag.position = c(0, 1),  
 plot.tag = element\_text(color="black"))

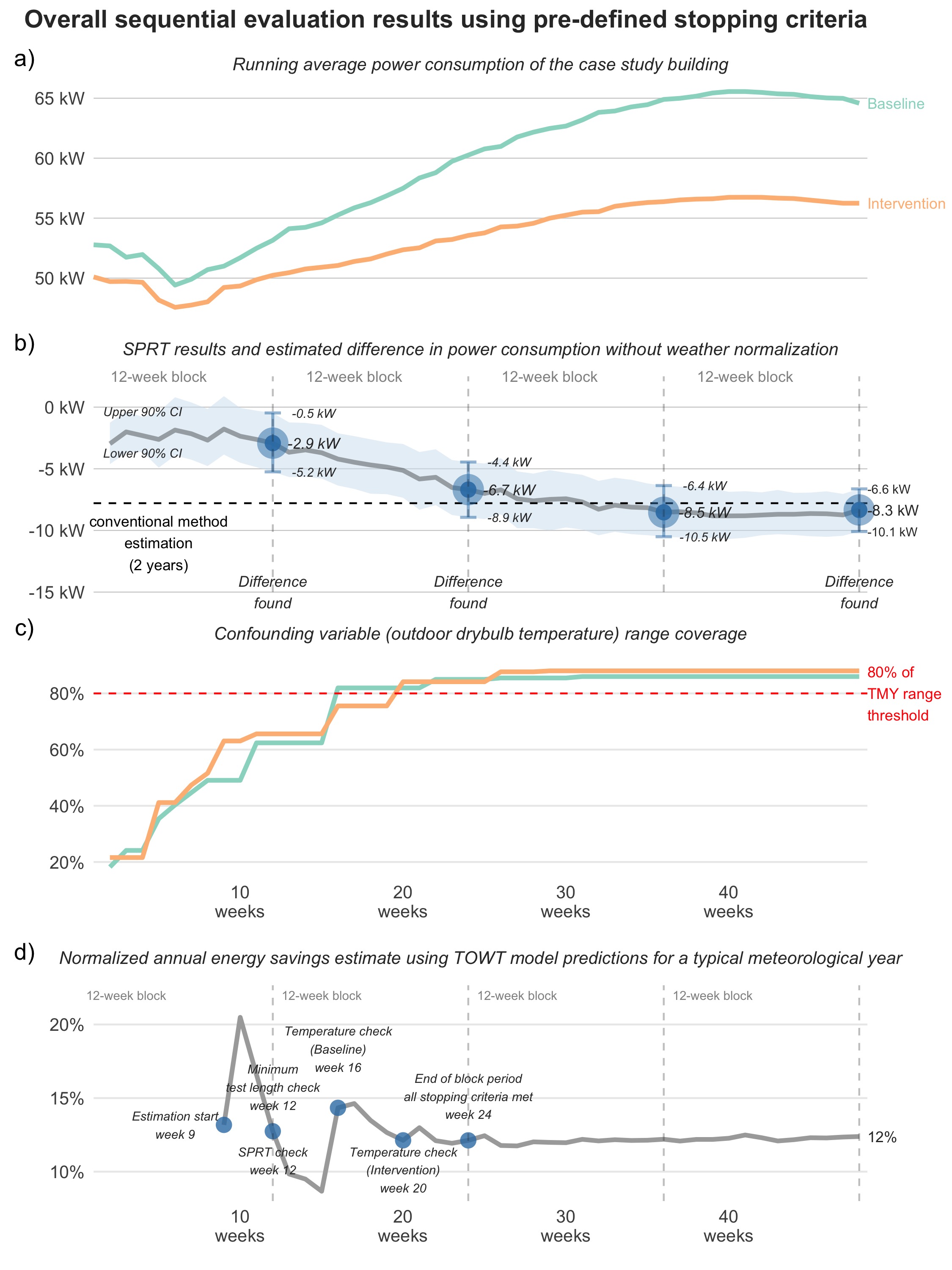


Figure 4.1: Comprehensive results summary of the proposed M&V method applied to the case study throughout a total of 60 weeks

As figure 4.1 indicates, the running average of power consumption in subplot a) shows more savings from the intervention strategy during warmer outdoor weather in general. This also corresponds to the sequential probability ratio test (SPRT) results shown in subplot b). The difference between baseline measurements and intervention measurements was detected early in week 12. However, subplot c) indicates that outdoor weather has not reached the 80% threshold range calculated with reference to the typical meteorological year (TMY) weather file, we need to continue the M&V. At week 16, the accumulated temperature measured in sampled baseline days covered 80% range and at week 24, sampled intervention days satisfied the criterion. Therefore, we report a 6.6 kW savings at the end of this blocking period (i.e. week 24). If the analyst decides to continue sampling, the plot further shows the uncertainty associated with saving estimation further decreased throughout the remaining 24 weeks. Subplot d) plots the sequentially estimated normalized savings on the TMY weather conditions and outlines the check-point for satisfying each critical stopping criterion. It shows although the saving was detected early by the statistical test at week 12, the normalized annual estimation still largely fluctuates at the point. After covered sufficient weather conditions, the annualized savings stabilize at 12%. If this is a real project, and the analyst follows the steps outlined in the standards, week 24 is merely the mid-point of baseline collection assuming no interruptions or delays and he/she has to wait for another one year and half to estimate savings. Subplot b) also shows the saving estimated from a conventional 2-year pre-/post- type M&V, which is similar to the results obtained at week 24 and onward if the analyst applies the novel randomized M&V.

# 5 Comparison between conventional M&V method

In this section, we demonstrate how random sampling between baseline and intervention strategy is more reliable than the conventional M&V method. Building performance frequently changes in response to changes in how the building is occupied or operated. The changes - those that are unrelated to the intervention being evaluated - cause the savings estimate to be less accurate, even if the modeling accuracy satisfies the minimum requirements from the standards. Sampling randomly, and frequently, throughout the entire distribution reduces the impact of these events on the reliability of the savings estimate. We highlight this by exploring three different scenarios.

## 5.1 Scenario A: A discrete non-routine event

One common and typical non-routine event in an energy-saving M&V project is a change in occupancy, or a substantial change in occupant behavior. This is because occupancy has a substantial influence on building energy consumption but is not related to the purpose of testing the intervention control. In reality, it is unusal to have occupancy measurements (compared to other independent variables such as outdoor temperature and therefore) when a analyst fits a regression model (e.g. TOWT), thus, the analyst normally assumes that occupancy remains unchanged throughout the study. ASHRAE G14 and IPMVP only require the analyst to check for the coefficient of the variation of the root mean square error (CVRMSE) and normalized mean square error (NMSE) of a fitted baseline model. Thus, if occupancy changes after the retrofit implementation, the analyst can still obtain an ‘accurate’ baseline model with qualified CV(RMSE) but it is biased due to the unrelated change event being overlooked. The novel M&V method, on the other hand, resolves this by sampling randomly and frequently throughout the full distribution of occupancy over the full study period.

