From Subway to Shelter: An Exploratory Analysis of the Relation Between Subway Station Location and Shelter Occupancy Rate

Paul Hodgetts & Thomas Rosenthal

Abstract

An exploratory data analysis was performed to examine if homeless shelter occupancy increases if the shelter is near a subway station. Using Open Data Toronto's **Daily Shelter Occupancy, 2019** dataset and TTC subway locations, this relationship is explored in terms of whether shorter distance from a subway station can positively correlate to higher occupancy rates of shelters. A bivariate logistic regression was used to measure the variance in shelter occupancy rates by minimum distance from a subway station. A small, positive relationship was found at a significance level of $p < .05$. However, due to the positive skew of the data and possible covariates, conclusions from this regression have not been drawn at this time. Additionally, conclusions have been avoided with consideration to population vulnerability. Data processes and limitations are discussed in some depth and are intended for additional investigation and study replication.

Introduction

We all seek a place of rest and shelter. At the end of a long day, home is often on the mind. But for the vulnerable population of a city, home may not be an option. Shelters aim to offer this population places of rest and respite. Shelters may serve as a temporary holdover until something more permanent is found, or as a known place of shelter when the larger world offers little comfort. However, people in this population may not always have a personal means of transportation to access these spaces and thus rely on public transit systems within close proximity to make use of these shelters. As such, this report aims to explore the relationship between subway stations and shelter occupancy through the use of an exploratory data analysis.

Research Question

Following the intent of this report, the posed question is: does the distance from the nearest TTC subway station influence the rate of occupancy of Toronto shelters?

Limitations and Ethics

A major limitation of the approach of this analysis is that only TTC subway stations were used as a means of measuring access to shelters by public transit. It may be that the shelters furthest from a subway station are served by other forms of transit (i.e. GO trains, streetcars, or buses). This approach has the implicit assumption that the people using Toronto shelters also use the subway as a consistent means of transportation. Indeed, it may be the case that the subway is one of the least used methods of transportation for this population.

Moreover, it may be the case that a given starting point of travel does not necessitate the use of a subway. For instance, because subway stations tend to be clustered around major social centres (e.g. schools, hospitals, libraries), people using shelters may be using social services in the area aside from the subway. Domain knowledge of typical transportation use and shelter use is necessary for drawing further conclusions, as it would aid in the efforts to understand how different shelter types are used and the populations they serve.

This analysis does not consider shelter turnover (see §Ethics). As such, it may be the case that shelters shown to be at or above capacity were so because the people using them were there for extended periods of time. In this regard, measuring occupancy rates is limited without further understanding of the length of time people may reside in a given shelter.

Regarding ethics, first and foremost, this data analysis examines a vulnerable population. Any conclusions drawn about this population should be done so with great care and consideration towards the people who make up the population. Additionally, examining turnover would require identifying and tracking individuals in and out of shelters. This dataset did not provide this data and we strongly discourage any efforts to analyze turnover without full ethics approval and appropriate qualifications.

Exploratory Analysis

Datasets

The datasets used in the following exploratory analysis are the **Daily Shelter Occupancy, 2019** and the **TTC Subway Delay Data, January 2018**. Only TTC subway station names were extracted from the latter. This analysis focused on daily shelter occupancy rates in 2019 only, though 2017 and 2018 were also available from Open Data Toronto.

Data Cleaning

As the base data source for Principal Component Analysis, the Daily Shelter Occupancy, 2019 dataset was quite clean. Using the visdat package, mistyped columns (character to integer) and missing values were resolved. Missing values were for days where shelter CAPACITY was unknown (126 of 39429 records). Duplicates (28) were removed with the get_dupes function from the janitor package.

Some additional columns were derived: 1) From OCCUPANCY_DATE, month and day of week by use of the SQLDF package and SQLite query syntax; 2) From OCCUPANCY and CAPACITY, rate [of occupancy], where rate is the OCCUPANCY/CAPACITY.

