Code for Output Results

Samuel Sim, Lee Min Shuen

November 2021

Function to calculate the mean of the 100 repetitive simulation runs. Output is the average waiting time for every station across a day.

```r
library(tidyverse)

Calculate_mean_across_replications <- function(df) {

    linetoremove <- df[3, 1]
    df <- df %>%
        select(X1) %>%
        filter(X1 != "") %>%
        filter(X1 != "#####") %>%
        filter(X1 != as.character(linetoremove)) %>%
        separate(X1, sep = "\t", c("SimTime",
            "Bukit Panjang (To Expo)", "Cashew (To Expo)",
            "Hillview (To Expo)", "Beauty World (To Expo)",
            "King Albert Park (To Expo)",
            "Sixth Avenue (To Expo)", "Tan Kah Kee (To Expo)",
            "Botanic Gardens (To Expo)",
            "Stevens (To Expo)", "Newton (To Expo)",
            "Little India (To Expo)", "Rochor (To Expo)",
            "Bugis (To Expo)", "Promenade (To Expo)",
            "Bayfront (To Expo)", "Downtown (To Expo)",
            "Telok Ayer (To Expo)", "Chinatown (To Expo)",
            "Fort Canning (To Expo)", "Bencoolen (To Expo)",
            "Jalan Besar (To Expo)", "Bendemeer (To Expo)",
            "Geylang Bahru (To Expo)", "Mattar (To Expo)",
            "MacPherson (To Expo)", "Ubi (To Expo)",
            "Kaki Bukit (To Expo)", "Bedok North (To Expo)",
            "Bedok Reservoir (To Expo)",
            "Tampines West (To Expo)", "Tampines (To Expo)",
            "Tampines East (To Expo)", "Upper Changi (To Expo)",
            "Expo (To Bukit Panjang)", "Upper Changi (To Bukit Panjang)",
            "Tampines East (To Bukit Panjang)",
            "Tampines (To Bukit Panjang)",
            "Tampines West (To Bukit Panjang)",
            "Bedok Reservoir (To Bukit Panjang)",
            "Bedok North (To Bukit Panjang)",
            "Kaki Bukit (To Bukit Panjang)",
            "Ubi (To Bukit Panjang)", "MacPherson (To Bukit Panjang)",
            "Mattar (To Bukit Panjang)",
            "Geylang Bahru (To Bukit Panjang)",
            "Bendemeer (To Bukit Panjang)",
            "Jalan Besar (To Bukit Panjang)",
            "Bencoolen (To Bukit Panjang)",
            "Fort Canning (To Bukit Panjang)",
            "Chinatown (To Bukit Panjang)",
            "Telok Ayer (To Bukit Panjang)",
            "Downtown (To Bukit Panjang)",
            "Bayfront (To Bukit Panjang)",
            "Promenade (To Bukit Panjang)",
            "Bugis (To Bukit Panjang)", "Rochor (To Bukit Panjang)",
            "Little India (To Bukit Panjang)",
            "Newton (To Bukit Panjang)",
            "Stevens (To Bukit Panjang)",
            "Botanic Gardens (To Bukit Panjang)",
            "Tan Kah Kee (To Bukit Panjang)",
            "Sixth Avenue (To Bukit Panjang)",
            "King Albert Park (To Bukit Panjang)",
            "Beauty World (To Bukit Panjang)",
            "Hillview (To Bukit Panjang)",
            "Cashew (To Bukit Panjang)"))
    df[df == "NaN"] <- "0"
    df <- as.data.frame(sapply(df, as.numeric))
    df <- aggregate(x = df, by = list(df$SimTime),
        FUN = mean)
    results <- subset(df, select = -c(Group.1))