Consider a scenario (Scenario A), where one tenant in a high rise office building moves out during the M&V study period, vacating one floor of the building. We expect the impact of non-routine events such as these to be balanced in sampled baseline days and intervention days if we sample randomly with equal probability. This means if occupancy is notably different over say, a 4 week period, the randomized M&V will sample approximately 2 weeks with baseline strategy and 2 weeks with intervention strategy. Therefore, the energy savings can still represent the intervention’s true effect. Under conventional M&V, either the baseline or the intervention estimate will be affected by this difference in occupancy.

To illustrate one example of this scenario, we model a simplified situation where the 12-month baseline occurs under stable occupancy, and after the retrofit implementation, the energy consumption decreased by 10 kW for each metered interval from April to September because of decreased occupancy (i.e. tenants moving out leaving one floor vacant) and that the analyst was not aware of this change in occupancy.

# Generate schedule for the post-intervention period  
schedule <- blocking(start\_date = "2022-01-01",  
 n\_weeks = 60,   
 n\_seasons = 5,   
 seed = 390,   
 searches = 20,   
 jumps = 20,   
 treatments = 2,   
 consec = 1)  
  
df\_schedule <- data.frame(strategy = schedule$schedule$strategy,  
 datetime = schedule$schedule$date)  
  
df\_base\_sch <- df\_base %>%   
 left\_join(df\_schedule, by = "datetime") %>%   
 fill(strategy, .direction = "down") %>%  
 filter(datetime >= as.Date("2022-01-01"))   
  
base\_pre <- df\_base %>%   
 filter(datetime < as.Date("2022-01-01"))  
  
df\_interv\_sch <- df\_interv %>%   
 left\_join(df\_schedule, by = "datetime") %>%   
 fill(strategy, .direction = "down") %>%  
 filter(datetime >= as.Date("2022-01-01"))   
  
# Apply occupancy changes  
df\_base\_alt <- df\_base\_sch %>%  
 mutate(power = ifelse(month(datetime) >= 4 & month(datetime) < 10, power - 10, power))  
  
df\_interv\_alt <- df\_interv\_sch %>%   
 mutate(power = ifelse(month(datetime) >= 4 & month(datetime) < 10, power - 10, power))  
  
df\_hourly\_sch <- bind\_rows(df\_base\_alt %>% filter(strategy == 1), df\_interv\_alt %>% filter(strategy == 2)) %>%   
 arrange(datetime) %>%   
 mutate(power = ifelse(month(datetime) >= 4 & month(datetime) < 10, power - 10, power))  
  
# Baseline projection  
model\_input\_options <- nmecr::assign\_model\_inputs(regression\_type = "TOWT", occupancy\_threshold = 0.65)  
  
baseline\_train <- base\_pre %>%  
 select(time = datetime,   
 eload = power,   
 temp = t\_out)  
  
# do baseline model to estimate occupancy  
towt\_base <- nmecr::model\_with\_TOWT(training\_data = baseline\_train,  
 model\_input\_options = model\_input\_options)  
  
# Add occupancy info to model specification  
base\_model\_input\_options <- map(model\_input\_options, ~.) %>%   
 append(list(find\_occ\_unocc(training\_data = baseline\_train, model\_input\_options = towt\_base$model\_input\_options))) %>%   
 set\_names(., c(names(model\_input\_options), "occupancy\_info"))  
  
# re-estimate baseline with occupancy  
towt\_base <- nmecr::model\_with\_TOWT(training\_data = baseline\_train,  
 model\_input\_options = base\_model\_input\_options)  
  
baseline\_results <- model\_pred(amod = towt\_base$model\_occupied,   
 bmod = towt\_base$model\_unoccupied,  
 time = df\_towt$time,   
 temp = df\_towt$temp,  
 temp\_knots = towt\_base$model\_input\_options$calculated\_temp\_knots,   
 dataframe = df\_towt,  
 occ\_info = towt\_base$model\_input\_options$occupancy\_info,  
 interval\_minutes = towt\_base$model\_input\_options$interval\_minutes) %>%  
 rename("datetime" = "time") %>%   
 mutate("datetime" = force\_tz(datetime, tzone = "America/Chicago"))

# Visualization  
  
p1 <- ggplot() +  
 geom\_point(data = base\_pre %>%   
 slice\_sample(n = 500),   
 aes(x = datetime, y = power, color = "Measured baseline"),   
 size = 0.5,   
 alpha = 0.3) +  
 geom\_smooth(data = base\_pre,   
 aes(x = datetime, y = power, color = "Measured baseline")) +  
 scale\_x\_datetime(date\_breaks = "2 months",  
 date\_labels = "%b") +  
 scale\_y\_continuous(expand = c(0, 0),   
 breaks = breaks\_pretty(n = 3),  
 labels = number\_format(suffix = " kW")) +  
 coord\_cartesian(clip = "on", ylim = c(30, 90)) +  
 scale\_color\_manual(values = ls\_colours) +  
 labs(x = NULL,   
 y = NULL,  
 color = NULL,   
 title = NULL,  
 subtitle = "Pre-retrofit period") +  
 theme(panel.grid.major.y = element\_line(colour = "grey80", size = 0.25),  
 legend.direction = "horizontal",  
 legend.position = "bottom",  
 plot.margin = margin(t = 2, r = 7, b = 2, l = 2, unit = "mm"))  
  
p2 <- ggplot() +  
 geom\_point(data = baseline\_results %>%   
 slice\_sample(n = 500),  
 aes(x = datetime, y = towt, color = "Projected baseline\n(no change)"),   
 size = 0.5,   
 alpha = 0.3) +  
 geom\_smooth(data = baseline\_results,   
 aes(x = datetime, y = towt, color = "Projected baseline\n(no change)")) +  
 geom\_point(data = df\_base\_alt %>%   
 slice\_sample(n = 500),   
 aes(x = datetime, y = power, color = "Adjusted baseline"),   
 size = 0.5,   
 alpha = 0.3) +  
 geom\_smooth(data = df\_base\_alt,   
 aes(x = datetime, y = power, color = "Adjusted baseline")) +  
 geom\_point(data = df\_interv\_alt %>%   
 slice\_sample(n = 500),  
 aes(x = datetime, y = power, color = "Measured interv"),   
 size = 0.5,   
 alpha = 0.3) +  
 geom\_smooth(data = df\_interv\_alt,   
 aes(x = datetime, y = power, color = "Measured interv")) +  
 scale\_x\_datetime(date\_breaks = "2 months",  
 date\_labels = "%b") +  
 scale\_y\_continuous(expand = c(0, 0),   
 breaks = breaks\_pretty(n = 3),  
 labels = number\_format(suffix = " kW")) +  
 coord\_cartesian(clip = "on", ylim = c(30, 90)) +  
 scale\_color\_manual(values = ls\_colours) +  
 labs(x = NULL,   
 y = NULL,  
 color = NULL,   
 title = NULL,  
 subtitle = "post-retrofit period") +  
 theme(panel.grid.major.y = element\_line(colour = "grey80", size = 0.25),  
 axis.text.y = element\_blank(),  
 legend.direction = "horizontal",  
 legend.position = "bottom",  
 plot.margin = margin(t = 2, r = 7, b = 2, l = 2, unit = "mm"))  
  
p1 + p2 +  
 plot\_annotation(title = "Building energy consumption over a 2-year period") +  
 plot\_annotation(tag\_levels = c('a'), tag\_suffix = ')') &  
 theme(plot.tag.position = c(0, 1),  
 plot.tag = element\_text(color="black"))

