Average Appointment Times 2021-2023

Preston

2023-08-11

## Problem

Technician suporvisor feels that certain “technician appointments” are taking longer then the scheduled amount of time.

## Goal

Identify how long the “diabetic consult” times take for technician appointments. Management is also curious about appointment lengths in general and how they can vary from each location (they have 3).

## Libraries

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(skimr)

## Importing and selecting columns needed for analysis

app\_trends <- read\_csv("app\_trends\_2023.csv")

## Rows: 26012 Columns: 14  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (10): patientid, name, description, staffid, datein, dateout, time\_dura...  
## dbl (2): statusid, census\_reason\_id  
## time (2): time\_in, time\_out  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

s\_app\_trends <- app\_trends %>%   
 select(patientid, description, staffid, datein, dateout, census\_reason\_txt, display\_id)  
View(s\_app\_trends)

## Cleaning begins

#### Starting with renaming of columns, cleaning up any whitespace/making things lowercase, and creating columns needed for later analysis.

# renaming column "census\_reason\_txt" to "visit\_details"  
s\_app\_trends <- s\_app\_trends %>%   
 rename(visit\_details = census\_reason\_txt)  
  
# renaming column "display\_id" to "location"  
s\_app\_trends <- s\_app\_trends %>%   
 rename(location = display\_id)  
  
# Convert character columns to lowercase and remove leading/trailing whitespace  
s\_app\_trends <- s\_app\_trends %>%  
 mutate\_if(is.character, tolower) %>%  
 mutate\_if(is.character, str\_trim)  
  
# creating new columns needed for analysis later.  
s\_app\_trends$app\_type <- NA  
s\_app\_trends$app\_subtype <- NA

## Working on date/time columns

#### Removing extreme outliers which are actually errors. The hospital is open Monday 7am and closes Sunday 2pm (8,256 minutes). They do not keep any patients past then so any duration past that time is an error. Another error is if the appointment time is less then a minute.

# Convert "datein" and "dateout" columns to proper date/time format  
s\_app\_trends$datein <- as.POSIXct(s\_app\_trends$datein, format = "%m/%d/%Y %H:%M:%S")  
s\_app\_trends$dateout <- as.POSIXct(s\_app\_trends$dateout, format = "%m/%d/%Y %H:%M:%S")  
  
# making a column of duration in minutes with the "datein" and "dateout" columns and rounding the result.   
# If you want to calculate the duration in hours, use units = "hours" instead  
s\_app\_trends$duration\_min <- as.numeric(difftime(s\_app\_trends$dateout, s\_app\_trends$datein, units = "mins"))  
s\_app\_trends$duration\_min <- round(s\_app\_trends$duration\_min, 0)  
  
# removing errors  
s\_app\_trends <- subset(s\_app\_trends, duration\_min > 1 & duration\_min <= 8256)

## Cleaning “staffid” column

# Each number in "staffid" correlates to a specific doctor, need to remove the "L" and "M" in front of numbers, these   
# signify location of where the doctor is at but we have a separate column for that and just want one ID for the doctor   
s\_app\_trends$staffid <- gsub("[A-Za-z](?=[0-9])", "", s\_app\_trends$staffid, perl = TRUE)  
  
# finding instances of "ltk", "mtk", and "tktk" and changing it to "tk" for the same reason listed above  
s\_app\_trends$staffid <- gsub("l|mtk|tktk", "tk", s\_app\_trends$staffid)  
  
# there are 5 rows that correlate to assistants/technicians but those rows are errors/and or used for in house things  
# 0 & 22 also need removed, 22 is a relief vet where management doesn't want those values messing things up and 0 is in house stuff  
s\_app\_trends <- s\_app\_trends[!(s\_app\_trends$staffid %in% c("ceb", "ed", "mc", "stkk", "vc", "0", "22")), ]

## Cleaning the “description” column

# change na values to unknown  
s\_app\_trends$description <- ifelse(is.na(s\_app\_trends$description), "unknown", s\_app\_trends$description)  
  
# changing different names for the same type of appointment to a consistent name  
s\_app\_trends <- s\_app\_trends %>%  
 mutate(description = ifelse(description %in% c("exam", "exam with vaccines"),  
 "preventative care exam", description))  
s\_app\_trends <- s\_app\_trends %>%  
 mutate(description = ifelse(description %in% c("not wellness"),  
 "sick", description))