As the supplement data source, the TTC subway data contained 39 (104 of 75) erroneous or duplicated subway stations. These were manually resolved, as it was relatively quick to check the 75 subway stations within the city. In hindsight, it might have been easier to pull from a cleaner data source, such as a web call to Wikipedia or TTC's Stations_A-Z.jsp.

With these cleaned data sources, the gmapsdistance package queried Google Maps through our Google Cloud Developer API, in order to derive the distance and the time to shelters from subway stations. The gmapsdistance package requires vectorized origins and destinations and returns a cartesian product: 4575 API calls were necessary to return 61 shelters as destinations and 75 subway stations as origins. The returned product includes TIME, DISTANCE, and STATUS. Status was discarded. Time and Distance were set as "walking" values. Though Time and Distance represent two distinct real-world measurements, they are mathematically corollary. The minimum Distance and Time was applied to each shelter and the station name was added as a supplement.

While working to explore the relationship between distance and occupancy, occupancy was made into a binary variable in which a shelter was rated as either "full" or "not full". Full was defined as 99% to 101% and not full was defined as below 99% and above 101%. This attribute was utilized for bivariate binomial logistic regression models.

Descriptive Statistics & Final Dataset Shape

In *Table 1*, the mean capacity for shelters was $M = 67$, with a standard deviation of $SD = 111$, while the occupancy for shelters had a mean of $M = 63$, with a standard deviation of $SD = 96$. Importantly, the mean rate of occupancy was $M = 0.96$ with a standard deviation of $SD = 0.15$. This means Toronto shelters experienced a high rate of occupancy throughout the year. Looking at the mean minimum distance, shelters are an average $M = 1653$ metres (1.653 km) away from a subway station with a standard deviation of $SD = 1838$ metres (1.838 km). However, examining *Chart 1*, we can see that the data was clustered between 0 km and 2.5 km, causing a positive skew, and thus better served by using the mode as a measure of central tendency.

Table 1: Data summary

Name	dataset_base
Number of rows	39316
Number of columns	14
Column type frequency:	
character	7
numeric	7
Group variables	

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
OCCUPANCY_DATE	0	1	19	19	0	365	0
day_of_week	0	1	3	3	0	7	0
SHELTER_NAME	0	1	9	32	0	56	0
SHELTER_ADDRESS	0	1	12	32	0	56	0
SHELTER_CITY	0	1	7	11	0	4	0
SECTOR	0	1	3	8	0	5	0
nearest_station	0	1	12	22	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
month	0	1	6.56	3.45	1	4.00	7	10	12.00	
week	0	1	25.88	15.07	0	13.00	26	39	52.00	
OCCUPANCY	0	1	63.45	99.57	0	17.00	40	71	826.00	
CAPACITY	0	1	67.09	111.29	1	18.00	40	74	902.00	
rate	0	1	0.96	0.15	0	0.95	1	1	1.77	
minimum_time	0	1	1252.70	1403.25	89	454.00	823	1347	6904.00	
minimum_distance	0	1	1653.05	1837.81	124	623.00	1079	1776	8995.00	

Table 1

To more closely examine the relationship between distance from a subway station and shelter occupancy rates, a logistic regression was performed using a binomial model, with shelter rating ("full" or "not full") as the dependent variable and minimum distance as the independent variable. Looking at *Table 2*, with a B_0 = -0.238 and a B_1 = 0.0000131, it can be understood that with each point increase in distance the predicted log odds of the shelter occupancy being full increased by 0.0000131. This relationship was significant at $p = 0.0177$. As a related exploration, the relationship between the closest and furthest shelters from subway stations and those shelters' occupancy rates were investigated. This relationship can be seen in *Chart 2*, where nearly all of the top 10% (n=7) of shelters furthest from subway stations operate at or above capacity, in contrast to the top 10% (n=7) of shelters closest to subways stations operating below total shelter capacity. Confounding variables (e.g. shelter capacity and other forms of transit) and data skewness may explain the significance of this relationship.

term	estimate	std.error	statistic	p.value
(Intercept)	-0.2376420	0.0136457	-17.415127	0.0000000
minimum_distance	0.0000131	0.0000055	2.372854	0.0176512