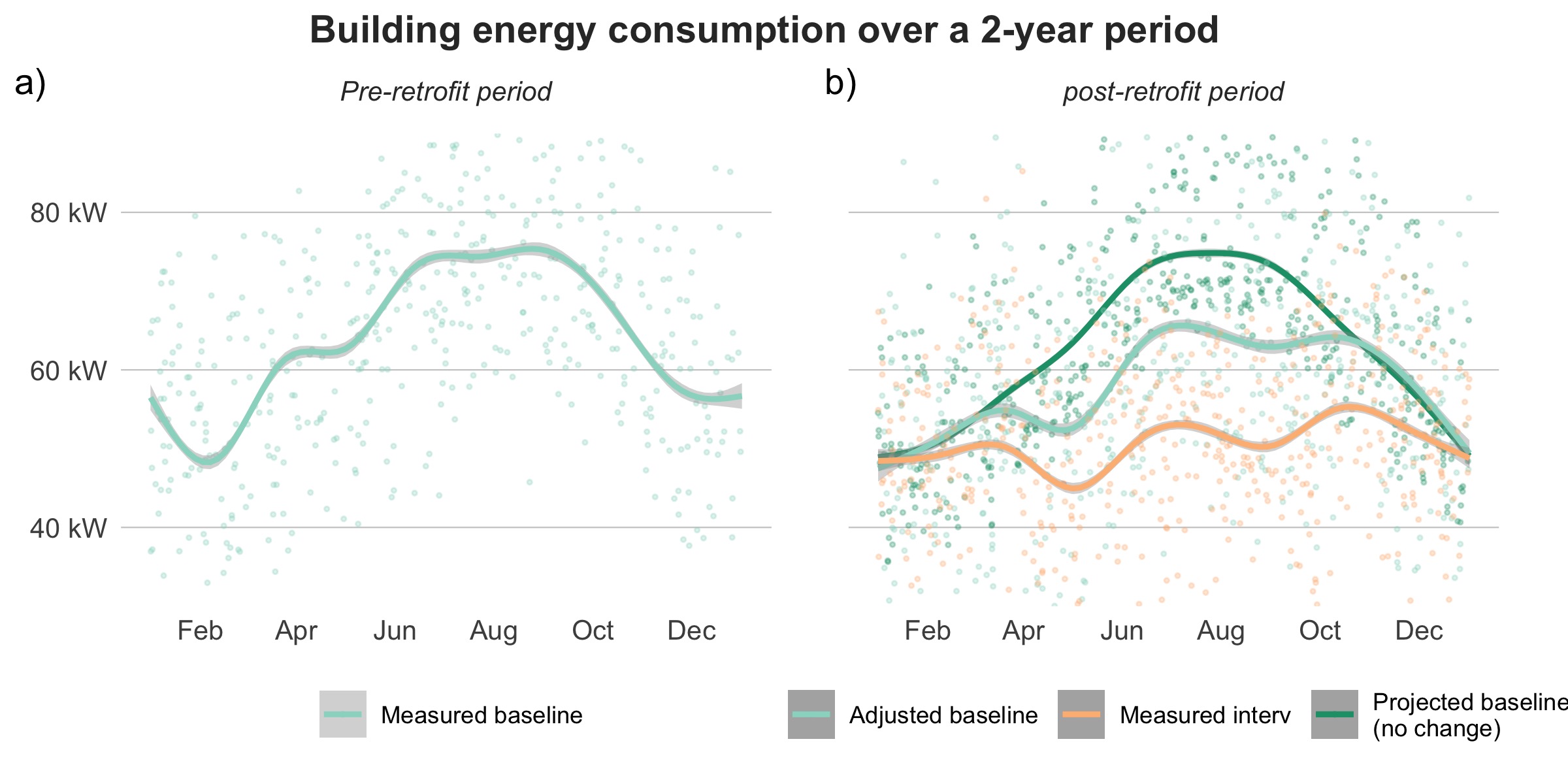


Figure 5.1: Energy measurement and modeling results for a 2-year conventional M&V project. Left: 500 random samples of measured baseline. Right: 500 random samples of measured intervention, projected baseline from modeling results without considering occupancy change and adjusted baseline measurement for occupancy change. (All lines fitted using loess function)

Figure 5.1 shows that If the analyst follows the steps described in G14 and IPMVP, they would start baseline measurements 12 months prior to the control retrofit implementation (subplot a) and fit an outdoor weather and time-dependent regression model using collected metered data as the baseline. In other words, the model is only valid for stable occupancy. Then the building manager enables the intervention control in the Building Automation System (BAS) and configures the existing baseline control as a backup when the intervention fails for safekeeping. Subplot b) in the figure 5.1 shows what happens after implementing the intervention, there is a 10 kW decrease at every metered interval due to a lower occupancy in the case study building. In the post-retrofit period, only intervention can be measured. The projected baseline is the baseline model prediction results using date/time features and outdoor weather conditions representing the analyst’s estimation in the post-retrofit period. The correctly adjusted baseline represents the underlying ground truth of this scenario but unobervable to the analyst, and the figure shows from April to September, the modeled/predicted baseline deviates from the correctly adjusted baseline results.

As a comparison, if the analyst follows the steps of the novel randomized M&V described in the manuscript, they would configure both control strategies in the BAS and switch back and forth between them depending on a pre-determined schedule. Figure 5.2 shows even if the analyst start the M&V during the occupancy change and has no prior knowledge about when and how long the occupancy has changed, there is a decent chance that each control strategy sampled 3 months with a balanced day-of-week category. In this way, at least the non-routine event has been measured.

# make daily totals  
df\_daily <- df\_hourly\_sch %>%  
 group\_by(datetime = floor\_date(datetime, unit = "day")) %>%  
 summarise(strategy = unique(strategy),  
 power\_ave = mean(power, na.rm = TRUE),  
 power\_peak = max(power, na.rm = TRUE),  
 t\_out = mean(t\_out, na.rm = TRUE)) %>%  
 ungroup()  
  
df\_daily %>%  
 mutate(strategy = as.factor(strategy),  
 strategy = recode\_factor(strategy, "1" = "Baseline", "2" = "Intervention")) %>%   
 ggplot(aes(x = datetime, y = power\_ave)) +   
 geom\_col(aes(fill = strategy), alpha = 0.3) +  
 geom\_line(aes(colour = strategy, group = 1)) +  
 scale\_x\_datetime(expand = c(0, 0),   
 date\_breaks = "3 months",   
 labels = label\_date\_short()) +  
 scale\_y\_continuous(expand = c(0, 0),   
 labels = number\_format(suffix =" kW")) +  
 scale\_color\_manual(values = ls\_colours) +  
 scale\_fill\_manual(values = ls\_colours) +  
 labs(title = "Building energy consumption measured\nusing randomized M&V method",  
 x = NULL,   
 y = NULL,   
 colour = NULL,   
 fill = NULL) +  
 guides(fill = guide\_legend(label.position = "bottom", label.hjust = 0.5, keywidth = 2.5, keyheight = 0.5, nrow = 1)) +  
 coord\_cartesian(clip = "off") +  
 theme(panel.grid.major.y = element\_line(colour = "grey80", linewidth = 0.25),  
 legend.position = "bottom",  
 axis.text = element\_text(size = 12))