    results
}

# Controls
Control_Weekend <- read_log("Blue_Line_Weekend-ExpressionLogger1.log",
    skip = 13)
Control_Weekday <- read_log("Blue_Line_Weekday-ExpressionLogger1.log",
    skip = 13)

# Lower train capacity
Covid_Weekend <- read_log("Blue_Line_Weekend_Covid-ExpressionLogger1.log",
    skip = 13)
Covid_Weekday <- read_log("Blue_Line_Weekday_Covid-ExpressionLogger1.log",
    skip = 13)

# Lower train interarrival times
LowerTIAT_Weekend <- read_log("Blue_Line_Weekend_LowerTIAT-ExpressionLogger1.log",
    skip = 13)
LowerTIAT_Weekday <- read_log("Blue_Line_Weekday_LowerTIAT-ExpressionLogger1.log",
    skip = 13)

# Perform mean calculation
Control_Weekend_mean <- Calculate_mean_across_replications(Control_Weekend)
Control_Weekday_mean <- Calculate_mean_across_replications(Control_Weekday)
Covid_Weekend_mean <- Calculate_mean_across_replications(Covid_Weekend)
Covid_Weekday_mean <- Calculate_mean_across_replications(Covid_Weekday)
LowerTIAT_Weekend_mean <- Calculate_mean_across_replications(LowerTIAT_Weekend)
LowerTIAT_Weekday_mean <- Calculate_mean_across_replications(LowerTIAT_Weekday)
```

```r
write.csv(Control_Weekend_mean, "Blue_Line_Weekend_mean.csv",
    row.names = F)
write.csv(Control_Weekday_mean, "Blue_Line_Weekday_mean.csv",
    row.names = F)
write.csv(Covid_Weekend_mean, "Blue_Line_Weekend_Covid_mean.csv",
    row.names = F)
write.csv(Covid_Weekday_mean, "Blue_Line_Weekday_Covid_mean.csv",
    row.names = F)
write.csv(LowerTIAT_Weekend_mean, "Blue_Line_Weekend_LowerTIAT_mean.csv",
    row.names = F)
write.csv(LowerTIAT_Weekday_mean, "Blue_Line_Weekday_LowerTIAT_mean.csv",
    row.names = F)
```

Function to calculate the mean of the daily waiting time in a station for each of the 100 simulation. Output is the average daily waiting time in a station for every each simulation. Hence we have to average across the time horizon of a day and across each station. This is to conduct a paired-t test confidence interval to evaluate which system is better.

```r
library(tidyverse)

Calculate_mean <- function(df) {

    linetoremove <- df[3, 1]
    df <- df %>%
        select(X1) %>%
        filter(X1 != "") %>%
        filter(X1 != "#####") %>%
        filter(X1 != as.character(linetoremove)) %>%
        separate(X1, sep = "\t", c("SimTime",
            "Bukit Panjang (To Expo)", "Cashew (To Expo)",
            "Hillview (To Expo)", "Beauty World (To Expo)",
            "King Albert Park (To Expo)",
            "Sixth Avenue (To Expo)", "Tan Kah Kee (To Expo)",
            "Botanic Gardens (To Expo)",
            "Stevens (To Expo)", "Newton (To Expo)",
            "Little India (To Expo)", "Rochor (To Expo)",
            "Bugis (To Expo)", "Promenade (To Expo)",
            "Bayfront (To Expo)", "Downtown (To Expo)",
            "Telok Ayer (To Expo)", "Chinatown (To Expo)",
            "Fort Canning (To Expo)", "Bencoolen (To Expo)",
            "Jalan Besar (To Expo)", "Bendemeer (To Expo)",
            "Geylang Bahru (To Expo)", "Mattar (To Expo)",
            "MacPherson (To Expo)", "Ubi (To Expo)",
            "Kaki Bukit (To Expo)", "Bedok North (To Expo)",
