Setup

```
# libraries
library(sf)
```

```
## Linking to GEOS 3.10.2, GDAL 3.4.1, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)

# data sets
restaurants = read.csv("restaurants.csv")
movements = read.csv("movements.csv")

# function for calculating buffer based on an input in sq. footage (circle->radius in feet->meters->coo
calc_buffer <- function(size)
{
  return((((size / pi)^(0.5))*0.3048)/111111)
}
```

Cleaning up movements

```
# Sort by datetime
movements <- movements %>%
  mutate(datetime = as.POSIXct(datetime, format = "%Y-%m-%d %H:%M:%S")) %>%
  arrange(datetime)

# Filter duplicates
movements <- movements %>%
  mutate(err = if_else(datetime > lag(datetime, default = as.POSIXct("2019-12-31 23:59:59", format = "%
  filter(err == 0)
```

EDA

```r
# Categories of restaurants
rest_types <- restaurants %>%
  count(Category)
rest_types
```

```
##                                        Category n
## 1                            Convenience Stores 3
## 2                      Fruit & Vegetable Markets 4
## 3                      Full-Service Restaurants 7
## 4                   Limited-Service Restaurants 5
## 5 Supermarkets/Other Grocery (Exc Convenience) Strs 2
## 6                  Warehouse Clubs & Supercenters 4
```

```r
restaurants
```

```
##    Restaurant.ID                   Name
## 1           R000      Pullman Fine Dine 0
## 2           R001     Pullman Quick Eats 1
## 3           R002   Pullman Fresh Market 2
## 4           R003      Pullman Fine Dine 3
## 5           R004      Pullman Fine Dine 4
## 6           R005      Pullman Groceries 5
## 7           R006    Pullman Supercenter 6
## 8           R007   Pullman Fresh Market 7
## 9           R008     Pullman Quick Eats 8
## 10          R009     Pullman Quick Stop 9
## 11          R010   Pullman Supercenter 10
## 12          R011     Pullman Groceries 11
## 13          R012 Pullman Fresh Market 12
## 14          R013    Pullman Quick Eats 13
## 15          R014     Pullman Fine Dine 14
## 16          R015    Pullman Quick Stop 15
## 17          R016 Pullman Fresh Market 16
## 18          R017   Pullman Supercenter 17
## 19          R018    Pullman Quick Stop 18
## 20          R019     Pullman Fine Dine 19
## 21          R020   Pullman Supercenter 20
## 22          R021     Pullman Fine Dine 21
## 23          R022    Pullman Quick Eats 22
## 24          R023    Pullman Quick Eats 23
## 25          R024     Pullman Fine Dine 24
##                                         Category Longitude Latitude
## 1                        Full-Service Restaurants -122.2725 47.65121
## 2                     Limited-Service Restaurants -122.2994 47.58553
## 3                       Fruit & Vegetable Markets -122.4015 47.60990
## 4                        Full-Service Restaurants -122.2828 47.64479
## 5                        Full-Service Restaurants -122.3048 47.60799
## 6  Supermarkets/Other Grocery (Exc Convenience) Strs -122.4267 47.57224
## 7                  Warehouse Clubs & Supercenters -122.3335 47.67040
## 8                       Fruit & Vegetable Markets -122.3818 47.54453
## 9                     Limited-Service Restaurants -122.3532 47.56739
## 10                             Convenience Stores -122.3389 47.57299
## 11                    Warehouse Clubs & Supercenters -122.3201 47.56489
## 12 Supermarkets/Other Grocery (Exc Convenience) Strs -122.3717 47.68925
## 13                         Fruit & Vegetable Markets -122.3508 47.59793
## 14                      Limited-Service Restaurants -122.3179 47.59217
## 15                         Full-Service Restaurants -122.4096 47.59302
## 16                               Convenience Stores -122.3068 47.56951
## 17                        Fruit & Vegetable Markets -122.3586 47.58226
## 18                   Warehouse Clubs & Supercenters -122.3270 47.54416
## 19                               Convenience Stores -122.3164 47.61851
## 20                         Full-Service Restaurants -122.3521 47.60202
## 21                   Warehouse Clubs & Supercenters -122.3028 47.60313
## 22                         Full-Service Restaurants -122.3107 47.62365
## 23                      Limited-Service Restaurants -122.3747 47.59370
## 24                      Limited-Service Restaurants -122.2835 47.66092
## 25                         Full-Service Restaurants -122.4359 47.61980
```