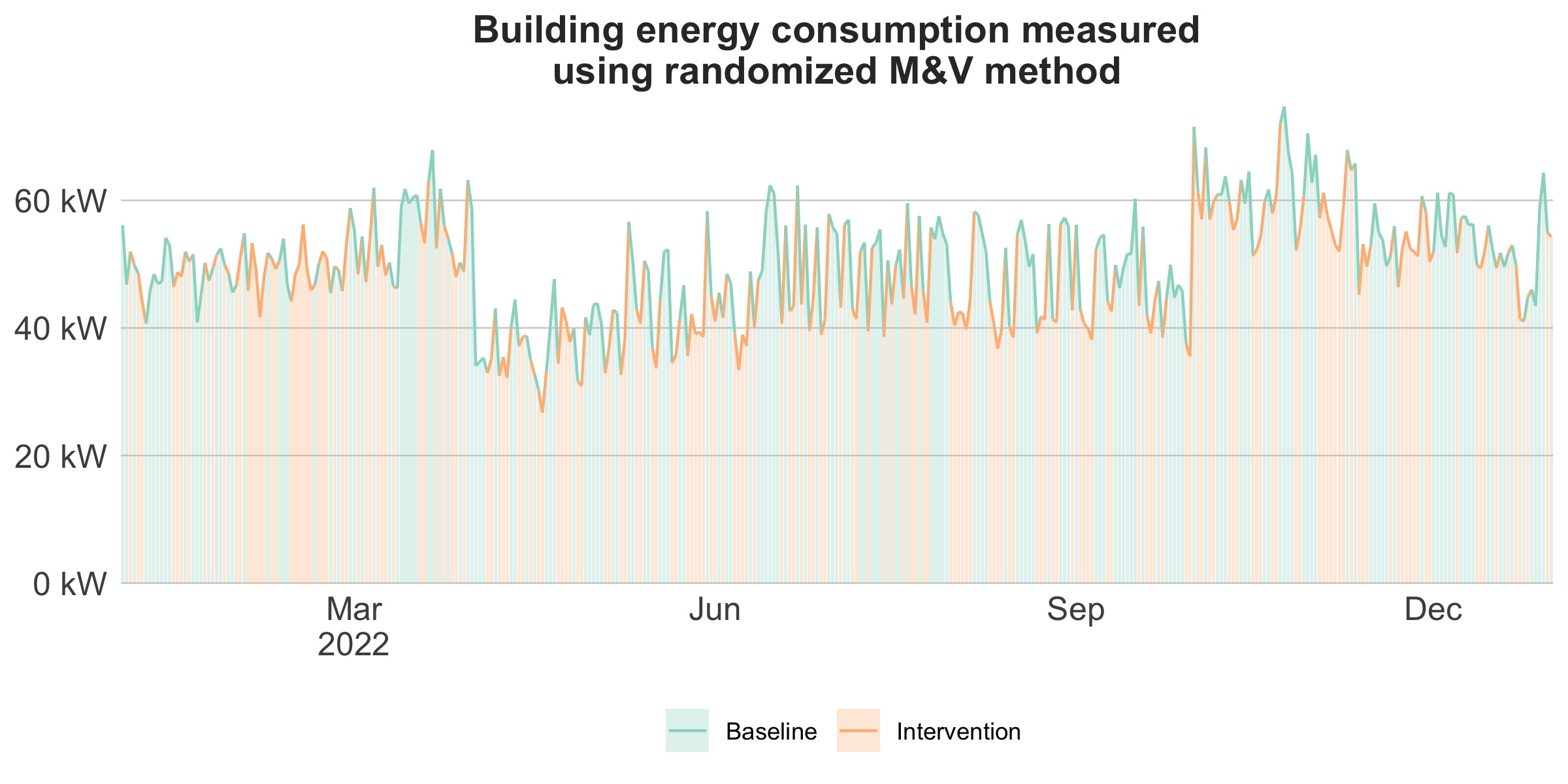


Figure 5.2: Averaged energy measurements at sampling interval (daily) using novel M&V method

The results shown in figure 5.3 demonstrate that the conventional pre-/post- method is likely to provide a higher saving estimation for this scenario if the analyst can not adjust for the proposed change, but the novel randomized method is robust and requires no additional assumptions or modeling effort. This is because sampling random at a 50%/50% ratio, the unexpected disturbance can mostly be balanced when using difference-in-mean as a saving-effec estimator.

# saving calculation  
FS\_true <- (mean(df\_base\_alt %>% .$power) -   
 mean(df\_interv\_alt %>% .$power)) /   
 (mean(df\_base\_alt %>% .$power)) \* 100  
  
FS\_new <- (mean(df\_hourly\_sch %>% filter(strategy == 1) %>% .$power) -   
 mean(df\_hourly\_sch %>% filter(strategy == 2) %>% .$power)) /   
 (mean(df\_hourly\_sch %>% filter(strategy == 1) %>% .$power)) \* 100  
  
FS\_conv <- (mean(baseline\_results %>% .$towt) -   
 mean(df\_interv\_alt %>% .$power)) /   
 (mean(baseline\_results %>% .$towt)) \* 100  
  
df\_FS <- data.frame(  
 values = c(FS\_true, FS\_conv, FS\_new),  
 category = factor(c("True", "Conventional M&V", "Randomized M&V"),   
 levels = c("True", "Conventional M&V", "Randomized M&V"))  
)  
  
df\_FS %>%   
 ggplot() +  
 geom\_col(aes(x = category, y = values),   
 alpha = 0.6,   
 position = "identity") +  
 geom\_text(aes(x = category, y = values, label = paste0(round(values, digits = 1), "%")),   
 position = position\_nudge(y = -0.5),  
 size = 4) +  
 scale\_y\_continuous(expand = c(0, 0),   
 label = number\_format(suffix = "%")) +  
 labs(title = "Estimated fractional savings\nby different methods",   
 x = NULL,   
 y = NULL) +  
 theme(legend.direction = "horizontal",  
 legend.position = "bottom",  
 plot.margin = margin(t = 2, r = 7, b = 2, l = 2, unit = "mm"))

