Animation: Identifying Minimal Cost Compute Commitment Level

Murray Stokely

2024-10-08

Contents

1 Introduction

This file includes code to generate a series of hundreds or thousands of figures to serve as frames in an animation of the optimization process.

We start by restricting to a 2-week subset of the data from Summer 2024, aggregate the different regions and SKUs together to a single timeseries of total VM demand, and normalize this to a 100 unit peak over that time window.

```
this.dir <- getwd()
data <- read_parquet("../hourly_normalized.parquet")
data <- data %>% subset(USAGE_HOUR >= as.POSIXct("2024-04-07", tz="UTC")) %>%
  subset(USAGE_HOUR < as.POSIXct("2024-04-21", tz="UTC"))
dim(data)
```

```
## [1] 11376     4
```

```
head(data)
```

```
## # A tibble: 6 x 4
##   USAGE_HOUR          REGION_NUM INSTANCE_TYPE NORM_USAGE
##   <dttm>                   <dbl> <chr>               <dbl>
## 1 2024-04-07 00:00:00          1 D                      10
## 2 2024-04-07 00:00:00          2 K                       4
## 3 2024-04-07 00:00:00          2 B                     501
## 4 2024-04-07 00:00:00          4 J                      13
## 5 2024-04-07 00:00:00          1 K                       7
## 6 2024-04-07 00:00:00          2 F                      23
```

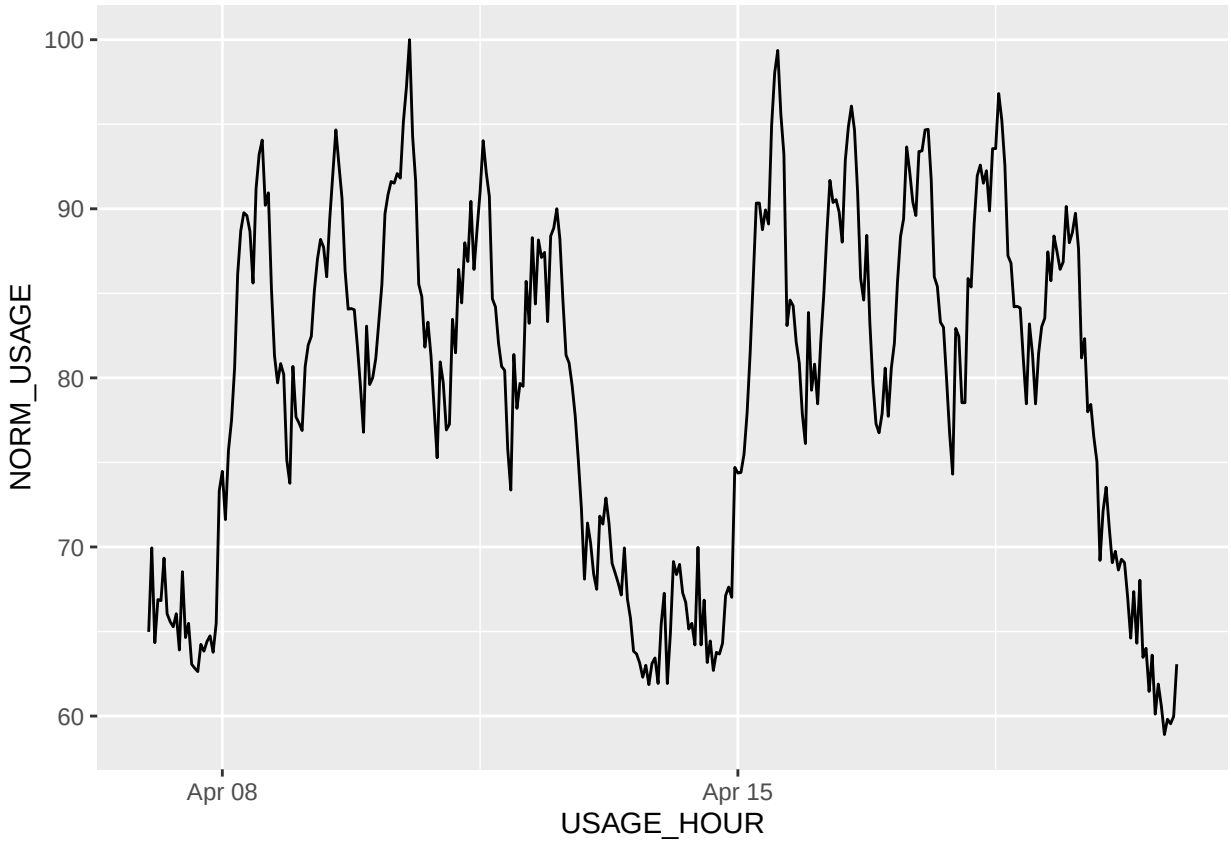
```
range(data$USAGE_HOUR)
```

```
## [1] "2024-04-07 00:00:00 UTC" "2024-04-20 23:00:00 UTC"
```

```r
data.all <- data %>% group_by(USAGE_HOUR) %>% summarise(across(NORM_USAGE, sum))
data.all$NORM_USAGE = 100 * data.all$NORM_USAGE / max(data.all$NORM_USAGE)
head(data.all)
```

```
## # A tibble: 6 x 2
##   USAGE_HOUR          NORM_USAGE
##   <dttm>                   <dbl>
## 1 2024-04-07 00:00:00       65.0
## 2 2024-04-07 01:00:00       69.9
## 3 2024-04-07 02:00:00       64.3
## 4 2024-04-07 03:00:00       66.9
## 5 2024-04-07 04:00:00       66.8
## 6 2024-04-07 05:00:00       69.3
```

```r
ggplot(data.all, aes(x=USAGE_HOUR, y=NORM_USAGE)) + geom_line()
```