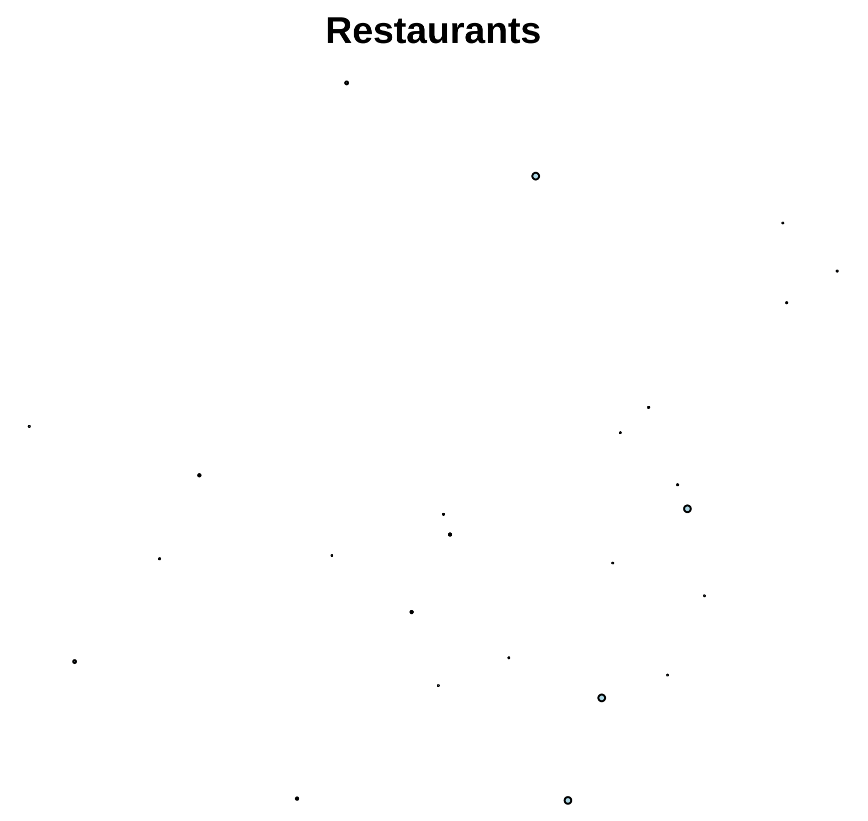
```
# Time range
#print(first(movements$datetime))
#print(last(movements$datetime))
#print(difftime(last(movements$datetime), first(movements$datetime)))
```

Parameters

```
# buffers for categories of restaurants (converted from ft. sq.)
buffer_full = calc_buffer(5000)
buffer_limited = calc_buffer(2000)
buffer_market = calc_buffer(22500)
buffer_supermarc = calc_buffer(33360)
buffer_warehouse = calc_buffer(187000)
buffer_conv = calc_buffer(2000)

# threshold for considering a new visit (in minutes)
time_gap_threshold <- 5
```