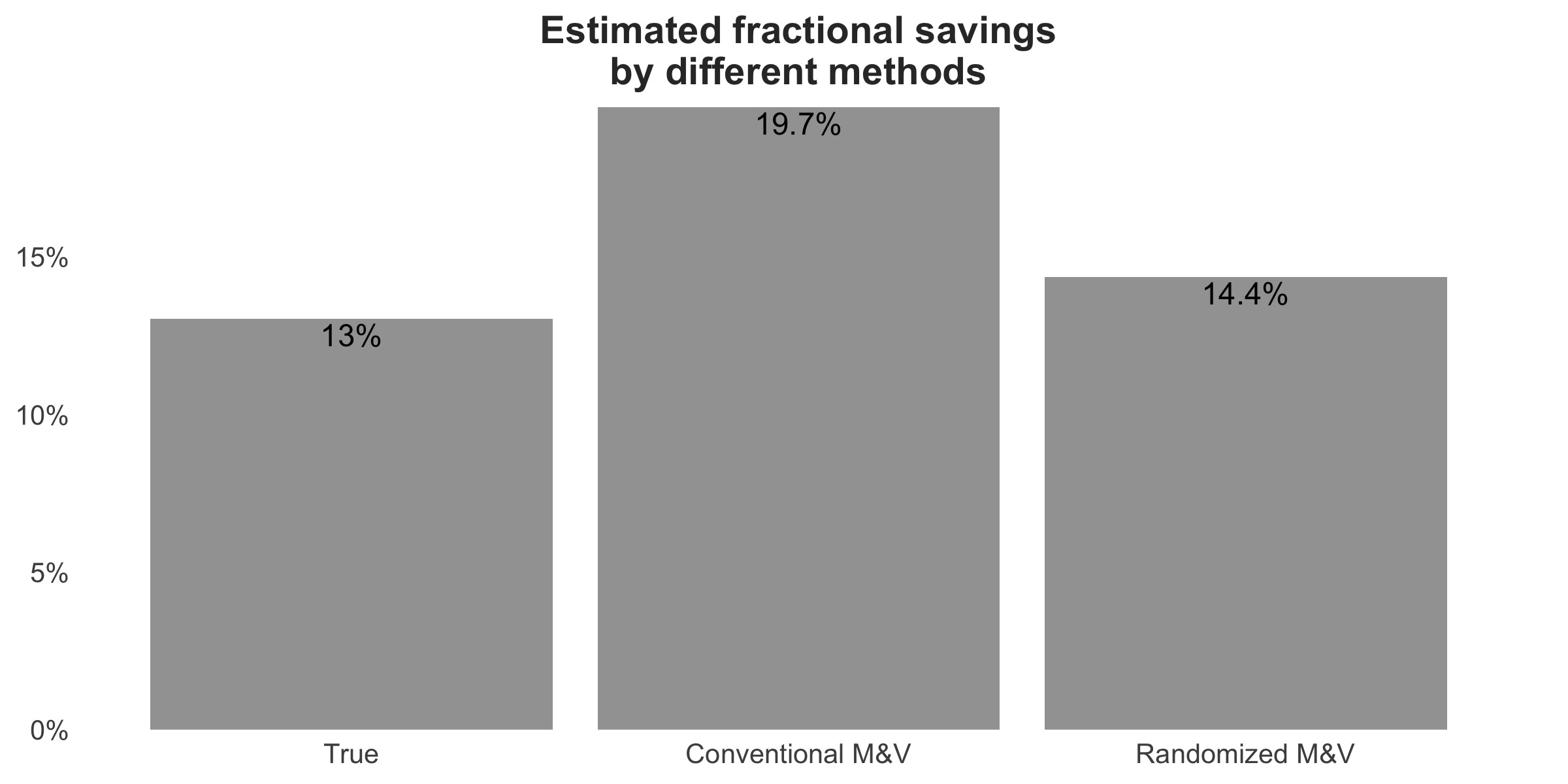


Figure 5.3: Comparison of savings estimation between conventional method and novel method proposed in the paper.

## 5.2 Scenario B: Continuous baseline change due to HVAC component degradation

A baseline can also change continuously over time, which makes it even harder to detect. One example is filter being gradually loaded in a AHU, increasing fan power due to increased static pressure to overcome the increased resistance of air passing through the filter. In this scenario, if the building manager changes the air filter before starting the M&V and the analyst chooses to use the conventional pre-/post-type analysis, the energy measured 12 months later in the post-retrofit period should correspond to a higher baseline due to increased fan energy as the particle accumulates in the filter. Here, the conventional M&V method will underestimate the intervention energy-saving effect. If the analyst chooses to use the novel random sampling M&V, the change in baseline can be documented in sampled baseline measurements.

In this scenario, we simulated a situation where the fan power gradually increased due to accumulated particles in the filter. We assume the fan power accounts for 25% of the HVAC electricity consumption and due to clogging, static pressure increases gradually by 50% over the course of the study period. The relationship between fan power and static pressure can be roughly estimated as , which means the fan power would also increase from 100% to 150%. We demonstrate such increase is omitted by the conventional method in the post-retrofit period where only the control intervention is measured.

# Generate schedule for the post-intervention period  
schedule <- blocking(start\_date = "2021-01-01",  
 n\_weeks = 108,   
 n\_seasons = 9,   
 seed = 390,   
 searches = 20,   
 jumps = 20,   
 treatments = 2,   
 consec = 1)  
  
df\_schedule <- data.frame(strategy = schedule$schedule$strategy,  
 datetime = schedule$schedule$date)  
  
df\_base\_sch <- df\_base %>%   
 left\_join(df\_schedule, by = "datetime") %>%   
 fill(strategy, .direction = "down") %>%   
 filter(strategy == 1)   
  
df\_interv\_sch <- df\_interv %>%   
 left\_join(df\_schedule, by = "datetime") %>%   
 fill(strategy, .direction = "down") %>%   
 filter(strategy == 2)  
  
df\_hourly\_sch <- bind\_rows(df\_base\_sch, df\_interv\_sch) %>%   
 arrange(datetime)  
  
# Add gradual change  
gradi\_week <- 1 / 106  
df\_base\_alt <- df\_base %>%  
 mutate(week = interval(min(datetime), datetime) %>% as.numeric('weeks') %>% floor()) %>%  
 mutate(ratio = gradi\_week \* week) %>%  
 mutate(delta\_fan = power \* 0.25 \* 0.5 \* ratio) %>%  
 mutate(power = power + delta\_fan)  
  
df\_interv\_alt <- df\_interv %>%  
 mutate(week = interval(min(datetime), datetime) %>% as.numeric('weeks') %>% floor()) %>%  
 mutate(ratio = gradi\_week \* week) %>%  
 mutate(delta\_fan = power \* 0.25 \* 0.5 \* ratio) %>%  
 mutate(power = power + delta\_fan)  
  
df\_hourly\_alt <- df\_hourly\_sch %>%   
 mutate(week = interval(min(datetime), datetime) %>% as.numeric('weeks') %>% floor()) %>%  
 mutate(ratio = gradi\_week \* week) %>%  
 mutate(delta\_fan = power \* 0.25 \* 0.5 \* ratio) %>%  
 mutate(power = power + delta\_fan)  
  
# Baseline projection  
model\_input\_options <- nmecr::assign\_model\_inputs(regression\_type = "TOWT", occupancy\_threshold = 0.65)  
  
baseline\_train <- df\_base\_alt %>%  
 filter(datetime < as.Date("2022-01-01")) %>%   
 select(time = datetime,   
 eload = power,   
 temp = t\_out)  
  
# do baseline model to estimate occupancy  
towt\_base <- nmecr::model\_with\_TOWT(training\_data = baseline\_train,  
 model\_input\_options = model\_input\_options)  
  
# Add occupancy info to model specification  
base\_model\_input\_options <- map(model\_input\_options, ~.) %>%   
 append(list(find\_occ\_unocc(training\_data = baseline\_train, model\_input\_options = towt\_base$model\_input\_options))) %>%   
 set\_names(., c(names(model\_input\_options), "occupancy\_info"))  
  