            "Bedok Reservoir (To Expo)",
            "Tampines West (To Expo)", "Tampines (To Expo)",
            "Tampines East (To Expo)", "Upper Changi (To Expo)",
            "Expo (To Bukit Panjang)", "Upper Changi (To Bukit Panjang)",
            "Tampines East (To Bukit Panjang)",
            "Tampines (To Bukit Panjang)",
            "Tampines West (To Bukit Panjang)",
            "Bedok Reservoir (To Bukit Panjang)",
            "Bedok North (To Bukit Panjang)",
            "Kaki Bukit (To Bukit Panjang)",
            "Ubi (To Bukit Panjang)", "MacPherson (To Bukit Panjang)",
            "Mattar (To Bukit Panjang)",
            "Geylang Bahru (To Bukit Panjang)",
            "Bendemeer (To Bukit Panjang)",
            "Jalan Besar (To Bukit Panjang)",
            "Bencoolen (To Bukit Panjang)",
            "Fort Canning (To Bukit Panjang)",
            "Chinatown (To Bukit Panjang)",
            "Telok Ayer (To Bukit Panjang)",
            "Downtown (To Bukit Panjang)",
            "Bayfront (To Bukit Panjang)",
            "Promenade (To Bukit Panjang)",
            "Bugis (To Bukit Panjang)", "Rochor (To Bukit Panjang)",
            "Little India (To Bukit Panjang)",
            "Newton (To Bukit Panjang)",
            "Stevens (To Bukit Panjang)",
            "Botanic Gardens (To Bukit Panjang)",
            "Tan Kah Kee (To Bukit Panjang)",
            "Sixth Avenue (To Bukit Panjang)",
            "King Albert Park (To Bukit Panjang)",
            "Beauty World (To Bukit Panjang)",
            "Hillview (To Bukit Panjang)",
            "Cashew (To Bukit Panjang)"))
    df[df == "NaN"] <- "0"
    df <- as.data.frame(sapply(df, as.numeric))
    df <- rowMeans(df[, 2:67])
    j <- 1
    table <- data.frame()
    for (i in 1:100) {
        average <- mean(df[j:j + 1440 - 1])
        table <- rbind(table, average)
        j <- j + 1440
    }
    table
}

# Perform mean calculation
Control_Weekend_mean2 <- Calculate_mean(Control_Weekend)
Control_Weekday_mean2 <- Calculate_mean(Control_Weekday)
Covid_Weekend_mean2 <- Calculate_mean(Covid_Weekend)
Covid_Weekday_mean2 <- Calculate_mean(Covid_Weekday)
LowerTIAT_Weekend_mean2 <- Calculate_mean(LowerTIAT_Weekend)
LowerTIAT_Weekday_mean2 <- Calculate_mean(LowerTIAT_Weekday)

# Pairwise t test
Z_CovidvsControl_Weekend <- Control_Weekend_mean2 -
    Covid_Weekend_mean2
Z_CovidvsControl_Weekday <- Control_Weekday_mean2 -
    Covid_Weekday_mean2
Z_LowerTIATvsControl_Weekend <- Control_Weekend_mean2 -
    LowerTIAT_Weekend_mean2
Z_LowerTIATvsControl_Weekday <- Control_Weekday_mean2 -
```

```
    LowerTIAT_Weekday_mean2

ci.diff1 <- t.test(Z_CovidvsControl_Weekend,
    conf.level = 0.95)$conf.int
ci.diff2 <- t.test(Z_CovidvsControl_Weekday,
    conf.level = 0.95)$conf.int
ci.diff3 <- t.test(Z_LowerTIATvsControl_Weekend,
    conf.level = 0.95)$conf.int
ci.diff4 <- t.test(Z_LowerTIATvsControl_Weekday,
    conf.level = 0.95)$conf.int

# Control vs Covid(Lower Train
# Capacity)
print(ci.diff1)  # Weekend
print(ci.diff2)  # Weekday

# Control vs Faster train arrivals
print(ci.diff3)  # Weekend
print(ci.diff4)  # Weekday
```

Function to generate graphics for output analysis.

