da420\_project5

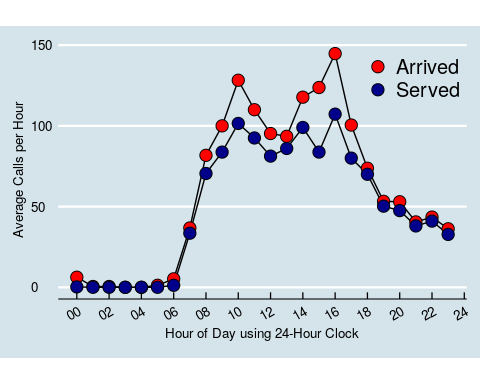
Jason Grahn

2/2/2019

Use R to develop a “Call Center Arrival and Service Rates on Thursday”, similar to Figure 6.7 in page 72. Make sure you include the command lines and the output/results. Interpret the graph in details.

# Workforce Scheduling for Anonymous Bank Call Center (R)  
# source("R\_utility\_program\_3.R") provides split-plotting utilities  
load("mtpa\_split\_plotting\_utilities.Rdata")  
# source("R\_utility\_program\_4.R") provides wait-time ribbon plots  
load("mtpa\_wait\_time\_ribbon\_utility.Rdata")  
  
# focus upon February 1999  
call.center.input.data <- read.table("data\_anonymous\_bank\_february.txt",   
 header = TRUE, colClasses = c("character","integer","numeric",  
 "integer","character","character","character","character","integer",  
 "character","character","integer","factor","character","character",  
 "integer","character"))  
   
# delete PHANTOM calls  
call.center.data <- subset(call.center.input.data, subset = (outcome != "PHANTOM"))  
  
# negative VRU times make no sense... drop these rows from data frame  
call.center.data <- subset(call.center.data, subset = (vru\_time >= 0))  
  
# calculate wait time as sum of vru\_time and q\_time  
call.center.data$wait\_time <-   
 call.center.data$vru\_time + call.center.data$q\_time  
  
# define four-digit year so year is not read as 2099  
# convert date string to date variable   
call.center.data$date <- paste("19", call.center.data$date, sep ="")  
call.center.data$date <- ymd(call.center.data$date)  
  
# identify day of the week 1 = Sunday ... 7 = Saturday  
call.center.data$day\_of\_week <- wday(call.center.data$date)  
call.center.data$day\_of\_week <- factor(call.center.data$day\_of\_week,  
 levels = c(1:7), labels = c("Sunday","Monday","Tuesday",  
 "Wednesday","Thursday","Friday","Saturday"))  
  
# identify the hour of entry into the system  
time.list <- strsplit(call.center.data$vru\_entry,":")  
call.hour <- numeric(nrow(call.center.data))  
for (index.for.call in 1:nrow(call.center.data))   
 call.hour[index.for.call] <- as.numeric(time.list[[index.for.call]][1])  
call.center.data$call\_hour <- call.hour  
  
# select first week of February 1999 for data visualization and analysis  
# that week began on Monday, February 1 and ended on Sunday, February 7  
selected.week <- subset(call.center.data, subset = (date < ymd("19990208")))  
  
# loop for day of week ignoring Saturdays in Isreal  
day.of.week.list <- c("Monday","Tuesday",  
 "Wednesday","Thursday","Friday","Sunday")

# select Wednesdays in February for the queueing model  
thursdays <- subset(call.center.data, subset = (day\_of\_week == "Thursday"))  
  
# compute arrival rate of calls as calls for hour   
# we do not use table() here because some hours could have zero calls  
calls.for.hour <- numeric(24)  
for(index.for.hour in 1:24) {   
# 24-hour clock has first hour coded as zero in input data file  
 coded.index.for.hour <- index.for.hour - 1   
 this.hour.calls <-   
 subset(thursdays, subset = (call\_hour == coded.index.for.hour))   
 if(nrow(this.hour.calls) > 0)   
 calls.for.hour[index.for.hour] <- nrow(this.hour.calls)   
 }  
  
# compute arrival rate as average number of calls into VRU per hour  
hourly.arrival.rate <- calls.for.hour/4 # four Wednesdays in February  
  
