R Notebook

Intro

This notebook plays around with some visualisation of the total weight/volume of the rubbish for the streets with bin sensors.

Load in the clean data:

```
library(tidyverse)
library(sf)
library(scales)

#read in the cleaned and combined bin & osm data
combined_bin_osm <- st_read(here::here("cleaned_data/combined_bin_osm_data/combined_bin_osm.shp"), quie

#undoing some shortened naming when read to shape file
combined_bin_osm <- combined_bin_osm %>%
  rename("street_name" = "strt_nm",
         "total_vol_l3" = "ttl_v_3",
        "total_weight_kg" = "ttl_wg_",
        "cumul_total_vol_l3" = "cm_t__3",
        "cumul_total_weight_kg" = "cm_tt__",
        "highway_group" = "hghwy_g")

# read in all the .shp files in the cleaned osm data folder
root <- here::here()
dir_path <- paste0(root, '/cleaned_data/osm_data/')
file_pattern <- '*.shp'
shp_files <- list.files(dir_path, pattern = file_pattern)


for (i in seq_along(shp_files)) {
  assign(str_remove(shp_files[i], ".shp"), st_read(paste0(dir_path, shp_files[i]),  quiet = TRUE))
}

#undoing some shortened naming when read to shape file
streets_simplified <- streets_simplified %>%
  rename("street_name" = "strt_nm",
         "highway_group" = "hghwy_g")
```

Cumulative total rubbish

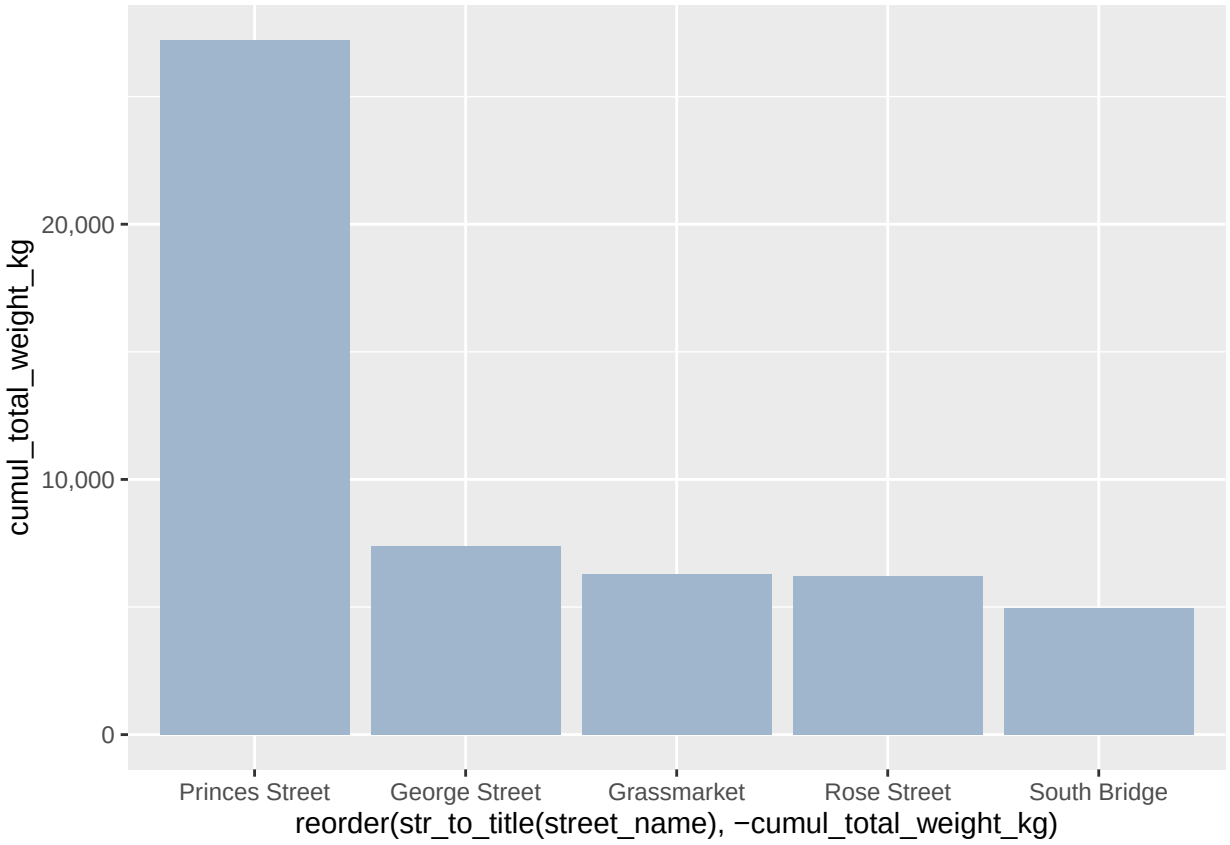
Going to look at the cumulative total rubbish. Start by subsetting the data to get the final bin collection date amount for each street:

```r
# subset the data so only the last bin collection date (so get 1 row per street)
last_date <- combined_bin_osm %>%
  st_drop_geometry() %>% #no need for geometry
  summarise(min_date = max(date))%>%
  pull()

last_date_bin_collection_sf <- combined_bin_osm %>%
  filter(date == last_date)
```

Visualising the top and bottom streets for total rubbish weight

```r
last_date_bin_collection_sf %>%
  slice_max(order_by = cumul_total_weight_kg, n = 5) %>%
  ggplot(aes(x = reorder(str_to_title(street_name), -cumul_total_weight_kg) , y = cumul_total_weight_kg
  geom_col(fill = "slategray3") +
  scale_y_continuous(labels = scales::comma)
```




```r
  theme_minimal() +
  labs(x = "Location", y = "Cumulative total rubbish weight (kg)", title = "Streets with largest cumula
```

```
## List of 95
##  $ line                        :List of 6
```

```
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ lineend      : chr "butt"
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ rect                      :List of 5
##   ..$ fill         : chr "white"
##   ..$ colour       : chr "black"
##   ..$ size         : num 0.5
##   ..$ linetype     : num 1
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_rect" "element"
##  $ text                      :List of 11
##   ..$ family       : chr ""
##   ..$ face         : chr "plain"
##   ..$ colour       : chr "black"
##   ..$ size         : num 11
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 0
##   ..$ lineheight   : num 0.9
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ title                     : chr "Streets with largest cumulative rubbish weight over 67 days"
##  $ aspect.ratio              : NULL
##  $ axis.title                : NULL
##  $ axis.title.x              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.top          :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
```

```
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.x.bottom       : NULL
##  $ axis.title.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 1
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.title.y.left         : NULL
##  $ axis.title.y.right        :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text                 :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey30"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
```

```
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.top           :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.x.bottom        : NULL
##  $ axis.text.y               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 1
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.text.y.left          : NULL
##  $ axis.text.y.right         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ axis.ticks                : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
```

```
##  $ axis.ticks.x              : NULL
##  $ axis.ticks.x.top          : NULL
##  $ axis.ticks.x.bottom       : NULL
##  $ axis.ticks.y              : NULL
##  $ axis.ticks.y.left         : NULL
##  $ axis.ticks.y.right        : NULL
##  $ axis.ticks.length         : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ axis.ticks.length.x       : NULL
##  $ axis.ticks.length.x.top   : NULL
##  $ axis.ticks.length.x.bottom: NULL
##  $ axis.ticks.length.y       : NULL
##  $ axis.ticks.length.y.left  : NULL
##  $ axis.ticks.length.y.right : NULL
##  $ axis.line                 : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.line.x               : NULL
##  $ axis.line.x.top           : NULL
##  $ axis.line.x.bottom        : NULL
##  $ axis.line.y               : NULL
##  $ axis.line.y.left          : NULL
##  $ axis.line.y.right         : NULL
##  $ legend.background         : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.margin             : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing            : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ legend.spacing.x          : NULL
##  $ legend.spacing.y          : NULL
##  $ legend.key                : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.key.size           : 'simpleUnit' num 1.2lines
##   ..- attr(*, "unit")= int 3
##  $ legend.key.height         : NULL
##  $ legend.key.width          : NULL
##  $ legend.text               :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.text.align         : NULL
##  $ legend.title              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
```