Convert restaurant set into sf object and add buffers (dependent on category)

```
# Convert restaurants to sf object. Keep "Name" and "Category" variables. Use Long and Lat for coordina
rest_sf <- st_as_sf(restaurants[, c("Name", "Category", "Longitude", "Latitude")], coords = c("Longitud

# Add column for appropriate buffers then apply them to the object
rest_sf <- rest_sf %>%
  mutate(buffer = case_when(
    Category == "Full-Service Restaurants" ~ buffer_full,
    Category == "Limited-Service Restaurants" ~ buffer_limited,
    Category == "Fruit & Vegetable Markets" ~ buffer_market,
    Category == "Supermarkets/Other Grocery (Exc Convenience) Strs" ~ buffer_supermarc,
    Category == "Warehouse Clubs & Supercenters" ~ buffer_warehouse,
    Category == "Convenience Stores" ~ buffer_conv
  ))
rest_sf <- st_buffer(rest_sf, dist = rest_sf$buffer)

# Plot of restaurants with buffers (!!!!! ADD LABELS AND STUFF !!!!!!!)
plot(select(rest_sf, -Category, -buffer), col = "lightblue", main = "Restaurants")
```



For each point in the movement set, find which (if any) restaurant's buffer it is within.

- Creates a data frame of TRUE and FALSE where movements are observations and each restaurant's buffer is a variable

```
# Use long and lat in movements data set against restaurant buffers using st_within
movements_within <- as.data.frame(st_within(st_as_sf(movements, coords = c("longitude", "latitude")), r
```

Reduce movements_within to only contain id of restaurant within or 0 if not within any. Bind the datetime as well.

```
# If whole row is FALSE, set to 0, otherwise set to index of TRUE. Convert to a data frame.
movements_within <- data.frame(location = ifelse(rowSums(movements_within) == 0, 0, max.col(movements_wi

# Bind datetime from movements set (convert to POSIX as well)
movements_within <- cbind(movements_within, datetime = as.POSIXct(movements$datetime))
```

View visits

```
head(movements_within)
```

```
##   location            datetime
## 1        0 2020-01-01 00:00:00
## 2        0 2020-01-01 00:01:07
## 3        0 2020-01-01 00:02:30
## 4        0 2020-01-01 00:03:40
## 5        0 2020-01-01 00:05:07
## 6        0 2020-01-01 00:05:54
```

Cleanup for memory

```
##             used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells    984860  52.6   48463348 2588.3  53591116 2862.1
## Vcells  20604623 157.3  299220204 2282.9 315876734 2410.0
```

Collect consecutive time spent in a location into a single “visit”

```
# Create data frame for visits
visits <- data.frame(loc = numeric(0), start = character(0), end = character(0), stringsAsFactors = FALS

# New visit starts when prev location is different from current then use cumsum to give each an ID
movements_within <- movements_within %>%
  mutate(new_visit = location != lag(location, default = FALSE))
movements_within <- movements_within %>%
  mutate(visit_id = cumsum(new_visit))

# Filter out non-visits and reduce the group forming a visit (by visit_id) into location, start datetim
visits <- movements_within %>%
  filter(location != 0) %>%
  group_by(visit_id) %>%
  summarize(loc = first(location), start = first(datetime), end = last(datetime)) %>%
  ungroup() %>%
  select(-visit_id)
```

For consecutive visits to the same location, if the time between visits in under threshold (in minutes), merge into one.

- Could be stepping out to car, or to smoke, etc.

```
# Check if location of next visit is same location And if the gap in time is under threshold.
# If yes, time_gap = difftime, otherwise make it 0. If a gap is present, mark the leading row for remov
# and replace the end time with proper end. Then remove redundant (marked) and reduce to loc, start, en
visits <- visits %>%
  mutate (time_gap = ifelse(loc == lead(loc) & difftime(lead(start), end, units = "mins") <= time_gap_tl
                    difftime(lead(start), end, units = "mins"), 0),
          rm = ifelse(lag(time_gap, default = 0) > 0, 1, 0),
          end = if_else(time_gap > 0, as.POSIXct(lead(end)), end)
         ) %>%
  filter(rm == 0) %>%
  select(-rm)
```