```
library(tidyverse)
library(gganimate)
library(gapminder)

intervals <- seq(0, 24, 0.5)

control_weekday <- read.csv("Blue_Line_Weekday_mean.csv")
lowerCapacity_weekday <- read.csv("Blue_Line_Weekday_Covid_mean.csv")
fasterTrainArrival_weekday <- read.csv("Blue_Line_Weekday_LowerTIAT_mean.csv")

control_weekend <- read.csv("Blue_Line_Weekend_mean.csv")
lowerCapacity_weekend <- read.csv("Blue_Line_Weekend_Covid_mean.csv")
fasterTrainArrival_weekend <- read.csv("Blue_Line_Weekend_LowerTIAT_mean.csv")


create_animation <- function(control, lower_capacity,
    faster_arrivals, day_type) {

    # control
    control_long <- pivot_longer(data = control,
        cols = "Bukit.Panjang..To.Expo.":"Cashew..To.Bukit.Panjang.",
        names_to = "Station", values_to = "Avg.Waiting.Time")

    control_long_direction1 <- subset(x = control_long,
        subset = grepl("To.Expo", control_long$Station))
    control_long_direction1$Station <- gsub("..To.Expo.",
        x = control_long_direction1$Station,
        "")
    control_long_direction1$Station <- factor(control_long_direction1$Station,
        levels = unique(control_long_direction1$Station))
    control_long_direction1$Scenario <- "Control"
```

```r
control_long_direction2 <- subset(x = control_long,
    subset = !grepl("To.Expo", control_long$Station))
control_long_direction2$Station <- gsub("..To.Bukit.Panjang.",
    x = control_long_direction2$Station,
    "")
control_long_direction2$Station <- factor(control_long_direction2$Station,
    levels = unique(control_long_direction2$Station))
control_long_direction2$Scenario <- "Control"

# lower capacity
lower_capacity_long <- pivot_longer(data = lower_capacity,
    cols = "Bukit.Panjang..To.Expo.":"Cashew..To.Bukit.Panjang.",
    names_to = "Station", values_to = "Avg.Waiting.Time")

lower_capacity_long_direction1 <- subset(x = lower_capacity_long,
    subset = grepl("To.Expo", lower_capacity_long$Station))
lower_capacity_long_direction1$Station <- gsub("..To.Expo.",
    x = lower_capacity_long_direction1$Station,
    "")
lower_capacity_long_direction1$Station <- factor(lower_capacity_long_direction1$Station,
    levels = unique(lower_capacity_long_direction1$Station))
lower_capacity_long_direction1$Scenario <- "LowerCapacity"

lower_capacity_long_direction2 <- subset(x = lower_capacity_long,
    subset = !grepl("To.Expo", lower_capacity_long$Station))
lower_capacity_long_direction2$Station <- gsub("..To.Bukit.Panjang.",
    x = lower_capacity_long_direction2$Station,
    "")
lower_capacity_long_direction2$Station <- factor(lower_capacity_long_direction2$Station,
    levels = unique(lower_capacity_long_direction2$Station))
lower_capacity_long_direction2$Scenario <- "LowerCapacity"

# faster arrivals
faster_arrivals_long <- pivot_longer(data = faster_arrivals,
    cols = "Bukit.Panjang..To.Expo.":"Cashew..To.Bukit.Panjang.",
    names_to = "Station", values_to = "Avg.Waiting.Time")

faster_arrivals_long_direction1 <- subset(x = faster_arrivals_long,
    subset = grepl("To.Expo", faster_arrivals_long$Station))
faster_arrivals_long_direction1$Station <- gsub("..To.Expo.",
    x = faster_arrivals_long_direction1$Station,
    "")
faster_arrivals_long_direction1$Station <- factor(faster_arrivals_long_direction1$Station,
    levels = unique(faster_arrivals_long_direction1$Station))
faster_arrivals_long_direction1$Scenario <- "FasterTrainArrivals"

faster_arrivals_long_direction2 <- subset(x = faster_arrivals_long,
    subset = !grepl("To.Expo", faster_arrivals_long$Station))