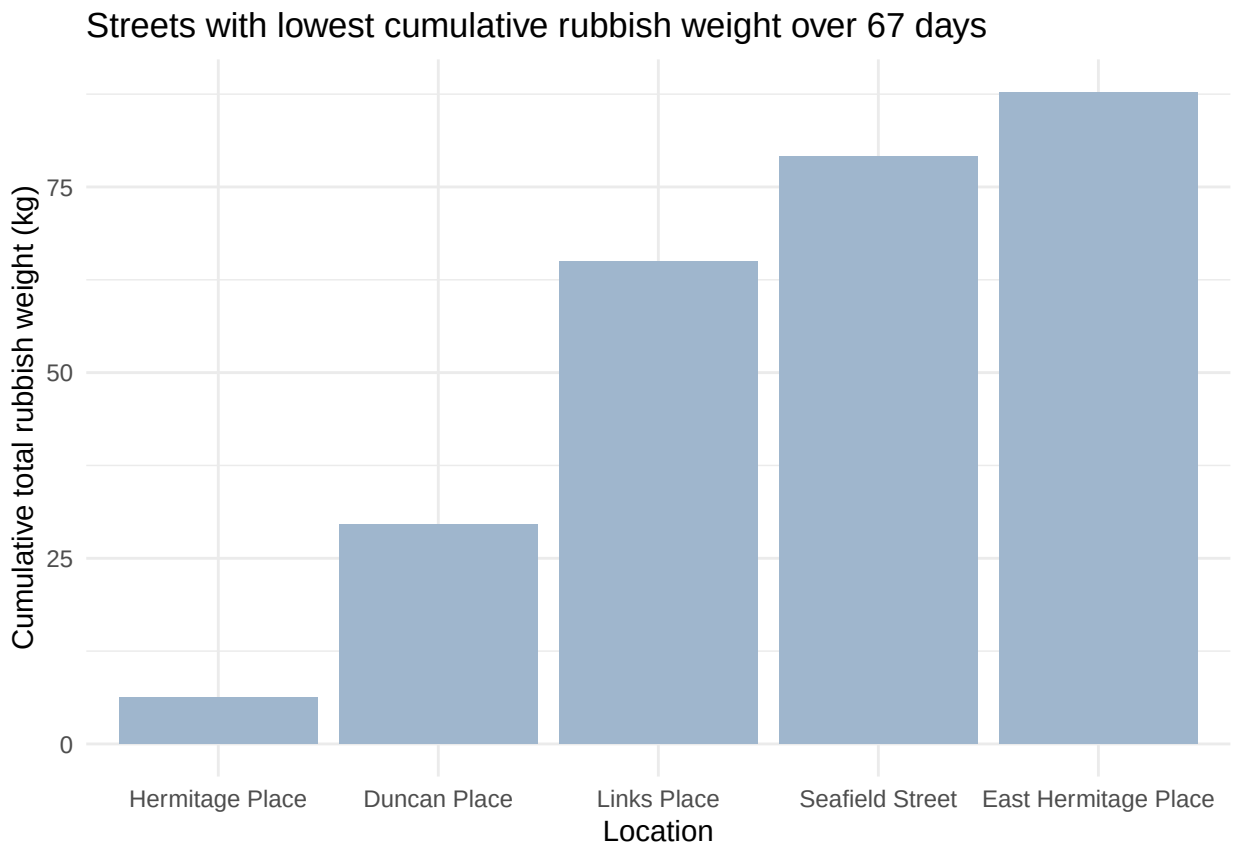
```
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ legend.title.align        : NULL
##  $ legend.position           : chr "right"
##  $ legend.direction          : NULL
##  $ legend.justification      : chr "center"
##  $ legend.box                : NULL
##  $ legend.box.just           : NULL
##  $ legend.box.margin         : 'margin' num [1:4] 0cm 0cm 0cm 0cm
##   ..- attr(*, "unit")= int 1
##  $ legend.box.background     : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.box.spacing        : 'simpleUnit' num 11points
##   ..- attr(*, "unit")= int 8
##  $ panel.background          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.border              : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ panel.spacing             : 'simpleUnit' num 5.5points
##   ..- attr(*, "unit")= int 8
##  $ panel.spacing.x           : NULL
##  $ panel.spacing.y           : NULL
##  $ panel.grid                :List of 6
##   ..$ colour       : chr "grey92"
##   ..$ size         : NULL
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major          : NULL
##  $ panel.grid.minor          :List of 6
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.5
##   ..$ linetype     : NULL
##   ..$ lineend      : NULL
##   ..$ arrow        : logi FALSE
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_line" "element"
##  $ panel.grid.major.x        : NULL
##  $ panel.grid.major.y        : NULL
##  $ panel.grid.minor.x        : NULL
##  $ panel.grid.minor.y        : NULL
##  $ panel.ontop               : logi FALSE
##  $ plot.background           : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ plot.title                :List of 11
```

```
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.title.position       : chr "panel"
##  $ plot.subtitle             :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 0points 0points 5.5points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : num 1
##   ..$ vjust        : num 1
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 5.5points 0points 0points 0points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.caption.position     : chr "panel"
##  $ plot.tag                  :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : 'rel' num 1.2
##   ..$ hjust        : num 0.5
##   ..$ vjust        : num 0.5
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
```

```
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ plot.tag.position         : chr "topleft"
##  $ plot.margin               : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
##   ..- attr(*, "unit")= int 8
##  $ strip.background          : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ strip.background.x        : NULL
##  $ strip.background.y        : NULL
##  $ strip.placement           : chr "inside"
##  $ strip.text                :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : chr "grey10"
##   ..$ size         : 'rel' num 0.8
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : 'margin' num [1:4] 4.4points 4.4points 4.4points 4.4points
##   .. ..- attr(*, "unit")= int 8
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.text.x              : NULL
##  $ strip.text.y              :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num -90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ strip.switch.pad.grid     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.switch.pad.wrap     : 'simpleUnit' num 2.75points
##   ..- attr(*, "unit")= int 8
##  $ strip.text.y.left         :List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : NULL
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi TRUE
```

```
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  $ x                           : chr "Location"
##  $ y                           : chr "Cumulative total rubbish weight (kg)"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi TRUE
##  - attr(*, "validate")= logi TRUE
```