## Creating data in “app\_type” column

# I have 6 different categories to make; doctor, tech\_app, admit, other, doctor/admit, and na.  
s\_app\_trends <- s\_app\_trends %>%  
 mutate(app\_type = case\_when(  
 description %in% c("acupuncture", "behavior consultation", "emergency", "euthanasia", "exotic exam", "new client",   
 "new pet", "pre surgical exam", "preventative care exam", "quality of life", "recheck",   
 "senior exam", "sick", "travel health certificate", "suture removal" ) ~ "doctor",  
 description %in% c("cytopoint injection", "diabetic consultation", "discharge", "lab work", "laser",   
 "technician appointment", "tnt" ) ~ "tech\_app",  
 description %in% c("dental", "drop off - admit", "feline castrate", "glucose curve", "jss surgery",   
 "laparascopic surgery", "short procedure", "surgery" ) ~ "admit",  
 description %in% c("boarding", "with dr approval", "scheduling notes") ~ "other",  
 description %in% c("ultrasound") ~ "doctor/admit",  
 TRUE ~ NA\_character\_ # If none of the conditions match, set the value as NA  
 ))  
# "consultation" & "radiology" are words not categorized because they can be either doctor, admit, or tech\_apps.   
# In order to classify those words, I look in the "staffid" column. It's listed as either "tk" for a tech\_app or  
# with a doctors number if it's a doctors number the appointment type could be a range of different things so   
# will set to doctor/admit  
s\_app\_trends$app\_type <- ifelse(is.na(s\_app\_trends$app\_type) & s\_app\_trends$staffid == "tk", "tech\_app",  
 ifelse(is.na(s\_app\_trends$app\_type), "doctor/admit", s\_app\_trends$app\_type))

## Creating data in the “app\_subtype” column

#### For now, this column is simply used to label diabetic consults that are technician appointments and not doctor appointments.

s\_app\_trends <- s\_app\_trends %>%  
 mutate(  
 app\_subtype = ifelse(  
 (grepl("diabetic", description, ignore.case = TRUE) |   
 grepl("diabetic", visit\_details, ignore.case = TRUE)) &  
 app\_type == "tech\_app",  
 "diabetic consult",  
 app\_subtype)  
 )

## Column creation, “details”

#### This column was made to clearly look at different appointment types while eliminating any errors/mixed appointment types

s\_app\_trends <- s\_app\_trends %>%  
 mutate(details\_placeholder = NA\_character\_) %>%  
 mutate(details = case\_when(  
 app\_subtype == "diabetic consult" ~ "diabetic consult",  
 description %in% c("sick", "preventative care exam", "recheck",   
 "senior exam", "new pet", "new client", "emergency", "exotic exam",   
 "suture removal", "quality of life", "tnt", "lab work", "laser") ~ description,  
 app\_type == "tech\_app" & is.na(details\_placeholder) ~ "misc tech appointment",  
 TRUE ~ NA\_character\_  
 )) %>%  
 select(-details\_placeholder) # Remove the placeholder column

## Analysis begins

#### Using median averages to help protect from large outliers

# Finding the average duration.  
median\_appointment\_times <- s\_app\_trends %>%  
 filter(!is.na(details)) %>% # Exclude rows with NA in details column  
 mutate(location\_column = location) %>%  
 group\_by(details, app\_type, location\_column) %>%  
 summarise(median\_duration = median(duration\_min)) %>%  
 pivot\_wider(names\_from = location\_column, values\_from = median\_duration)

## `summarise()` has grouped output by 'details', 'app\_type'. You can override  
## using the `.groups` argument.

# Melt the data for plotting  
melted\_data <- median\_appointment\_times %>%  
 gather(location, average\_time, mpv:met) %>%  
 filter(!is.na(average\_time)) # Exclude NA values

## Bar plot of results

#### Creating date range so that if I use data from different years it will automaticall update title/names of files.

# Calculate the date range from now to 2 years ago  
current\_year <- year(Sys.Date())  
date\_range\_start <- as.Date(paste0(current\_year - 2, "-01-01"))  
date\_range\_end <- Sys.Date()  
  
# Format the date range for the file names  
formatted\_date\_range <- paste(format(date\_range\_start, "%Y"), format(date\_range\_end, "%Y"), sep = "-")  
  
# Create the title with the years  
title\_text <- paste("Average Appointment Times by Category and Location",  
 formatted\_date\_range, sep = " ")  
  
# Create a grouped bar plot with centered title and modified y-axis  
bar\_plot <- ggplot(melted\_data, aes(x = details, y = average\_time, fill = location)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(title = title\_text,  
 x = "Appointment Type", y = "Average Appointment Time (Minutes)") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1),  
 plot.title = element\_text(hjust = 0.5)) + # Centered title  
 scale\_y\_continuous(breaks = seq(0, ceiling(max(melted\_data$average\_time)), 5)) # Modify y-axis breaks

## Exporting the data/bar plot

#### Saving into a folder called “project\_results”.

# Create a folder if it doesn't exist  
if (!dir.exists("project\_results")) {  
 dir.create("project\_results")  
}  
  
# Export the cleaned s\_app\_trends w/ year info in title  
write.csv(s\_app\_trends, file = paste("project\_results/cleaned\_appointment\_times\_", formatted\_date\_range, ".csv", sep = ""), row.names = FALSE)  
  
# Export the median\_appointment\_times w/ year info in title  
write.csv(median\_appointment\_times, file = paste("project\_results/average\_appointment\_times\_", formatted\_date\_range, ".csv", sep = ""), row.names = FALSE)  
  
# Set the file path for saving the bar plot  
image\_path <- file.path("project\_results", paste("bar\_plot\_average\_appointment\_times\_", formatted\_date\_range, ".png", sep = ""))  
# Save the bar plot as a high quality image w/ year info in title  
ggsave(filename = image\_path, plot = bar\_plot, width = 10, height = 6, dpi = 400, bg = "white")