Visualize visits

```
# Column for visit length
visits <- mutate(visits, length = round(difftime(end, start, units = "mins"), 2))

# Filter out 0 length visits
visits <- filter(visits, visits$length > 0)

# Replace loc with restaurant names and add categories from the data set
visits$Category <- restaurants$Category[visits$loc]
visits$loc <- restaurants$Name[visits$loc]

# Tabular of min max mean stdev of visit by location, category, etc.
print("Visit Stats By Category")
```

```
## [1] "Visit Stats By Category"
```

```
category_stats <- visits %>%
  group_by(Category) %>%
  summarize(
    min = min(length),
    max = max(length),
    avg = mean(length),
    sdev = sd(length),
    total = n()
  )
print(category_stats)
```

```
## # A tibble: 6 x 6
##   Category                                          min   max   avg    sdev total
##   <chr>                                           <drt> <drt> <drt>  <dbl> <int>
## 1 Convenience Stores                              0.07~ 11.6~ 2.39~   1.55 16797
## 2 Fruit & Vegetable Markets                          0.02~ 23.7~ 3.47~  2.40 41955
## 3 Full-Service Restaurants                           0.02~ 64.5~ 2.64~  1.74 53088
## 4 Limited-Service Restaurants                        0.03~ 11.6~ 2.39~  1.54 28188
## 5 Supermarkets/Other Grocery (Exc Convenience) St~ 0.18~ 25.4~ 3.84~  2.75 22160
## 6 Warehouse Clubs & Supercenters                     0.18~ 73.4~ 7.41~  6.14 47383
```

```
# Tabular of min max mean stdev of visit by location, category, etc.
print("Visit Stats By Location")
```