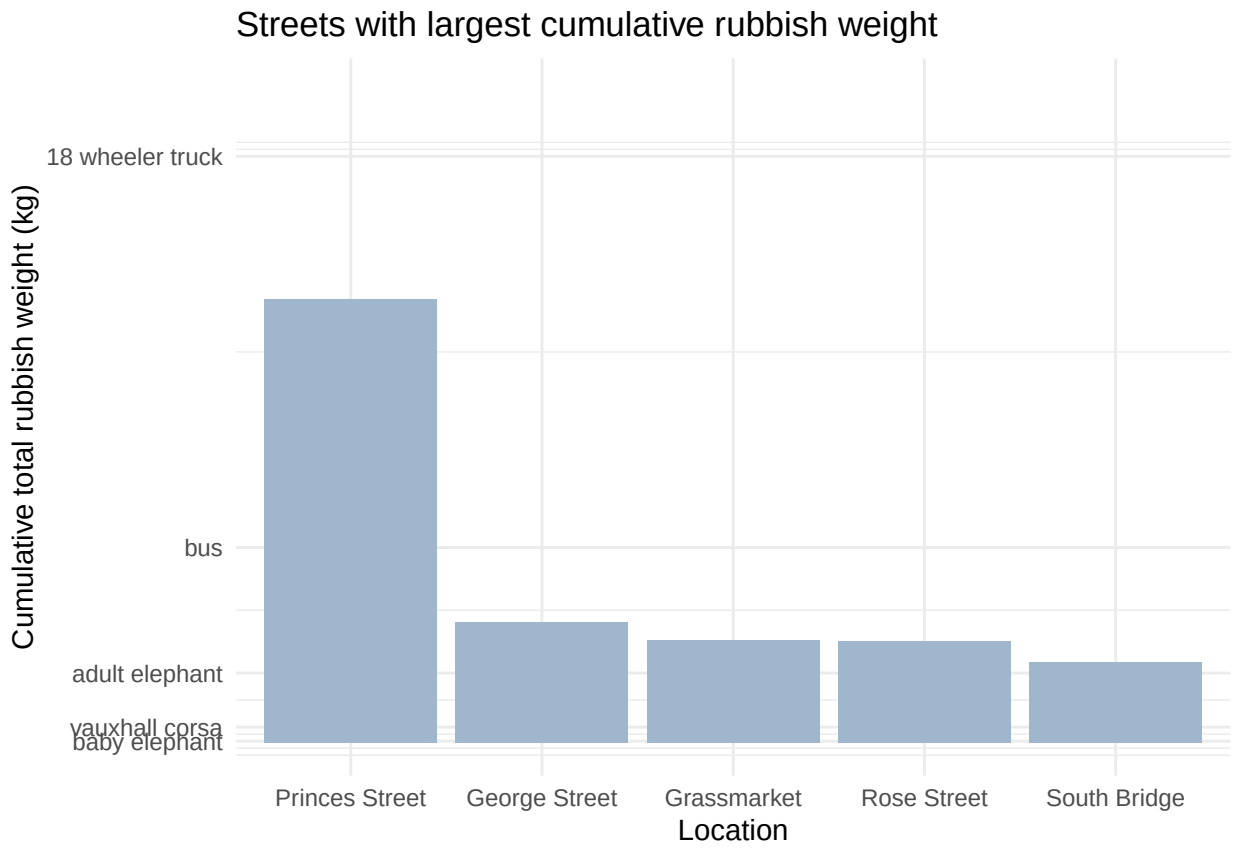
```
last_date_bin_collection_sf %>%
  slice_min(order_by = cumul_total_weight_kg, n = 5) %>%
  ggplot(aes(x = reorder(str_to_title(street_name), cumul_total_weight_kg) , y = cumul_total_weight_kg)
  geom_col(fill = "slategray3") +
  theme_minimal() +
  #ylim(c(0, max(last_date_bin_collection_sf$cumul_total_weight_kg))) +
  labs(x = "Location", y = "Cumulative total rubbish weight (kg)", title = "Streets with lowest cumulati
```



It can be difficult to think about the weights i.e. how heavy is 20,000 kg, so have also produced these plots with some contextual weights:

Object	Average weight (kg)
Baby elephant	120
Vauxhall Corsa	980
Adult elephant (	4,300
UK bus	12,000
18 wheeler truck	36,000

```r
last_date_bin_collection_sf %>%
  slice_max(order_by = cumul_total_weight_kg, n = 5) %>%
  ggplot(aes(x = reorder(str_to_title(street_name), -cumul_total_weight_kg) , y = cumul_total_weight_kg
  geom_col(fill = "slategray3") +
  scale_y_continuous(breaks = c(120, 980, 4300,12000,  36000) , labels = c('baby elephant', 'vauxhall c
  theme_minimal() +
  labs(x = "Location", y = "Cumulative total rubbish weight (kg)", title = "Streets with largest cumula
```