faster_arrivals_long_direction2$Station <- gsub("..To.Bukit.Panjang.",
    x = faster_arrivals_long_direction2$Station,
    "")
faster_arrivals_long_direction2$Station <- factor(faster_arrivals_long_direction2$Station,
    levels = unique(faster_arrivals_long_direction2$Station))
```

```
faster_arrivals_long_direction2$Scenario <- "FasterTrainArrivals"

# create animation
direction1 <- rbind(control_long_direction1,
    lower_capacity_long_direction1, faster_arrivals_long_direction1)
direction1$Scenario <- as.factor(direction1$Scenario)
direction1$Station <- as.factor(direction1$Station)
direction1 <- subset(direction1, direction1$SimTime %in%
    intervals)

direction2 <- rbind(control_long_direction2,
    lower_capacity_long_direction2, faster_arrivals_long_direction2)
direction2$Scenario <- as.factor(direction2$Scenario)
direction2$Station <- as.factor(direction2$Station)
direction2 <- subset(direction2, direction2$SimTime %in%
    intervals)

anim <- ggplot(direction1, aes(Station,
    Avg.Waiting.Time, color = Scenario),
    palette = c("#00AFBB", "#E7B800",
        "#FC4E07")) + geom_point(alpha = 0.55) +
    labs(title = day_type, subtitle = "Time (h): {frame_time}",
        x = "Station", y = "Avg Waiting Time") +
    theme_bw(base_size = 8) + theme(axis.text.x = element_text(angle = 90,
    margin = margin(1, unit = "cm"),
    hjust = 1, vjust = 1)) + ylim(0,
    2) + # Here comes the gganimate specific bits 2)
    2) + # Here comes the gganimate specific bits +
    2) + # Here comes the gganimate specific bits #
    2) + # Here comes the gganimate specific bits Here
    2) + # Here comes the gganimate specific bits comes
    2) + # Here comes the gganimate specific bits the
    2) + # Here comes the gganimate specific bits gganimate
    2) + # Here comes the gganimate specific bits specific
    2) + # Here comes the gganimate specific bits bits
transition_time(SimTime) + ease_aes("linear")

animate(plot = anim, height = 800, width = 1280,
    res = 150, renderer = gifski_renderer(),
    end_pause = 30)

anim_save(paste0(day_type, "_comparison (Direction 1).gif"))

anim <- ggplot(direction2, aes(Station,
    Avg.Waiting.Time, color = Scenario),
    palette = c("#00AFBB", "#E7B800",
        "#FC4E07")) + geom_point(alpha = 0.55) +
    labs(title = day_type, subtitle = "Time (h): {frame_time}",
        x = "Station", y = "Avg Waiting Time") +
    theme_bw(base_size = 8) + theme(axis.text.x = element_text(angle = 90,
    margin = margin(1, unit = "cm"),
    hjust = 1, vjust = 1)) + ylim(0,
    2) + # Here comes the gganimate specific bits 2)
```

```r
        2) + # Here comes the gganimate specific bits +
        2) + # Here comes the gganimate specific bits #
        2) + # Here comes the gganimate specific bits Here
        2) + # Here comes the gganimate specific bits comes
        2) + # Here comes the gganimate specific bits the
        2) + # Here comes the gganimate specific bits gganimate
        2) + # Here comes the gganimate specific bits specific
        2) + # Here comes the gganimate specific bits bits
    transition_time(SimTime) + ease_aes("linear")

    animate(anim, height = 800, width = 1280,
        res = 150, renderer = gifski_renderer(),
        end_pause = 30)

    anim_save(paste0(day_type, "_comparison (Direction 2).gif"))

}


# Generate for weekday and weekend
create_animation(control_weekday, lowerCapacity_weekday,
    fasterTrainArrival_weekday, "Weekday")
create_animation(control_weekend, lowerCapacity_weekend,
    fasterTrainArrival_weekend, "Weekend")
```