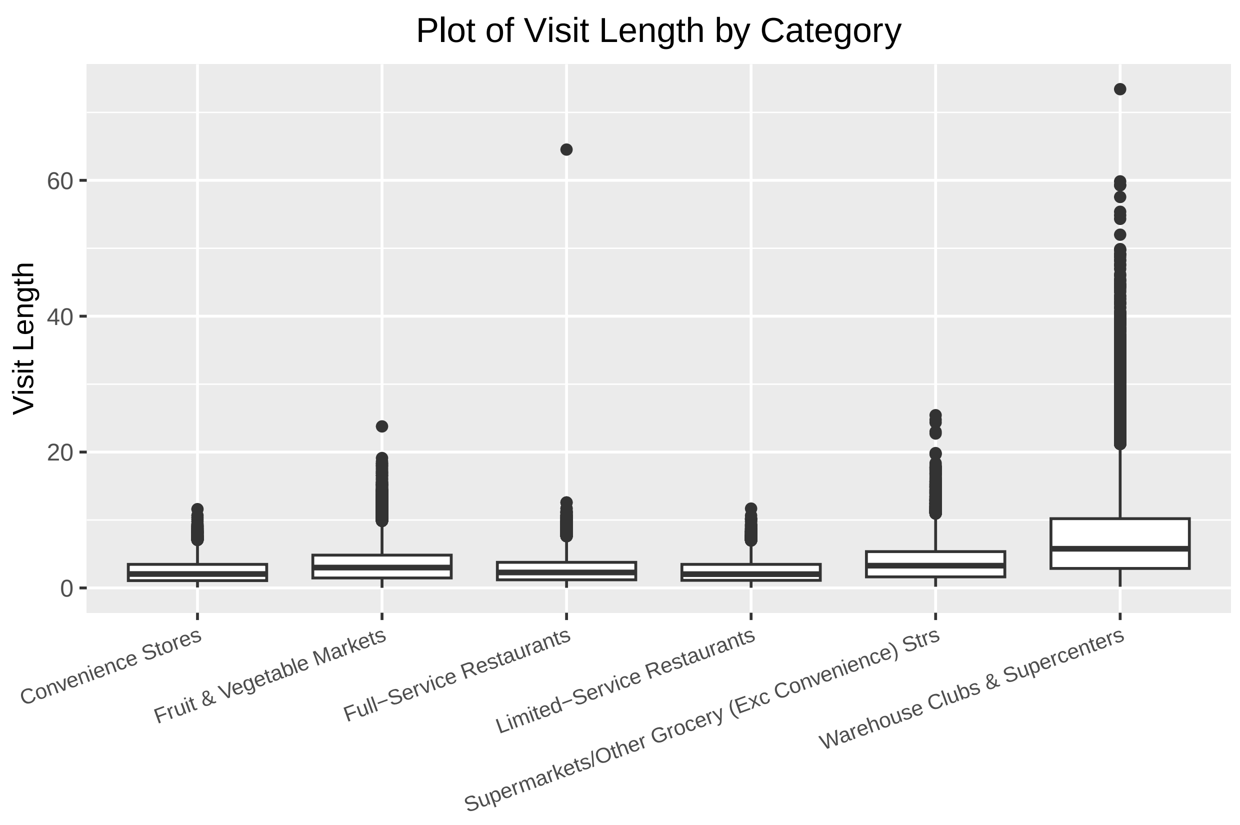
```
## [1] "Visit Stats By Location"
```

```
loc_stats <- visits %>%
  group_by(loc) %>%
  summarize(
    min = min(length),
    max = max(length),
    avg = mean(length),
    sdev = sd(length),
    total = n()
  )
print(loc_stats)
```

```
## # A tibble: 25 x 6
##    loc                    min       max        avg            sdev total
##    <chr>                  <drtn>    <drtn>     <drtn>         <dbl> <int>
##  1 Pullman Fine Dine 0    0.18 mins 11.08 mins 2.642750 mins  1.74  7337
##  2 Pullman Fine Dine 14   0.05 mins 11.38 mins 2.669903 mins  1.74  7764
##  3 Pullman Fine Dine 19   0.07 mins 11.70 mins 2.626833 mins  1.71  7670
##  4 Pullman Fine Dine 21   0.02 mins 12.58 mins 2.648751 mins  1.71  7620
##  5 Pullman Fine Dine 24   0.50 mins 11.03 mins 2.632092 mins  1.72  7477
##  6 Pullman Fine Dine 3    0.13 mins 10.37 mins 2.645433 mins  1.73  7514
##  7 Pullman Fine Dine 4    0.50 mins 64.53 mins 2.642566 mins  1.85  7706
##  8 Pullman Fresh Market 12 0.25 mins 18.63 mins 3.477116 mins  2.42 10404
##  9 Pullman Fresh Market 16 0.50 mins 18.38 mins 3.463806 mins  2.38 10445
## 10 Pullman Fresh Market 2  0.02 mins 23.78 mins 3.489846 mins  2.42 10623
## # i 15 more rows
```