Visualsing spatially the total rubbish weight per street

Setting up the 'base' spatial plot of Edinburgh city centre using the OSM data. The line thickness denotes the type of street/road. The wider the thickness the more 'major' the street/road is (either small/medium/large).

```r
# Plots ---------------------------------------

# setting the max and min lat/long for cropping visual
min_max_coords <- c(ymin= 55.928479867725436,
                    xmin= -3.227624857214751,
                    ymax= 55.98333902701634,
                    xmax= -3.1407356284353 86)


map_theme <-theme(axis.line=element_blank(),
                  axis.text.x=element_blank(),
```

```
                axis.text.y=element_blank(),
                axis.ticks=element_blank(),
                axis.title.x=element_blank(),
                axis.title.y=element_blank(),
                plot.background=element_blank(),
                panel.grid.minor=element_blank(),
                panel.background=element_blank(),
                panel.grid.major=element_blank())

base_plot <- ggplot() +
  map_theme +
  geom_sf(data = water,
          fill = "steelblue",
          # size = .8,
          lwd = 0,
          alpha = .3) +
  geom_sf(data = park_multipoly,
          fill = "green",
          # size = .8,
          lwd = 0,
          alpha = .3) +
  geom_sf(data = park_poly,
          fill = "green",
          # size = .8,
          lwd = 0,
          alpha = .3) +
  geom_sf(data = railways,
          color = "grey30",
          size = .2,
          linetype="dotdash",
          alpha = .5) +
  geom_sf(data = filter(streets_simplified, highway_group == "small"),
          size = .1,
          color = "grey40") +
  geom_sf(data = filter(streets_simplified, highway_group == "medium"),
          size = .3,
          color = "grey35") +
  geom_sf(data = filter(streets_simplified, highway_group == "large"),
          size = .5,
          color = "grey30") +
    coord_sf(ylim = c(min_max_coords[1], min_max_coords[3]),
           xlim = c(min_max_coords[2], min_max_coords[4]),
           expand = FALSE)

base_plot +
   labs(caption = 'Edinburgh - base plot using OpenStreetMap data', size = 2)
```

Edinburgh − base plot using OpenStreetMap data

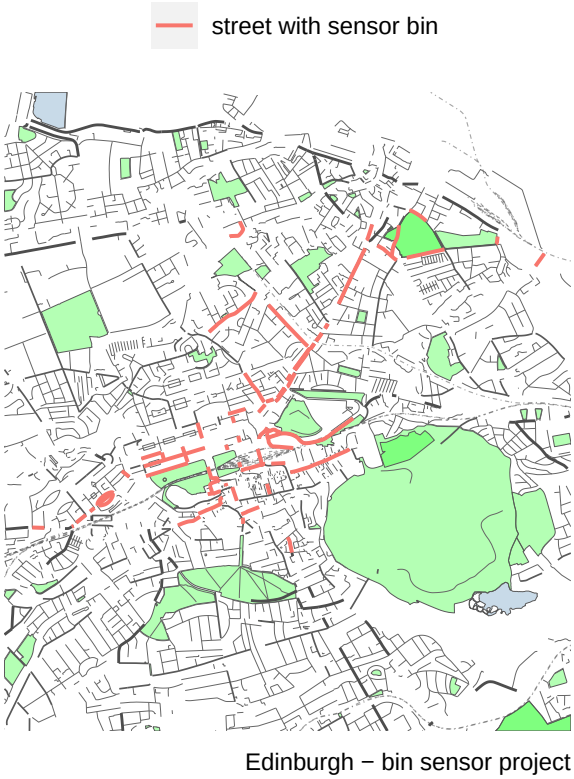
Now playing around with different ways to visualise the data:

Highlighting which streets have sensor bins on:

```
# Highlighting which streets have sensor bins on ----------------------------
# to have in README

base_plot +
  geom_sf(data =  last_date_bin_collection_sf, aes(color = "street with sensor bin"), size = .7, show.l
  labs(caption = 'Edinburgh - bin sensor project', title= "Plot of which streets have bin sensors", size
  coord_sf(ylim = c(min_max_coords[1], min_max_coords[3]),
           xlim = c(min_max_coords[2], min_max_coords[4]),
           expand = FALSE) +
  theme(legend.position = "top", legend.title = element_blank())
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```



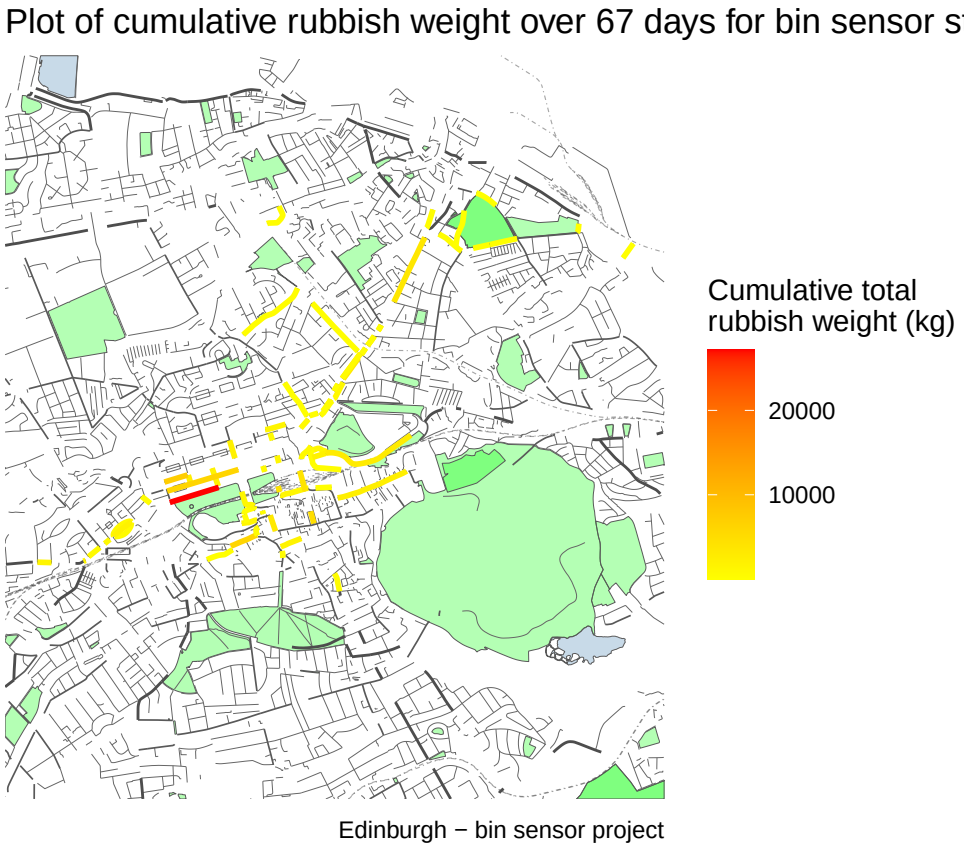

```
ggsave("plot_images/streets_highlighted.png")
```

```
## Saving 6.5 x 4.5 in image
```

A lot of sequential colour palettes begin at very light colours which would make it hard to see some of the streets with lower levels of rubbish, tested out a few colour schemes:

```
# Last bin collection colour showing weight ----------------------------
base_plot +
  geom_sf(data =  last_date_bin_collection_sf, aes(colour = cumul_total_weight_kg),
          size = 1,
          show.legend = "line") +
  scale_colour_gradient(low = "yellow", high = "red", name = "Cumulative total \nrubbish weight (kg)") +
  labs(caption = 'Edinburgh - bin sensor project', title = "Plot of cumulative rubbish weight over 67 da
  coord_sf(ylim = c(min_max_coords[1], min_max_coords[3]),
           xlim = c(min_max_coords[2], min_max_coords[4]),
           expand = FALSE)
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```




```
ggsave("plot_images/streets_by_weight_red_yellow.png")
```