# re-estimate baseline with occupancy  
towt\_base <- nmecr::model\_with\_TOWT(training\_data = baseline\_train,  
 model\_input\_options = base\_model\_input\_options)  
  
baseline\_results <- model\_pred(amod = towt\_base$model\_occupied,   
 bmod = towt\_base$model\_unoccupied,  
 time = df\_towt$time,   
 temp = df\_towt$temp,  
 temp\_knots = towt\_base$model\_input\_options$calculated\_temp\_knots,   
 dataframe = df\_towt,  
 occ\_info = towt\_base$model\_input\_options$occupancy\_info,  
 interval\_minutes = towt\_base$model\_input\_options$interval\_minutes) %>%  
 rename("datetime" = "time") %>%   
 mutate("datetime" = force\_tz(datetime, tzone = "America/Chicago"))

p1 <- ggplot() +  
 geom\_point(data = df\_base\_alt %>%   
 filter(datetime < as.Date("2022-01-01")) %>%   
 slice\_sample(n = 500),   
 aes(x = datetime, y = power, color = "Measured baseline"),   
 size = 0.5,   
 alpha = 0.3) +  
 geom\_smooth(data = df\_base\_alt %>%   
 filter(datetime < as.Date("2022-01-01")),   
 aes(x = datetime, y = power, color = "Measured baseline")) +  
 scale\_x\_datetime(date\_breaks = "2 months",  
 date\_labels = "%b") +  
 scale\_y\_continuous(expand = c(0, 0),   
 breaks = breaks\_pretty(n = 3),  
 labels = number\_format(suffix = " kW")) +  
 coord\_cartesian(clip = "on", ylim = c(30, 90)) +  
 scale\_color\_manual(values = ls\_colours) +  
 labs(x = NULL,   
 y = NULL,  
 color = NULL,   
 title = NULL,  
 subtitle = "Pre-retrofit period") +  
 theme(panel.grid.major.y = element\_line(colour = "grey80", size = 0.25),  
 legend.direction = "horizontal",  
 legend.position = "bottom",  
 plot.margin = margin(t = 2, r = 7, b = 2, l = 2, unit = "mm"))  
  
p2 <- ggplot() +  
 geom\_point(data = baseline\_results %>%   
 slice\_sample(n = 500),   
 aes(x = datetime, y = towt, color = "Projected baseline\n(no change)"),   
 size = 0.5,   
 alpha = 0.3) +  
 geom\_smooth(data = baseline\_results,   
 aes(x = datetime, y = towt, color = "Projected baseline\n(no change)")) +  
  
 geom\_point(data = df\_base\_alt %>%   
 filter(datetime >= as.Date("2022-01-01")) %>%   
 slice\_sample(n = 500),   
 aes(x = datetime, y = power, color = "Adjusted baseline"),   
 size = 0.5,   
 alpha = 0.3) +  
 geom\_smooth(data = df\_base\_alt %>%   
 filter(datetime >= as.Date("2022-01-01")),   
 aes(x = datetime, y = power, color = "Adjusted baseline")) +  
 geom\_point(data = df\_interv\_alt %>%   
 filter(datetime >= as.Date("2022-01-01")) %>%   
 slice\_sample(n = 500),   
 aes(x = datetime, y = power, color = "Measured interv"),   
 size = 0.5,   
 alpha = 0.3) +  
 geom\_smooth(data = df\_interv\_alt %>% filter(datetime >= as.Date("2022-01-01")),   
 aes(x = datetime, y = power, color = "Measured interv")) +  
 scale\_x\_datetime(date\_breaks = "2 months",  
 date\_labels = "%b") +  
 scale\_y\_continuous(expand = c(0, 0),   
 breaks = breaks\_pretty(n = 3),  
 labels = number\_format(suffix = " kW")) +  
 coord\_cartesian(clip = "on", ylim = c(30, 90)) +  
 scale\_color\_manual(values = ls\_colours) +  
 labs(x = NULL,   
 y = NULL,  
 color = NULL,   
 title = NULL,   
 subtitle = "Post-retrofit period") +  
 theme(panel.grid.major.y = element\_line(colour = "grey80", size = 0.25),  
 axis.text.y = element\_blank(),  
 legend.direction = "horizontal",  
 legend.position = "bottom",  
 plot.margin = margin(t = 2, r = 7, b = 2, l = 2, unit = "mm"))  
  
p1 + p2 +  
 plot\_annotation(title = "Building energy consumption over a 2-year period") +  
 plot\_annotation(tag\_levels = c('a'), tag\_suffix = ')') &  
 theme(plot.tag.position = c(0, 1),  
 plot.tag = element\_text(color="black"))