Table 2

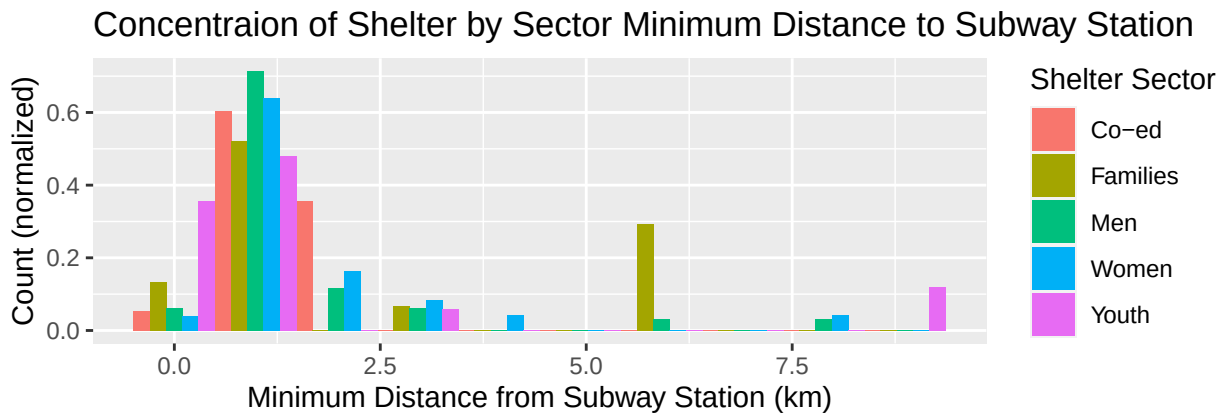

Chart 1

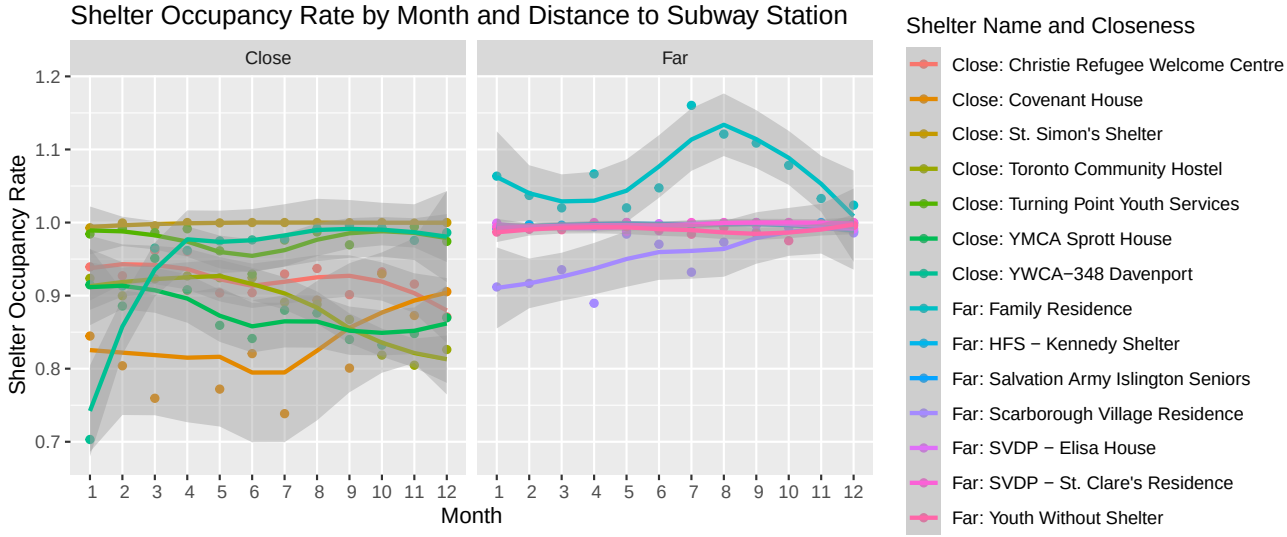

Chart 2

Conclusion

The aim of this analysis was to answer the question: does the distance from any nearest TTC subway station influence the rate of occupancy of Toronto shelters? By combining both Daily Shelter Occupancy for 2019 and TTC subway locations, we explored this relationship by measuring the nearest subway station to each shelter. From this dataset, shelter occupancy rate was considered as a binary variable ("full" or "not full") to correlate distance between subway stations and shelters' occupancy rate by use of bivariate logistic regression.