2 Visualization and Cost Computations

In order to illustrate the impact of savings plans on the amount of VM demand that is (1) covered by a savings plan, (2) purchased with on-demand rates, and (3) wasted as an unused savings plan, we introduce a simple 3-color area visualization with time on the x-axis and normalized cost on the y-axis.

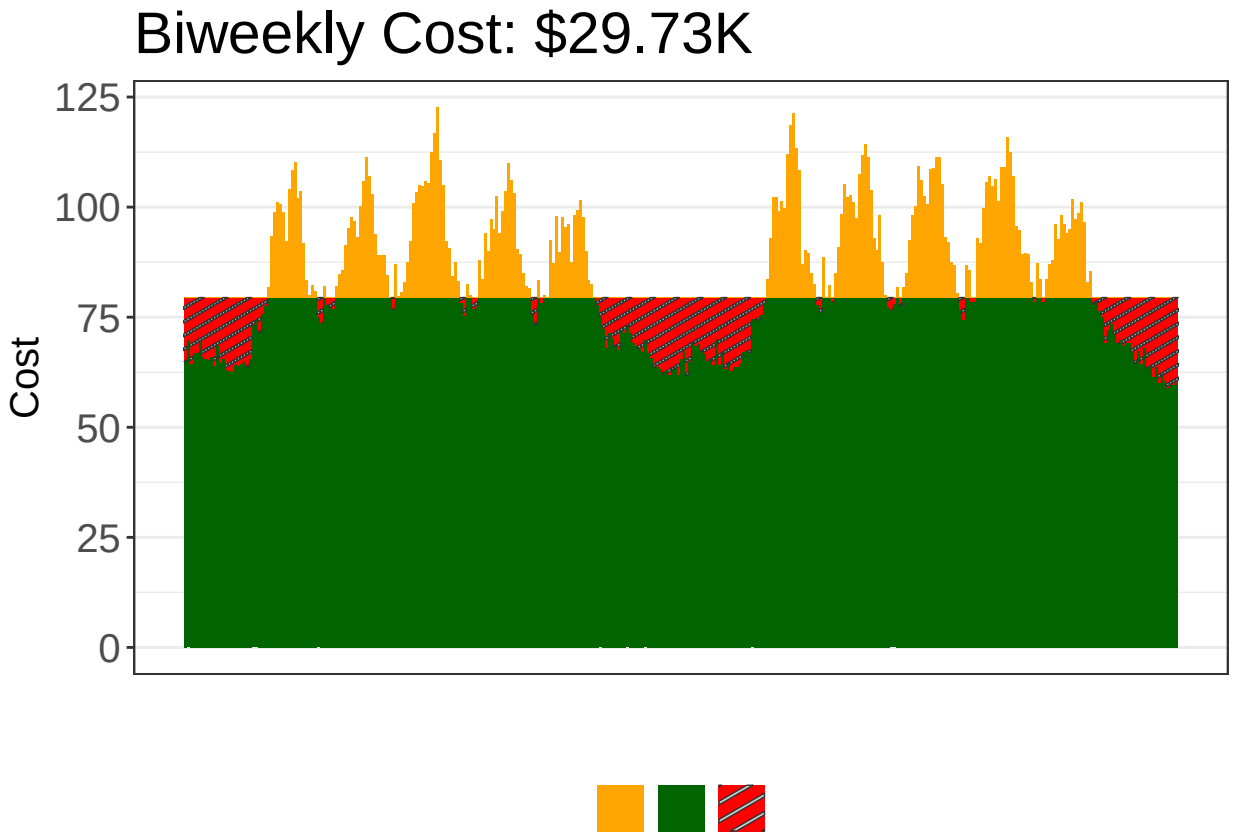
Note that since the y-axis is cost, instead of VM instance hours, the more of the demand that is covered at expensive on-demand rates means the higher the y-axis will be.

```r
TotalCostPremium <- function(df) {
  # TotalCost - Computes the Total Cost Premium
  #
```

```
  # Args:
  #   df - A data.frame with 3 areas - unused, covered, on-demand.
  # Returns
  #   Cost premium

  l1 <- subset(df, Pricing == "SP Commitment")
  l2 <- subset(df, Pricing == "Unused Commitment")
  l3 <- subset(df, Pricing == "On Demand")
  plan.cost <- sum(l1$ymax - l1$ymin)
  unused.cost <- sum(l2$ymax - l2$ymin)
  ondemand.cost <- sum(l3$ymax - l3$ymin)
  total.cost <- plan.cost + unused.cost + ondemand.cost
  return(total.cost)
}

GenerateBoxes <- function(df, on.demand.premium=2.1, label=NULL) {
  # GenerateBoxes - Create data.frame with pricing
  # Args:
  #   df: A data.frame with columns
  #     USAGE_HOUR: POSIXct hourly time
  #     NORM_USAGE: The VM demand at that hour
  # 3 boxes each time range - ununused, used with sp/ri coverage,
  # used above sp/ri coverage level.
  data.boxes <- data.frame(
    xmin = rep(head(df$USAGE_HOUR, -1), 3),
    xmax = rep(tail(df$USAGE_HOUR, -1), 3),
    Pricing = c(rep("SP Commitment", length(head(df$USAGE_HOUR, -1))),
                rep("Unused Commitment", length(head(df$USAGE_HOUR, -1))),
                rep("On Demand", length(head(df$USAGE_HOUR, -1)))),
    ymin = c(rep(0, length(head(df$USAGE_HOUR, -1))),
             head(ifelse(df$NORM_USAGE < df$SPRI_LEVEL, df$NORM_USAGE,
                         df$SPRI_LEVEL), -1), head(df$SPRI_LEVEL, -1)),
    ymax = c(head(ifelse(df$NORM_USAGE < df$SPRI_LEVEL,
                         df$NORM_USAGE, df$SPRI_LEVEL), -1),