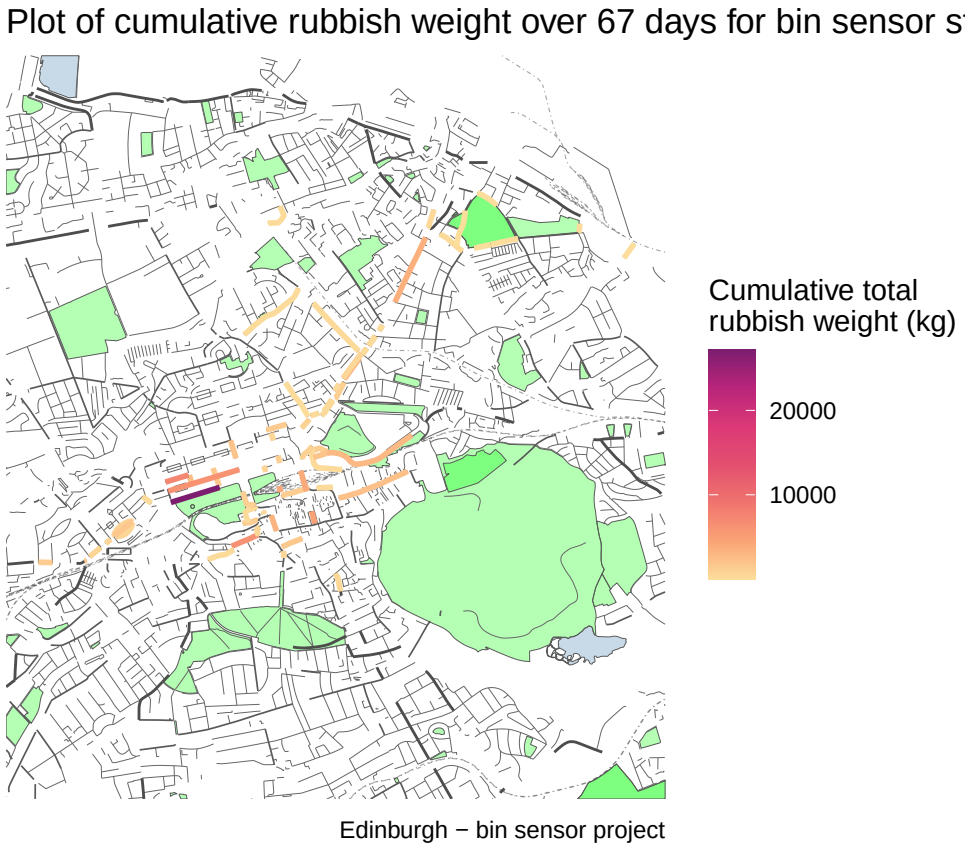
```
## Saving 6.5 x 4.5 in image
```

From this blog (here)[https://blog.datawrapper.de/which-color-scale-to-use-in-data-vis/] by Lisa Charlotte Rost she said on sequential colour schemes *'Using two or even more hues increases the color contrast between segments of your gradient, making it easier for readers to distinguish between them'* so came across some on CARTO so gave one a go:

```
library(rcartocolor)

base_plot +
  geom_sf(data =  last_date_bin_collection_sf, aes(colour = cumul_total_weight_kg),
          size = 1,
          show.legend = "line") +
  scale_color_carto_c(palette = "SunsetDark", name = "Cumulative total \nrubbish weight (kg)") +
  #name = "Cumulative total \nrubbish weight (kg)") +
  labs(caption = 'Edinburgh - bin sensor project', title = "Plot of cumulative rubbish weight over 67 da
  coord_sf(ylim = c(min_max_coords[1], min_max_coords[3]),
           xlim = c(min_max_coords[2], min_max_coords[4]),
           expand = FALSE)
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```



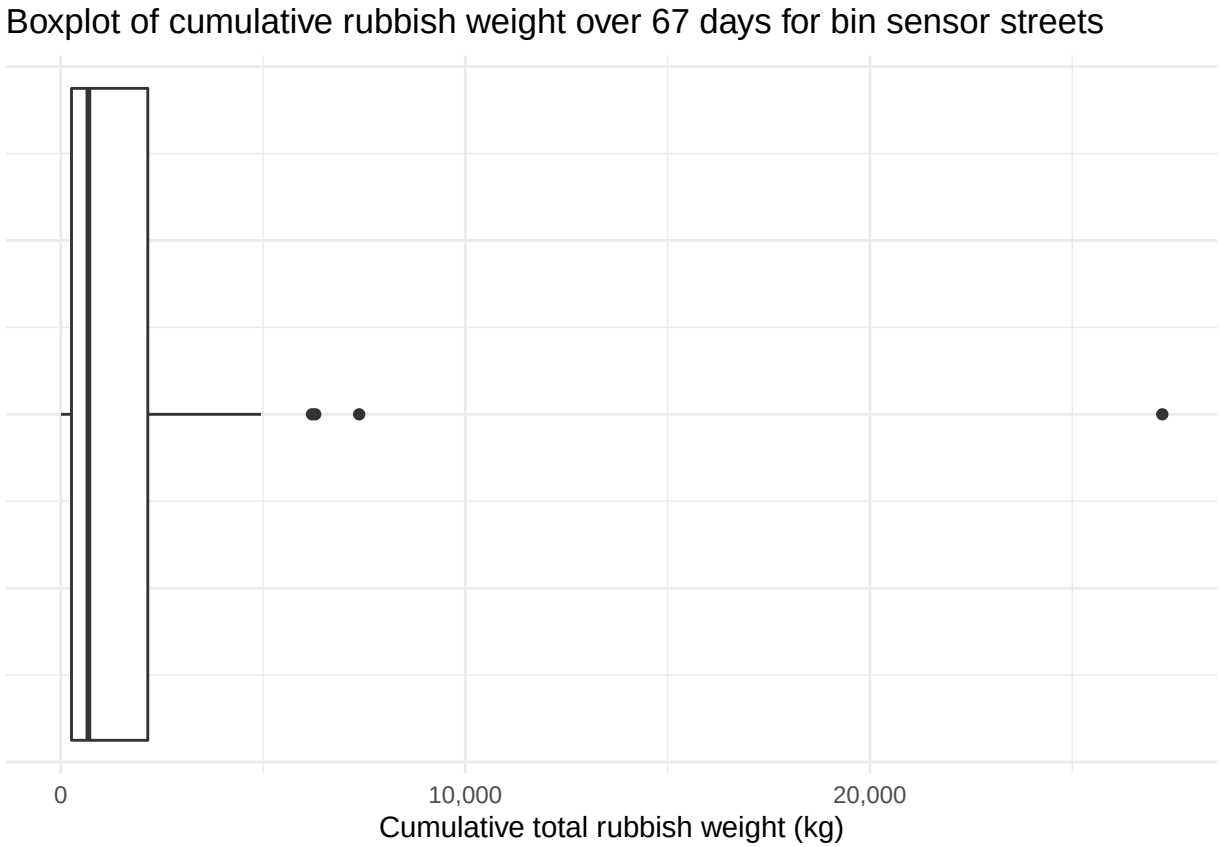

```
ggsave("plot_images/streets_by_weight_sunset.png")
```

```
## Saving 6.5 x 4.5 in image
```

In future want to give this more of an investigation and look into some more palettes (or potentially using a different coloured background).

Princes Street is a bit of an outlier here so hard to see the differences between the other streets.