As mentioned within §Ethics, we do not feel this question has been answered by this exploratory data analysis. While a significant, small, positive relationship was revealed, confounding variables may explain the significance of this relationship. Special consideration has been made to maintain the general structure and design of the original dataset in order to allow domain experts to supplement additional confounding factors.

Finally, we note that this analysis revealed that shelters are often operating at capacity regardless of distance from a subway station.

References

Achim Zeileis, Torsten Hothorn (2002). Diagnostic Checking in Regression Relationships. R News 2(3), 7-10. URL https://CRAN.R-project.org/doc/Rnews/

David Robinson and Alex Hayes (2019). broom: Convert Statistical Analysis Objects into Tidy Tibbles. R package version 0.5.2. https://CRAN.R-project.org/package=broom

G. Grothendieck (2017). sqldf: Manipulate R Data Frames Using SQL. R package version 0.4-11. https://CRAN.R-project.org/package=sqldf

Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL http://www.jstatsoft.org/v40/i03/.

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2019). dplyr: A Grammar of Data Manipulation. R package version 0.8.3. https://CRAN.R-project.org/package=dplyr

Hadley Wickham (2019). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.4.0. https://CRAN.R-project.org/package=stringr

Paul Poncet (2019). bazar: Miscellaneous Basic Functions. R package version 1.0.11. https://CRAN.R-project.org/package=bazar

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rodrigo Azuero Melo & & Demetrio Rodriguez T & David Zarruk (2018). gmapsdistance: Distance and Travel Time Between Two Points from Google Maps. R package version 3.4. https://CRAN.R-project.org/package=gmapsdistance

Sam Firke (2019). janitor: Simple Tools for Examining and Cleaning Dirty Data. R package version 1.2.0. https://CRAN.R-project.org/package=janitor

Sharla Gelfand (2019). opendatatoronto: Access the City of Toronto Open Data Portal. R package version 0.1.1. https://CRAN.R-project.org/package=opendatatoronto

Tierney N (2017). "visdat: Visualising Whole Data Frames." *JOSS*, *2*(16), 355. doi: 10.21105/joss.00355 (URL: https://doi.org/10.21105/joss.00355), <URL: http://dx.doi.org/10.21105/joss.00355>.