             head(df$SPRI_LEVEL,-1),
             head(ifelse(df$NORM_USAGE > df$SPRI_LEVEL,
                         df$SPRI_LEVEL +
                           (df$NORM_USAGE - df$SPRI_LEVEL)*on.demand.premium,
                         df$SPRI_LEVEL), -1)))

  if (!is.null(label)) {
    data.boxes$LABEL = label
  }
  return(data.boxes)
}

AnnotateSPRILevel <- function(df, sp.level) {
  df$SPRI_LEVEL = sp.level
  return(df)
}

# DEMAND = sp + ri + (ondemand / on.demand.premium) - unused
# sp + ri + ondemand = 41567.9
```

```
#
PlotBoxes <- function(df, label.size=15, label=NULL, ymax=NULL,
                      hide.legend=TRUE) {
  days <- c("S", "M", "T", "W", "R", "F", "S")
  p <- ggplot(df) +
    geom_rect_pattern(aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax,
                          fill=Pricing, pattern=Pricing), colour=NA,
                      pattern_size=0.25, pattern_spacing=0.02) +
    theme_bw() +
    theme(axis.text=element_text(size=label.size),
          title=element_text(size=1.2*label.size),
          axis.title=element_text(size=label.size),
          legend.text = if (hide.legend) element_blank() else
            element_text(size=label.size),
          legend.title = if (hide.legend) element_blank() else
            element_text(size=label.size),
          legend.position="bottom", strip.text=element_text(size=label.size)) +
    ylab("Cost") + xlab("") +
    scale_pattern_manual(values=c("none", "none", "stripe")) +
    scale_x_continuous(breaks=seq(from=as.POSIXct("2024-06-02 12:00:00"),
                                  to=as.POSIXct("2024-06-15 12:00:00"),
                                  length.out=14),
                       labels=rep(days,2)) +
    scale_fill_manual(values=c("orange", "darkgreen", "red"))
  if (!is.null(ymax)) {
    p <- p + ylim(c(0,ymax))
  }

  if (is.null(label)) {
    return(p + ggtitle(paste0("Biweekly Cost: $",
                              round(TotalCostPremium(df)/1000, 2), "K")))
  } else {
    return(p + facet_wrap(~LABEL) + theme(strip.text = element_text(size=16)))
  }
}
```