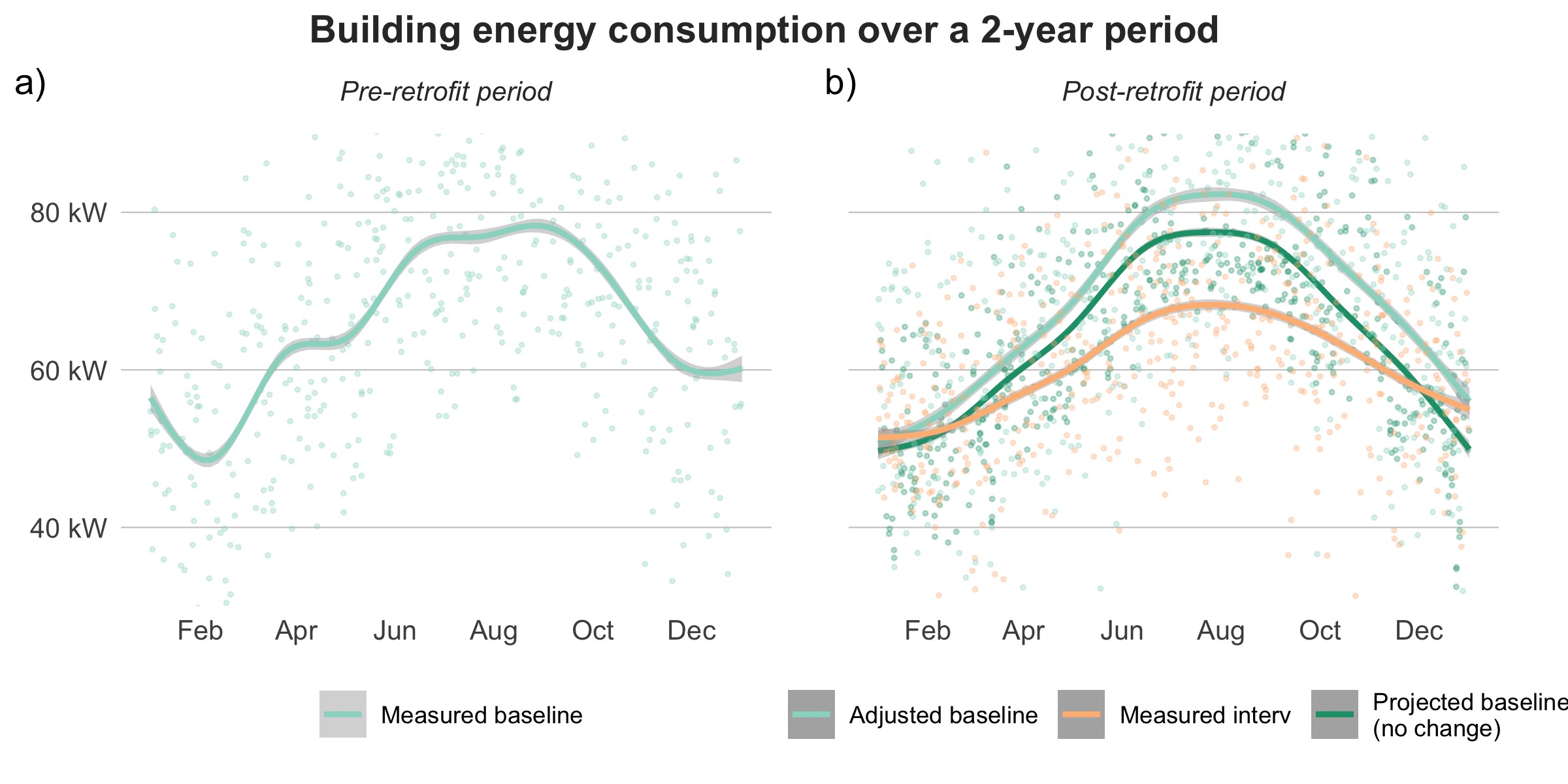


Figure 5.4: Energy measurement and modeling results for a 2-year conventional M&V project. Left: 500 random samples of measured baseline. Right: 500 random samples of measured intervention, projected baseline from modeling results without considering filter clogging and adjusted baseline measurement for filter clogging. (All lines fitted using loess function)

Similarly, the above figure 5.4 shows both metered energy consumption and modeled baseline for a 2-year M&V project in this scenario. The figure shows if using the conventional method, the analyst would start a 12-month baseline and fit a regression model using the collected measurements. If the static pressure increase covers the course of the study period, there is a chance that the fitted model may indirectly account for the change in baseline in this case. However, subplot b) shows that the projected modeling results still under-estimate the fan energy increase compared to correctly adjusted baseline in the post-retrofit period. Normally the fan energy increase can be detected using sensors connected to the AHU, but it is outside the scope for a whole building metered energy consumption M&V. However, if the analyst choose to apply the random switching method shown in figure 5.5, the energy increase can be measured any time throughout the study period.

# make daily totals  
df\_daily <- df\_hourly\_alt %>%  
 group\_by(datetime = floor\_date(datetime, unit = "day")) %>%  
 summarise(strategy = unique(strategy),  
 power\_ave = mean(power, na.rm = TRUE),  
 power\_peak = max(power, na.rm = TRUE),  
 t\_out = mean(t\_out, na.rm = TRUE)) %>%  
 ungroup() %>%   
 filter(datetime >= as.Date("2022-01-01"))  
  
df\_daily %>%  
 mutate(strategy = as.factor(strategy),  
 strategy = recode\_factor(strategy, "1" = "Baseline", "2" = "Intervention")) %>%   
 ggplot(aes(x = datetime, y = power\_ave)) +   
 geom\_col(aes(fill = strategy), alpha = 0.3) +  
 geom\_line(aes(colour = strategy, group = 1)) +  
 scale\_x\_datetime(expand = c(0, 0),   
 date\_breaks = "3 months",   
 labels = label\_date\_short()) +  
 scale\_y\_continuous(expand = c(0, 0),   
 labels = number\_format(suffix =" kW")) +  
 scale\_color\_manual(values = ls\_colours) +  
 scale\_fill\_manual(values = ls\_colours) +  
 labs(title = "Building energy consumption measured\nusing randomized M&V method",  
 x = NULL,   
 y = NULL,   
 colour = NULL,   
 fill = NULL) +  
 guides(fill = guide\_legend(label.position = "bottom", label.hjust = 0.5, keywidth = 2.5, keyheight = 0.5, nrow = 1)) +  
 coord\_cartesian(clip = "off") +  
 theme(panel.grid.major.y = element\_line(colour = "grey80", linewidth = 0.25),  
 legend.position = "bottom",  
 axis.text = element\_text(size = 12))

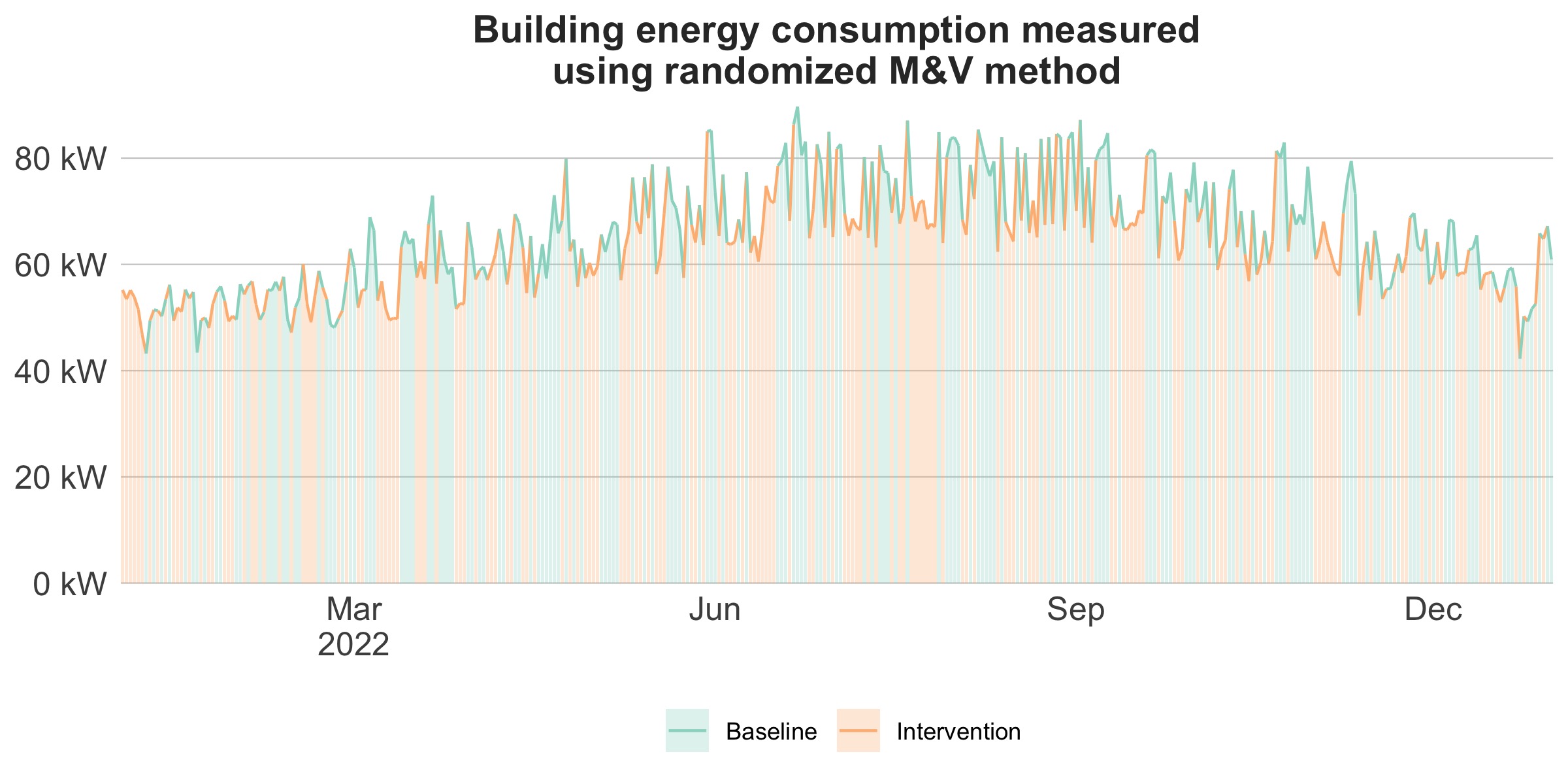


Figure 5.5: Averaged energy measurements at sampling interval (daily) using novel M&V method