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

Appendices

Appendix A

```
library(opendatatoronto)
library(dplyr)
library(tidyverse)
library(sqldf)
library(stringr)
library(gmapsdistance)
library(bazar)
library(lubridate)
library(janitor)
library(lmtest)
library(visdat)
library(broom)
library(skimr)

#builds the base tables from opendatatoronto
# Search packages (this returns a table)
search_packages <- search_packages("shelter")

shelters <- search_packages %>%
  filter(title == "Daily Shelter Occupancy") # Only keep the row(s) where the title is "Parking Tickets"

shelter_occupancy_2019 <- shelters %>% # Start with the package
  list_package_resources() %>% # List the resources in the package
  filter(name == "daily-shelter-occupancy-2019-csv") %>% # Only keep the resource we want
  get_resource()

shelter_occupancy_all_base <- shelter_occupancy_2019

delay_packages <- search_packages("delay")
subway_delay_package <- delay_packages %>%
  filter(title == "TTC Subway Delay Data") # Only keep the row(s) where the title is "TTC Subway Delay Da
subway_jan2018 <- subway_delay_package %>% # Start with the package
  list_package_resources() %>% # List the resources in the package
  filter(name == "ttc-subway-delay-january-2018") %>% # Only keep the resource we want
  get_resource()

shelter_occupancy_combined <- shelter_occupancy_all_base %>%
  unite(addr_city, c(SHELTER_ADDRESS, SHELTER_CITY), sep = "+", remove = FALSE)

#begin to add the API call details
#let's get the 75 subway stations from the delays data, just because we know this data, many other optio
distinct_subways = subway_jan2018 %>% distinct(Station)
distinct_subways

#write_csv(distinct_subways,"subways.csv")

#we manipulated this manually, removed a few junk rows, from ~103to 75, compared to wikipedia

#cleaned_subways <- read_csv("subways_clean.csv")
```

```
distinct_shelters = shelter_occupancy_combined %>% distinct(addr_city)
distinct_shelters

#this is the API call to google maps, do not rerun
#we have removed our google key "KEY"

# set.api.key("KEY")
# origin  =  c(gsub(' ','+', distinct_shelters$addr_city,))
# destination  =  c(gsub(' ','+', cleaned_subways$Station,))


#results are the outcomes, we wrote them to the the following csvs

#results = gmapsdistance(origin, destination, mode = "walking", shape= "long")
#results
#write_csv(results$Distance,"shelter_to_subway.csv")
#write_csv(results$Time,"time_shelter_to_subway.csv")


#csv_time and csv_distance were products of the API run

csv_time <- as.data.frame(read_csv("time_shelter_to_subway.csv"))
csv_dist <- as.data.frame(read_csv("shelter_to_subway.csv"))
output_cartesiancsv<- merge(csv_time,csv_dist,by=c("or","de") )
output_cartesiancsv

time <- as.data.frame(tapply(output_cartesiancsv$Time, output_cartesiancsv$or, min))
dist <- as.data.frame(tapply(output_cartesiancsv$Distance, output_cartesiancsv$or, min))

minimum_values <- merge(time,dist,by=0) %>%
  rename("Shelter_addr" = "Row.names") %>%
  rename("minimum_time" = "tapply(output_cartesiancsv$Time, output_cartesiancsv$or, min)" ) %>%
  rename("minimum_distance" = "tapply(output_cartesiancsv$Distance, output_cartesiancsv$or, ")
minimum_values

nearest_station <- #merge(minimum_values,output_cartesiancsv, by=c("Shelter_addr","minimum_time","minimum
  inner_join(minimum_values,output_cartesiancsv, by=c(c("Shelter_addr" ="or"),c("minimum_time"="Time")))
nearest_station


#start to make the final dataset, clean up,  add relevent columns
shelter_occupancy_merge <-
  shelter_occupancy_combined %>%
  unite(addr_city, c(SHELTER_ADDRESS, SHELTER_CITY), sep = "+", remove = FALSE) %>%
  mutate(Shelter_addr = c(gsub(' ','+', addr_city,)))


dataset_prep <- merge(shelter_occupancy_merge,nearest_station,by = "Shelter_addr") %>%
  rename("nearest_station" = "de") %>%
  select("OCCUPANCY_DATE","SHELTER_NAME","SHELTER_ADDRESS","SHELTER_CITY","SECTOR","OCCUPANCY","CAPACITY"