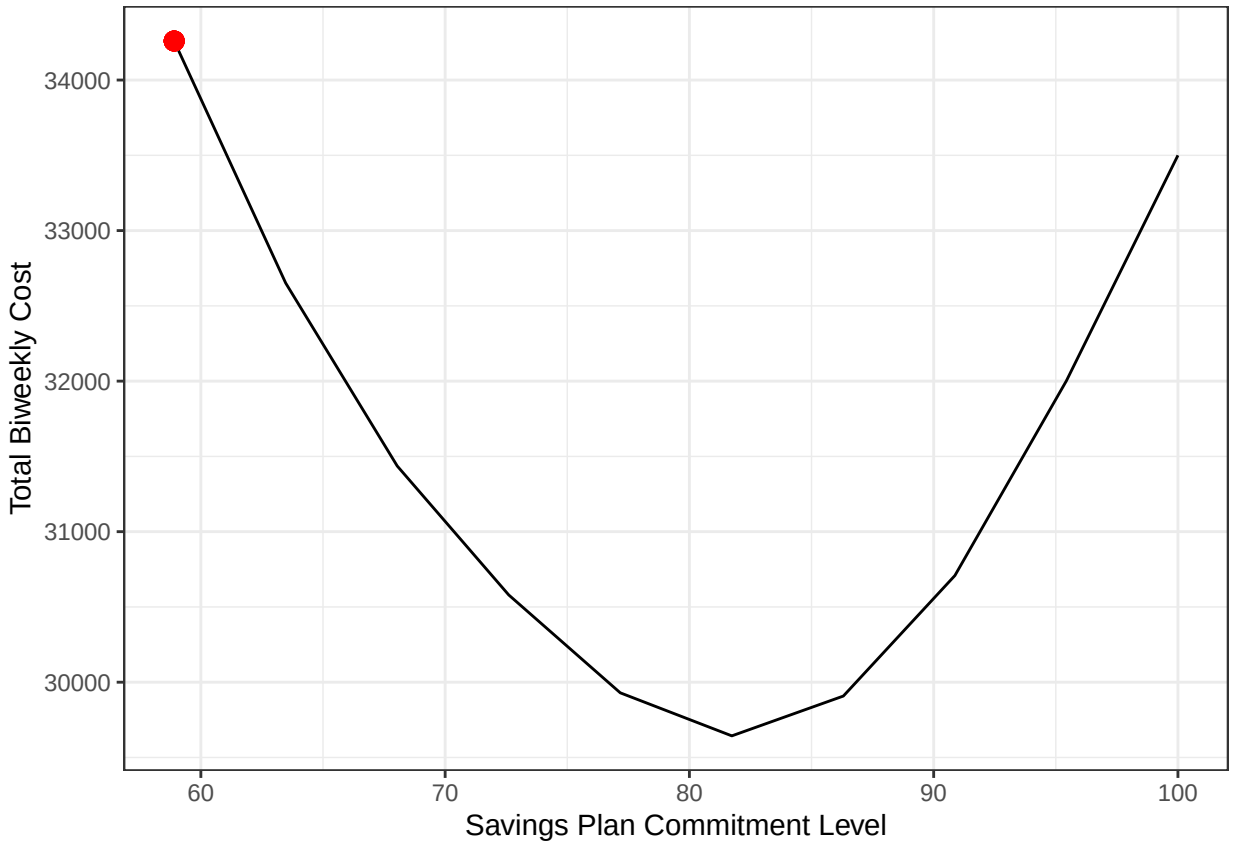
```
df <- AnnotateSPRILevel(data.all,
                        (max(data.all$NORM_USAGE) + min(data.all$NORM_USAGE))/2)
data.boxes <- GenerateBoxes(df)

PlotBoxes(data.boxes)
```