```
last_date_bin_collection_sf %>%
  ggplot(aes(cumul_total_weight_kg)) +
  geom_boxplot() +
  xlab("Cumulative total rubbish weight (kg)") +
  ggtitle("Boxplot of cumulative rubbish weight over 67 days for bin sensor streets") +
  theme_minimal() +
  scale_x_continuous(labels = scales::comma) +
  theme(axis.text.y=element_blank(), axis.ticks.y=element_blank())
```

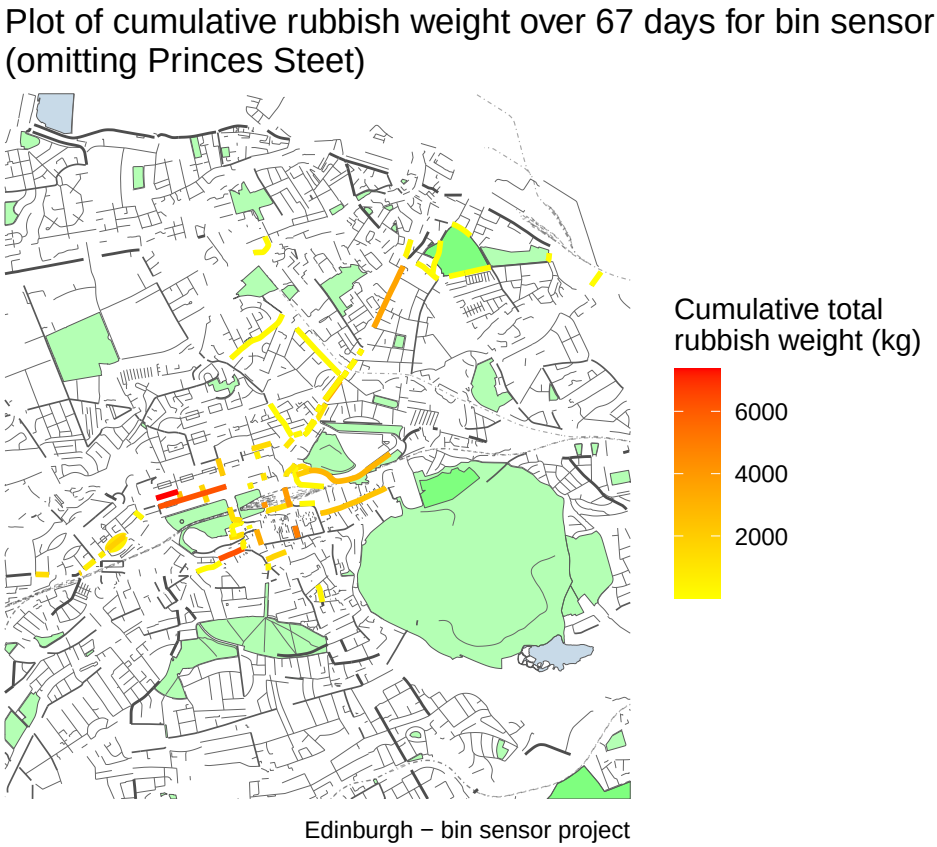


Yes Princes Street is a big outlier. Makes sense as it is the main shopping street in Edinburgh and lots of footfall. In future look to see if any open footfall data to add to analysis.

Could do a transformation of the variable to show the differences between the other streets more, or show some plot without Princes Street (which have below).

```
# Last bin collection colour showing weight removing Princes Street as is  ----------------------------
base_plot +
  geom_sf(data =  filter(last_date_bin_collection_sf, street_name != "princes street"), aes(colour = cu
          size = 1,
          show.legend = "line") +
  scale_colour_gradient(low = "yellow", high = "red", name = "Cumulative total \nrubbish weight (kg)") +
  labs(caption = 'Edinburgh - bin sensor project', title = "Plot of cumulative rubbish weight over 67 da
  coord_sf(ylim = c(min_max_coords[1], min_max_coords[3]),
           xlim = c(min_max_coords[2], min_max_coords[4]),
           expand = FALSE)
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```



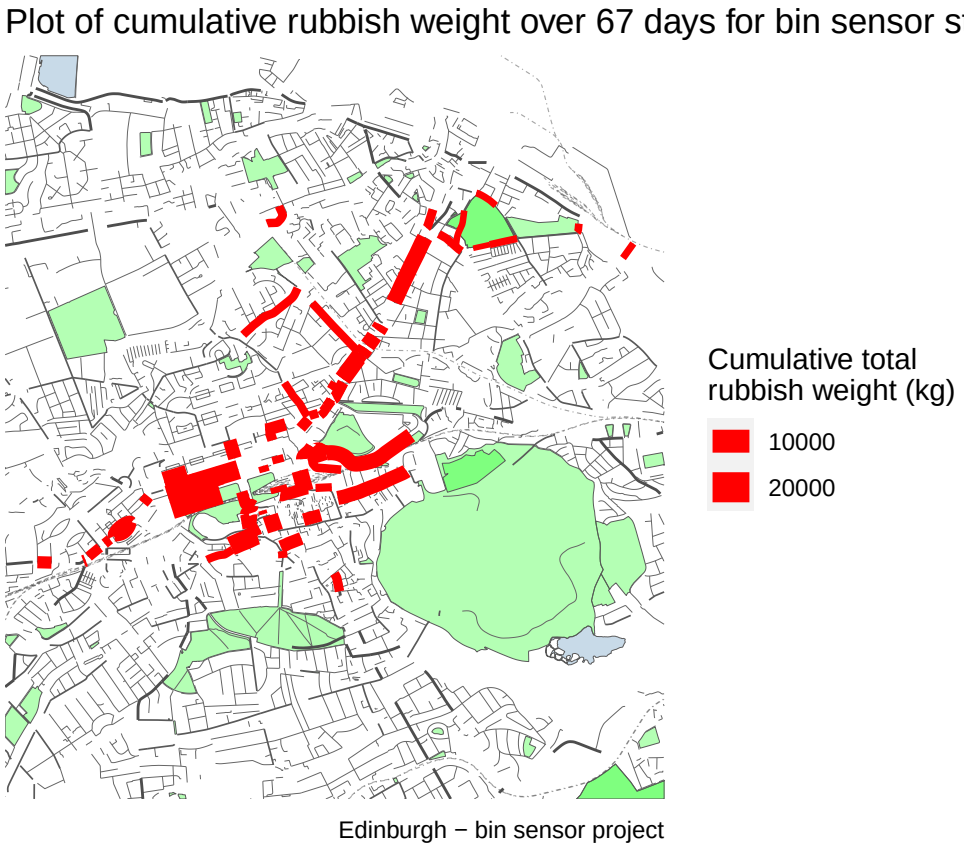

```
ggsave("plot_images/streets_by_weight_red_yellow_nops.png")
```

```
## Saving 6.5 x 4.5 in image
```