As a result, figure 5.6 implies that the conventional M&V significantly underestimates the intervention energy-saving effect due to a lack of baseline measurements in the post-retrofit duration. Additionally as expected, when using the randomized M&V, savings are estimated much more accurately.

# True M&V savings  
FS\_true <- (mean(df\_base\_alt %>% .$power) -   
 mean(df\_interv\_alt %>% .$power)) /   
 (mean(df\_base\_alt %>% .$power)) \* 100  
  
# Conventional M&V savings (biased)  
FS\_conv <- (mean(baseline\_results %>% .$towt) -   
 mean(df\_interv\_alt %>% filter(datetime >= as.Date("2022-01-01")) %>% .$power)) /  
 (mean(baseline\_results %>% .$towt)) \* 100  
  
# Noevl M&V savings   
  
FS\_new <- (mean(df\_hourly\_alt %>% filter(strategy == 1) %>% .$power) -   
 mean(df\_hourly\_alt %>% filter(strategy == 2) %>% .$power)) /   
 (mean(df\_hourly\_alt %>% filter(strategy == 1) %>% .$power)) \* 100  
  
df\_FS <- data.frame(  
 values = c(FS\_true, FS\_conv, FS\_new),  
 category = factor(c("True", "Conventional M&V", "Randomized M&V"),   
 levels = c("True", "Conventional M&V", "Randomized M&V"))  
)  
  
df\_FS %>%   
 ggplot() +  
 geom\_col(aes(x = category, y = values),   
 alpha = 0.6,   
 position = "identity") +  
 geom\_text(aes(x = category, y = values, label = paste0(round(values, digits = 1), "%")),   
 position = position\_nudge(y = -0.5),  
 size = 4) +  
 scale\_y\_continuous(expand = c(0, 0),   
 label = number\_format(suffix = "%")) +  
 labs(title = "Estimated fractional savings\nby different methods",   
 x = NULL,   
 y = NULL) +  
 theme(legend.direction = "horizontal",  
 legend.position = "bottom",  
 plot.margin = margin(t = 2, r = 7, b = 2, l = 2, unit = "mm"))

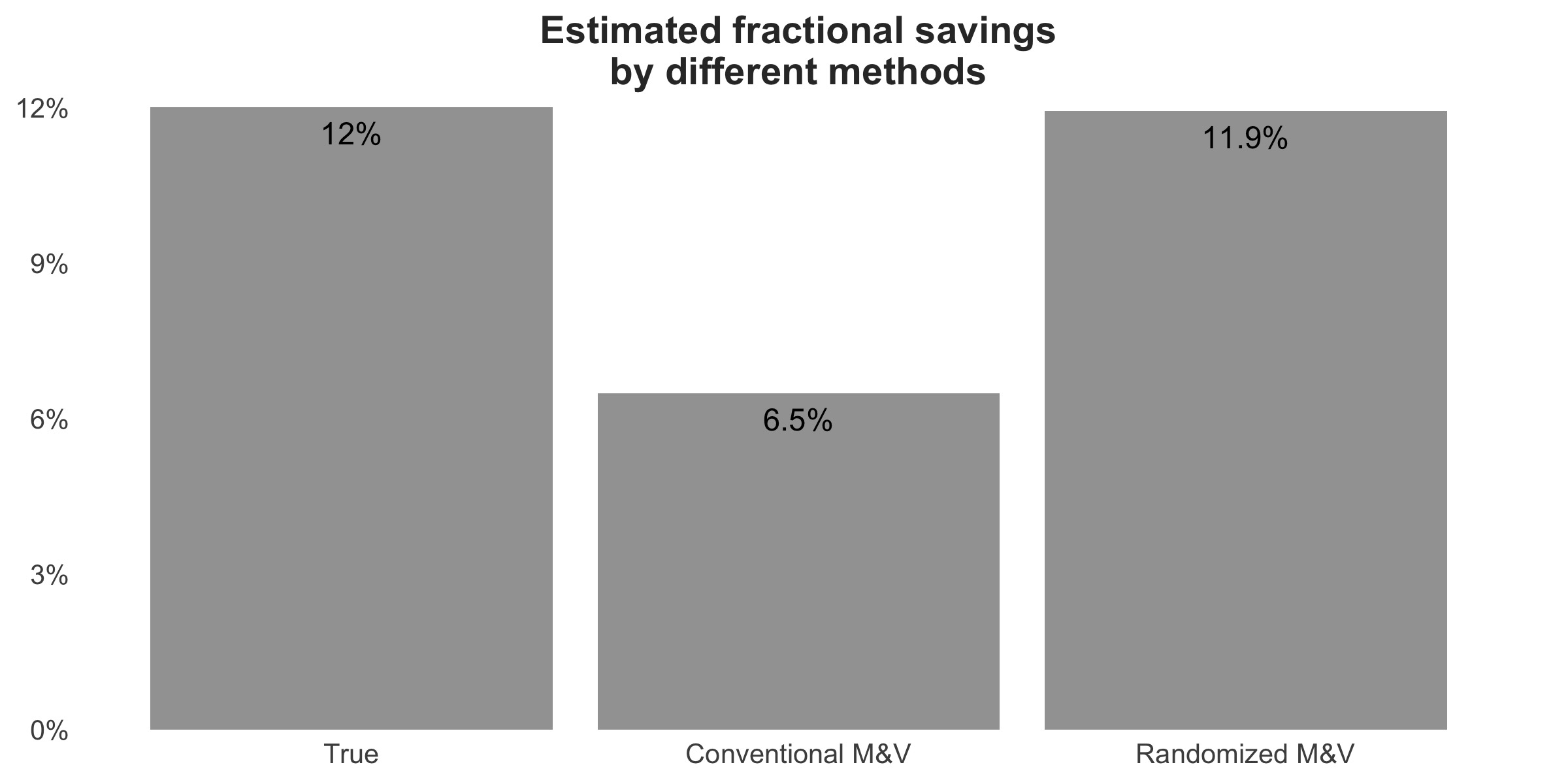


Figure 5.6: Saving estimation comparison between true savings and the two methods (conventional M&V method: no prior knowledge of filter clogging and does not measure baseline, and the novel randomized M&V method: no prior knowledge of filter clogging but measures updated baseline).