3 Comparing Commitment Level (x-axis) vs Total Cost (y-axis)

As we iterate through many possible solutions to identify the lowest cost option we keep track and plot the different commitment levels and the associated total cost.

```
xs <- seq(min(data.all$NORM_USAGE), max(data.all$NORM_USAGE), length=10)
ys <- c()
for (commit.level in xs) {
  df.tmp <- AnnotateSPRILevel(data.all, sp.level=commit.level)
  data.boxes.tmp <- GenerateBoxes(df.tmp, label=commit.level)
  cost.tmp <- TotalCostPremium(data.boxes.tmp)
  ys <- c(ys, cost.tmp)
}

opt.curve <- data.frame(x=xs, y=ys)
ggplot(opt.curve, aes(x=x, y=y)) + geom_line() + theme_bw() +
  xlab("Savings Plan Commitment Level") + ylab("Total Biweekly Cost") +
  geom_point(x=xs[1], y=ys[1], size=3, color="red")
```



4 Animations

4.1 Commitment Level Area Plot

Our basic approach for generating animations is to create a number of plots in a loop with identical x and y axis limits and then stitch them together with ffmpeg into an animation.

We first look at our area plot showing the impact of different commitment levels on the amount of used commitment, unused commitment, and on demand capacity.

The individual frames are written out to an ephermal temporary directory created by the R session, but the final output animations are written out to the current directory.

For a smoother animation, create more frames by increasing the steps number below.

```
#steps <- 100
steps <- 10
ymin <- NULL
ymax <- NULL
ls <- round(seq(from=min(data.all$NORM_USAGE), to=max(data.all$NORM_USAGE), length=steps))
i <- 1
tdir <- tempdir()

for (commit.level in ls) {
  df.sub.tmp <- AnnotateSPRILevel(data.all, sp.level=commit.level)
  data.boxes.tmp <- GenerateBoxes(df.sub.tmp, label=commit.level)
  y <- TotalCostPremium(data.boxes.tmp)
  if (is.null(ymin)) {
```

```
    ymin <- y
  } else if (y < ymin) {
    ymin <- y
  }
  if (is.null(ymax)) {
    ymax <- y
  } else if (y > ymax) {
    ymax <- y
  }
  filename <- file.path(tdir, paste0(sprintf("%03d", i), ".png"))
  print(paste0("Working on ", commit.level, " filename: ", filename))
  i <- i + 1
  #CairoPNG(filename, width=640, height=480)
  CairoPNG(filename, width=3840, height=2160)
  show(PlotBoxes(data.boxes.tmp, label.size=100, ymax=150))
  dev.off()
}
```

```
## [1] "Working on 59 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/001.png"
## [1] "Working on 63 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/002.png"
## [1] "Working on 68 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/003.png"
## [1] "Working on 73 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/004.png"
## [1] "Working on 77 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/005.png"
## [1] "Working on 82 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/006.png"
## [1] "Working on 86 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/007.png"
## [1] "Working on 91 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/008.png"
## [1] "Working on 95 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/009.png"
## [1] "Working on 100 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/010.png"
```

```
input.files <- file.path(tdir, "%03d.png")
out.file <- file.path(this.dir, "out.mp4")
cmd <- paste0("ffmpeg -y -r 10 -i ", input.files,
              " -c:v libx264 -r 30 -pix_fmt yuv420p ", out.file)
system(cmd)
```

4.2 Cost vs Commitment Level Optimization Curve

We then look at generating some static images for the cost vs commitment level curve.

```
steps <- 10
label.size <- 10
xs <- seq(min(data.all$NORM_USAGE), max(data.all$NORM_USAGE), length=10)
i <- 1

tdir <- tempdir()
x <- c()
ys <- c()
xmin <- min(data.all$NORM_USAGE)
xmax <- max(data.all$NORM_USAGE)
```

```
for (commit.level in xs) {
  df.tmp <- AnnotateSPRILevel(data.all, sp.level=commit.level)
  data.boxes.tmp <- GenerateBoxes(df.tmp, label=commit.level)
  cost.tmp <- TotalCostPremium(data.boxes.tmp)
  ys <- c(ys, cost.tmp)
  x <- c(x, commit.level)
  # Now make a graph
  opt.curve <- data.frame(x=x, y=ys)


  filename <- file.path(tdir, paste0(sprintf("s%03d", i), ".png"))
  i <- i+1
  print(paste0("Working on ", filename))
  label.size <- 100
  CairoPNG(filename, width=3840, height=2160)

  show(ggplot(opt.curve, aes(x=x, y=y)) + geom_line() + theme_bw() +
         theme(axis.text=element_text(size=label.size),
               axis.title=element_text(size=label.size)) +
         ylim(c(.95*ymin, 1.05*ymax)) +
         xlim(c(xmin, xmax)) +
         xlab("Savings Plan Commitment Level") +
         ylab("Total Cost to Snowflake") +
         geom_point(x=commit.level, y=cost.tmp, size=label.size*.5,
                    color="red"))
  dev.off()
}
```

```
## [1] "Working on /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/s001.png"
```

```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```