         "minimum_time","minimum_distance","nearest_station") %>%
  mutate(rate = OCCUPANCY/CAPACITY)

dataset_base <- sqldf("select *, replace([nearest_station],'+',' ') as neareststation
```

```
                      ,strftime('%m', OCCUPANCY_DATE) AS month
                      ,strftime('%W', OCCUPANCY_DATE) AS week
                      ,substr('SunMonTueWedThuFriSat', 1 + 3*strftime('%w', OCCUPANCY_DATE), 3) AS day_of
                      from dataset_prep") %>%
  select("OCCUPANCY_DATE","month","week","day_of_week","SHELTER_NAME","SHELTER_ADDRESS","SHELTER_CITY","S
         ,"OCCUPANCY","CAPACITY","rate","minimum_time","minimum_distance","neareststation") %>%
  rename("nearest_station" = "neareststation")

vis_dat(dataset_base)

 #Find and remove duplicate data
 get_dupes(dataset_base)
 dataset_base <- dataset_base %>% distinct()

dataset_base$month <- as.integer(as.character(dataset_base$month))
dataset_base$week <- as.integer(as.character(dataset_base$week))
dataset_base$minimum_time <- as.integer(as.character(dataset_base$minimum_time))
dataset_base$minimum_distance <- as.integer(as.character(dataset_base$minimum_distance))
dataset_base <- na.omit(dataset_base)

skim(dataset_base)


dataset_base %>%
  group_by(OCCUPANCY,CAPACITY) %>%
  summarize(n())


logi_data <- dataset_base %>%
  mutate(isfull = case_when(between(rate, .99,1.01) ~ 'full'
                            , rate > .99 ~ 'notfull'
                            , rate < 1.01~ 'notfull'))

logi_data$isfull <- as.factor(logi_data$isfull)

logi_data %>%
  group_by(isfull) %>%
  summarize(n())

glm.occ <- glm(isfull ~ minimum_distance, binomial, data = logi_data)
summary(glm.occ)

tidy(glm.occ)

p1 <- ggplot(dataset_base) +
  geom_histogram(aes(x = minimum_distance/1000, y= ..density.., fill = SECTOR), position = 'dodge', bins
  labs(title = "Concentraion of Shelter by Sector Minimum Distance to Subway Station",
       caption = "Chart 1",
       x = "Minimum Distance from Subway Station (km)",
       y = "Count (normalized)",
       fill = "Shelter Sector") +
  theme(plot.caption = element_text(hjust = 0, face = "italic"))

subway_freq <- dataset_base %>%
```

```r
  select(SHELTER_NAME, nearest_station) %>% distinct %>%
  group_by(nearest_station) %>%
  filter( n() >= 6) %>%
  summarise(n())

top_subway <-
inner_join(dataset_base, subway_freq, by = "nearest_station") %>%
  select(SHELTER_NAME, month, rate, nearest_station) %>%
  group_by(SHELTER_NAME,month,nearest_station) %>%
  summarise(mean(rate, na.rm = TRUE)) %>%
  rename(mean = "mean(rate, na.rm = TRUE)")
top_subway

furthest_shelter <- dataset_base %>%
  select(SHELTER_NAME, minimum_distance ) %>% distinct %>%
  arrange(-minimum_distance) %>%
  slice(1:7)
furthest_shelter <-   sqldf("select *,'Far' as [rank] from furthest_shelter")

closest_shelter <- dataset_base %>%
  select(SHELTER_NAME, minimum_distance ) %>% distinct %>%
  arrange(minimum_distance) %>%
  slice(1:7)
closest_shelter <- sqldf("select *,'Close' as rank from closest_shelter")

max_rate_furthest <-
inner_join(dataset_base, rbind(closest_shelter,furthest_shelter), by = "SHELTER_NAME") %>%
  select(SHELTER_NAME, month, rate,rank) %>%
  group_by(SHELTER_NAME,month,rank) %>%
  summarise(mean(rate, na.rm = TRUE)) %>%
  rename(mean = "mean(rate, na.rm = TRUE)")
max_rate_furthest <-  sqldf("select *,[rank] || ': ' || [SHELTER_NAME]  as [shelternamе-rank] from max_ra

p2 <- max_rate_furthest %>%
  group_by(month,rank) %>%
  ggplot(aes(month, mean, colour = `shelternamе-rank`)) +
  geom_point() +
  geom_smooth(method = "loess") +
  facet_grid(~rank) +
  labs(title = "Shelter Occupancy Rate by Month and Distance to Subway Station",
       caption = "Chart 2",
       x = "Month",
       y = "Shelter Occupancy Rate",
       colour = "Shelter Name and Closeness") +
  scale_x_discrete(limits = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)) +
  theme(plot.caption = element_text(hjust = 0, face = "italic"))

p1
p2
```