# begin by selecting calls that receive service  
thursdays.served <- subset(thursdays, subset = (server != "NO\_SERVER"))  
  
hourly.mean.service.time <- numeric(24)  
served.for.hour <- numeric(24)  
for(index.for.hour in 1:24) {   
# 24-hour clock has first hour coded as zero in input data file  
 coded.index.for.hour <- index.for.hour - 1   
 this.hour.calls <-   
 subset(thursdays.served, subset = (call\_hour == coded.index.for.hour))  
 if(nrow(this.hour.calls) > 0) {  
 served.for.hour[index.for.hour] <- nrow(this.hour.calls)  
 hourly.mean.service.time[index.for.hour] <- mean(this.hour.calls$ser\_time)  
 }  
 }   
   
# hourly service rate given the current numbers of service operators  
hourly.served.rate <- served.for.hour/4 # four Thursdays in February  
  
# build data frame for plotting arrival and service rates  
hour <- 1:24 # hour for horizontal axix of line chart  
type <- rep("Arrived", length = 24)  
value <- hourly.arrival.rate  
arrival.data.frame <- data.frame(hour, value, type)   
type <- rep("Served", length = 24)  
value <- hourly.served.rate  
service.data.frame <- data.frame(hour, value, type)   
arrival.service.data.frame <- rbind(arrival.data.frame, service.data.frame)  
  
plotting.object <- ggplot(data = arrival.service.data.frame,   
 aes(x = hour, y = value, fill = type)) +   
 geom\_line() +  
 geom\_point(size = 4, shape = 21) +  
 ggthemes::theme\_economist() +   
 ggthemes::scale\_color\_economist() +  
 scale\_x\_continuous(breaks = c(1,3,5,7,9,11,13,15,17,19,21,23,25),  
 labels =   
 c("00","02","04","06","08","10","12","14","16","18","20","22","24")) +  
 theme(axis.text.x = element\_text(angle = 30, hjust = 1, vjust = 1)) +  
 labs(x = "Hour of Day using 24-Hour Clock", y = "Average Calls per Hour") +  
 scale\_fill\_manual(values = c("red","dark blue"),   
 guide = guide\_legend(title = NULL)) +  
 theme(legend.position = c(1,1), legend.justification = c(1,1)) +  
 theme(legend.text = element\_text(size=15)) +  
 coord\_fixed(ratio = 1/10)   
  
plotting.object



# Interpretation

My initial reaction to the chart is that few hours are accurately staffed and I would like to see how many agents are on staff at any given hour inside this chart as well - possibly as a secondary Y-axis. If the business rules are such that *zero* calls are abandoned, we would want the dark-blue “Served” line to be equal or greater-than the red “Arrived” line. As we look through the delta we see there are areas more interesting than others. The calls volumes are bimodal, with two **peak** call arrival times; first during the 9am hour and second during the 1600 (4pm) hour. I believe the 9am peak needs specific attention to make sure agents are not overwhelmed, cascading effects through the rest of the day.

The mid-day spread requires attention. Only as people return from lunch and get back to work (1300 / 1pm-ish), do we receive reprieve. However, this happens alongside of what-appears-to-be lunchtime for our own call center agents between 1100 and 1300. Staggering lunches or staffing additional headcount here could alleviate affects and catch us up to the incoming volume. It’s likely that as we lagged behind in calls in the morning, effects have compounded. We observe our two largest deltas at at 1500 (3pm) and 1600 (4pm). While it’s likely need to staff more agents across all daylight hours to overcome the Arrival of calls we need more agents in the afternoon to prevent customer frustration as we work toward closing hours.

We are just *barely* under the requirements between 1900 (7pm) through 0600 (6am); so it’s likely one additional staff could start later in the evening to take up the slack during this shift until midnight. Bringing on another staff member earlier in the morning might be more costly, but it could account for the small delta at 5, 6, and 7am; before we start to hit the larger morning volume.

Additional investigation should be done to determine what drives AM calls versus PM calls; and if we can move these drivers into self-service via the website, or make the questions and answers possible via phone system automation.