```
## [1] "Working on /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/s002.png"
```

```
## [1] "Working on /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/s003.png"
```

```
## [1] "Working on /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/s004.png"
```

```
## [1] "Working on /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/s005.png"
```

```
## [1] "Working on /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/s006.png"
```

```
## [1] "Working on /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/s007.png"
```

```
## [1] "Working on /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/s008.png"
```

```
## [1] "Working on /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/s009.png"
```

```
## [1] "Working on /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/s010.png"
```

4.3 Combined Animation with both plots

The final step is to combine both plots side by side in the animation to see the impact on total cost spend as the savings plan commitment level changes.

```
#steps <- 600
steps <- 10
min.commit <- min(data.all$NORM_USAGE)
```

```
max.commit <- max(data.all$NORM_USAGE)
commit.levels <- round(seq(from=min(data.all$NORM_USAGE),
                           to=max(data.all$NORM_USAGE), length=steps))
print(commit.levels)
```

```
##  [1]  59  63  68  73  77  82  86  91  95 100
```

```
i <- 1
curr.d <- getwd()

x <- c()
ys <- c()
label.size <- 100
min.cost <- NULL
min.level <- NULL

for (commit.level in commit.levels) {
  df.sub.tmp <- AnnotateSPRILevel(data.all, sp.level=commit.level)
  data.boxes.tmp <- GenerateBoxes(df.sub.tmp, label=commit.level)
  cost.tmp <- TotalCostPremium(data.boxes.tmp)
  if (is.null(min.cost)) {
    min.cost <- cost.tmp
    min.level <- commit.level
  }
  if (min.cost > cost.tmp) {
    min.cost <- cost.tmp
    min.level <- commit.level
  }
  ys <- c(ys, cost.tmp)
  x <- c(x, commit.level)

  filename <- file.path(tdir, paste0(sprintf("c%04d", i), ".png"))
  print(paste0("Working on ", commit.level, " filename: ", filename))
  i <- i + 1
  #CairoPNG(filename, width=640, height=480)
  CairoPNG(filename, width=3840, height=2160)
  p1 <- PlotBoxes(data.boxes.tmp, label.size=label.size*0.9, ymax=150)
  opt.curve <- data.frame(x=x, y=ys)
  p2 <- ggplot(opt.curve, aes(x=x, y=y)) + geom_line() + theme_bw() +
    theme(axis.text=element_text(size=label.size*0.8),
          axis.title=element_text(size=label.size*0.8)) +
    ylim(c(.98*ymin, 1.02*ymax)) + xlim(c(xmin, xmax)) +
    xlab("Hourly Commitment Level") + ylab("Total Biweekly Cost") +
    geom_point(x=commit.level, y=cost.tmp, size=label.size*.5, color="red") +
    theme(plot.margin=unit(c(500,50,300,400), "points"))

  show(plot_grid(p1, p2, labels = c('A', 'B'), label_size = 12))
  dev.off()
}
```

```
## [1] "Working on 59 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/c0001.png"
```

```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```

```
## [1] "Working on 63 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/c0002.png"
```

```
## [1] "Working on 68 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/c0003.png"
## [1] "Working on 73 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/c0004.png"
## [1] "Working on 77 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/c0005.png"
## [1] "Working on 82 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/c0006.png"
## [1] "Working on 86 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/c0007.png"
## [1] "Working on 91 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/c0008.png"
## [1] "Working on 95 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/c0009.png"
## [1] "Working on 100 filename: /var/folders/ml/dklf89m936vd0hmmtct7vcbh0000gn/T//RtmpbfOeG7/c0010.png
```

```
input.files <- file.path(tdir, "c%04d.png")
out.file <- file.path(this.dir, "combined.mp4")
cmd <- paste0("ffmpeg -y -i ", input.files,
              " -c:v libx264 -r 24 -pix_fmt yuv420p ", out.file)
system(cmd)
```