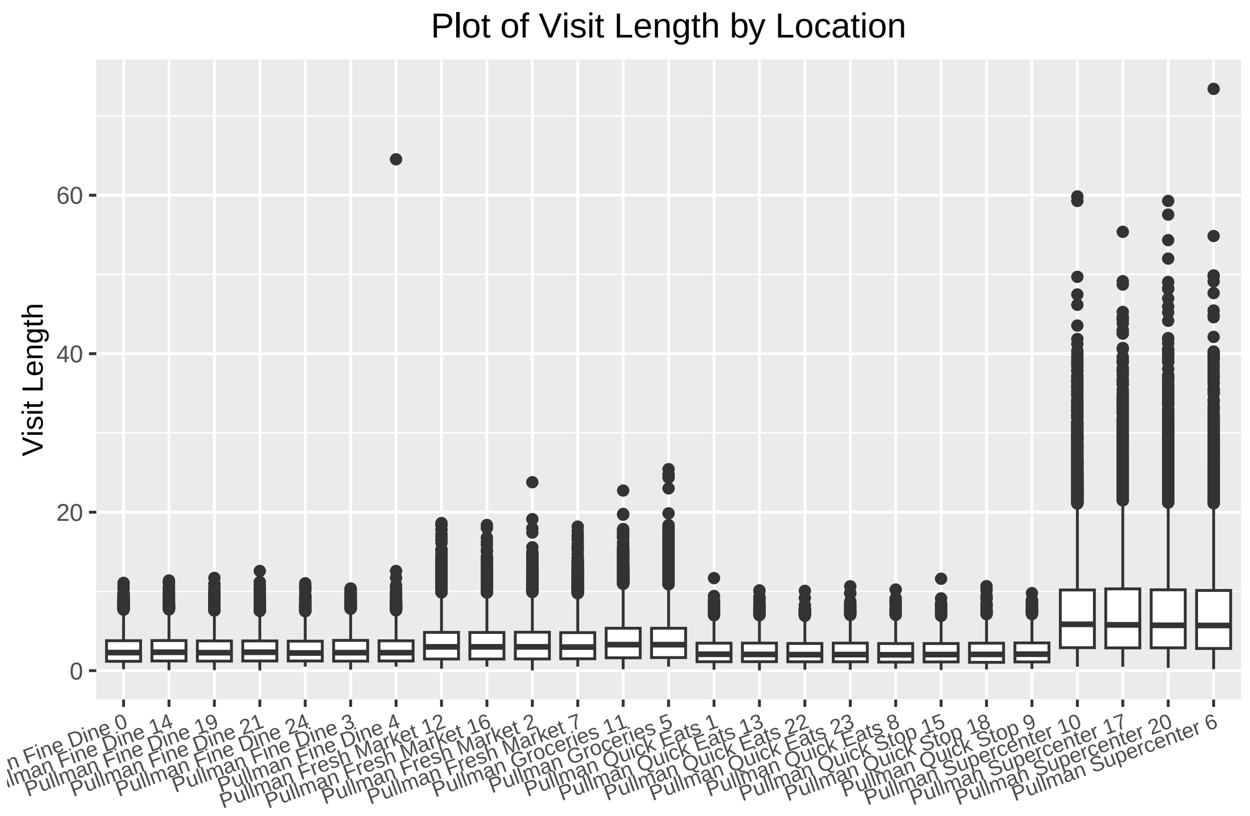
```
# Boxplot by Category
ggplot(visits, aes(x = Category, y = length)) +
  geom_boxplot() +
  labs(title = "Plot of Visit Length by Category", x="", y = "Visit Length") +
  theme(axis.text.x = element_text(angle = 20, hjust = 1, size = 8),
        plot.title = element_text(hjust = 0.5))
```

```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



```r
# Boxplot by Location
ggplot(visits, aes(x = loc, y = length)) +
  geom_boxplot() +
  labs(title = "Plot of Visit Length by Location", x="", y = "Visit Length") +
  theme(axis.text.x = element_text(angle = 20, hjust = 1, size = 8),
        plot.title = element_text(hjust = 0.5))
```

```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



```r
head(visits)
```

```
## # A tibble: 6 x 6
##   loc           start               end                 time_gap length Category
##   <chr>         <dttm>              <dttm>                 <dbl> <drtn> <chr>
## 1 Pullman Quic~ 2020-01-01 00:38:02 2020-01-01 00:38:42      0   0.67 ~ Limited~
## 2 Pullman Groc~ 2020-01-01 01:18:46 2020-01-01 01:19:26      0   0.67 ~ Superma~
## 3 Pullman Groc~ 2020-01-01 01:20:42 2020-01-01 01:22:11      0   1.48 ~ Superma~
## 4 Pullman Quic~ 2020-01-01 01:47:33 2020-01-01 01:48:39      1.1 1.10 ~ Limited~
## 5 Pullman Fine~ 2020-01-01 02:50:02 2020-01-01 02:50:43      0   0.68 ~ Full-Se~
## 6 Pullman Groc~ 2020-01-01 03:32:51 2020-01-01 03:34:31      0   1.67 ~ Superma~
```