Play around with using the using the weight of the street to denote the weight:

```
# Last bin collection thickness showing weight ----------------------------
# decided against this way to visualise as not very clear
base_plot +
  geom_sf(data =  last_date_bin_collection_sf, aes(size = cumul_total_weight_kg),
          show.legend = "line", colour = "red") +
  labs(caption = 'Edinburgh - bin sensor project', size = "Cumulative total \nrubbish weight (kg)", titl
  coord_sf(ylim = c(min_max_coords[1], min_max_coords[3]),
           xlim = c(min_max_coords[2], min_max_coords[4]),
           expand = FALSE)
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```

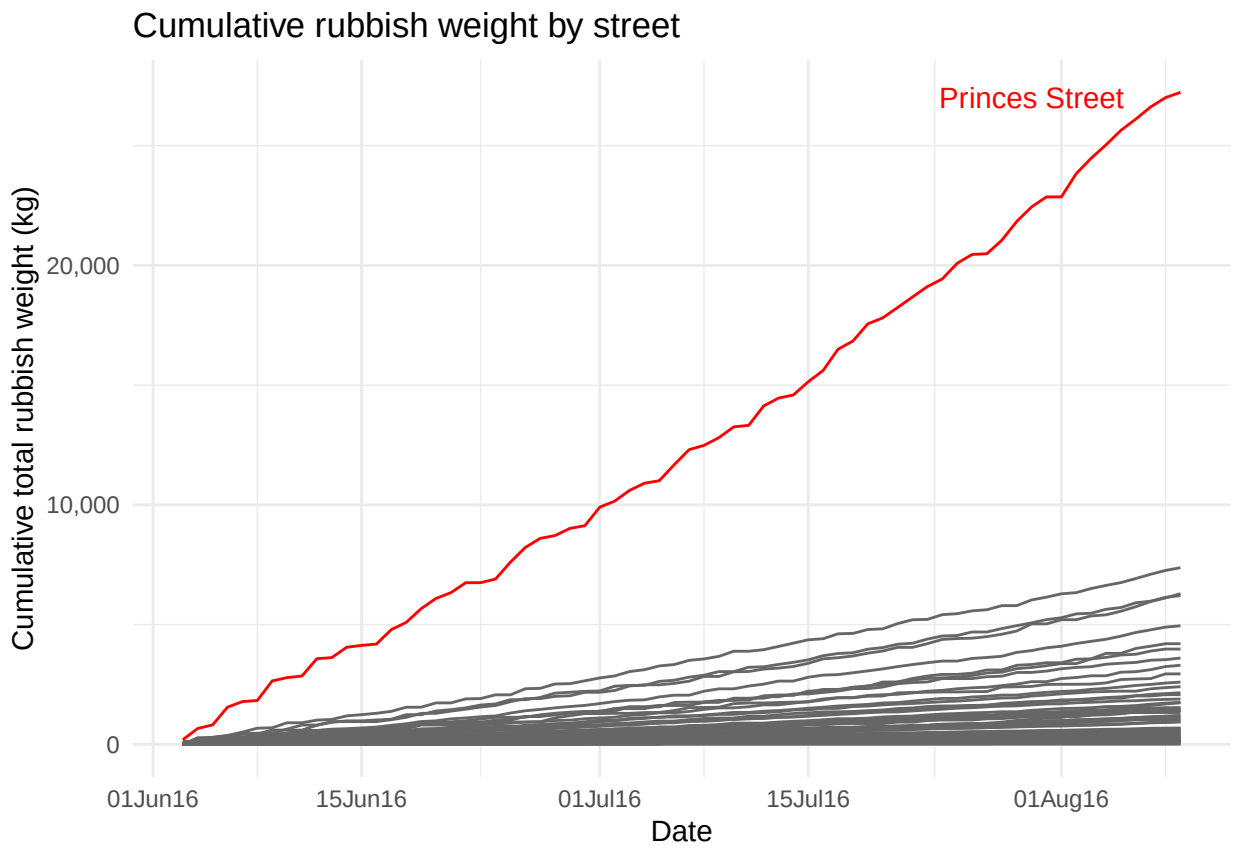


Don't think this is a good way of visualising this data!

Visalising the cumulative rubbish on each street over time

Now going to look at how the weight changes over time:

```
# Streets over time ----------------------------
combined_bin_osm %>%
  ggplot(aes(x = date, y = cumul_total_weight_kg, group = street_name, colour = street_name == "princes
  geom_line() +
  scale_colour_manual(values = c("grey40", "red"), guide = "none") +
  annotate("text", x=as.Date("2016-07-30"), y=27000, label="Princes Street", color = "red") +
  theme_minimal() +
  scale_x_date(date_labels = "%d%b%y") +
  labs(x = "Date", y = "Cumulative total rubbish weight (kg)", title = "Cumulative rubbish weight by st
  scale_y_continuous(labels = scales::comma)
```



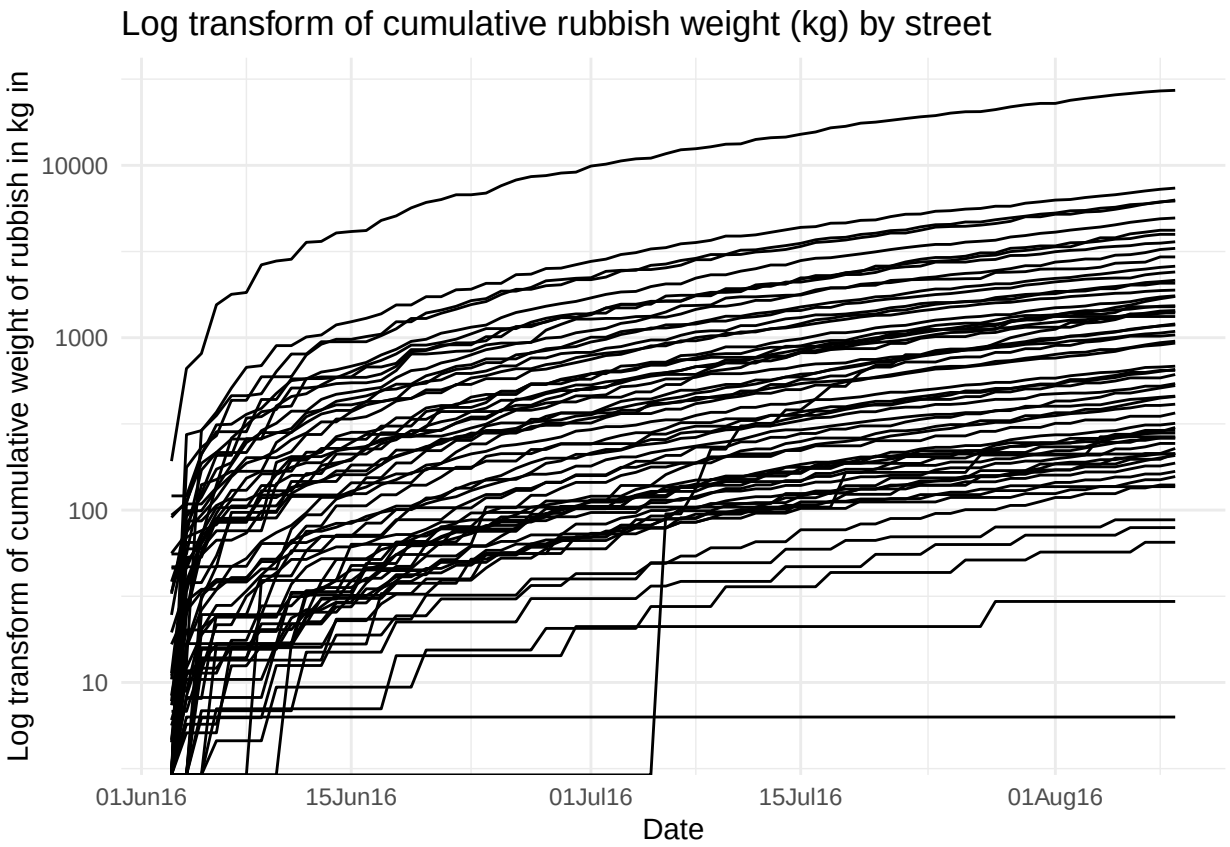

```
ggsave("plot_images/weight_over_time.png")
```

```
## Saving 6.5 x 4.5 in image
```

Again, Princes Street dominates the chart here but it has a steady rate of change. Going to look more at the rate of change for each of the streets by visualising the log transformations.

```
#transform to a log axis on y to compare rate of change
combined_bin_osm %>%
  ggplot(aes(x = date, y = cumul_total_weight_kg, group = street_name)) +
  geom_line() +
  scale_y_continuous(trans = "log10") +
  scale_x_date(date_labels = "%d%b%y") +
   theme_minimal() +
  labs(y = "Log transform of cumulative weight of rubbish in kg in" , x = "Date", title = "Log transfor
```

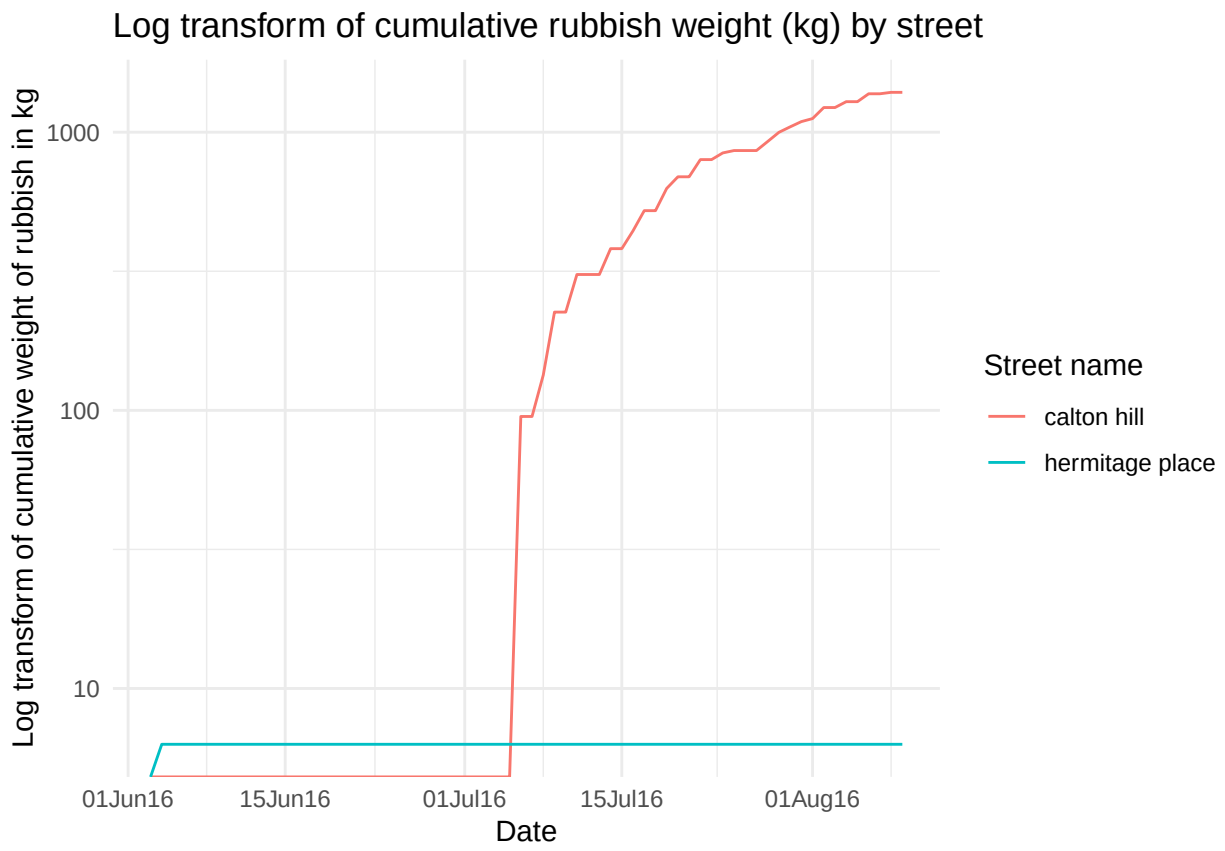
```
## Warning: Transformation introduced infinite values in continuous y-axis
```



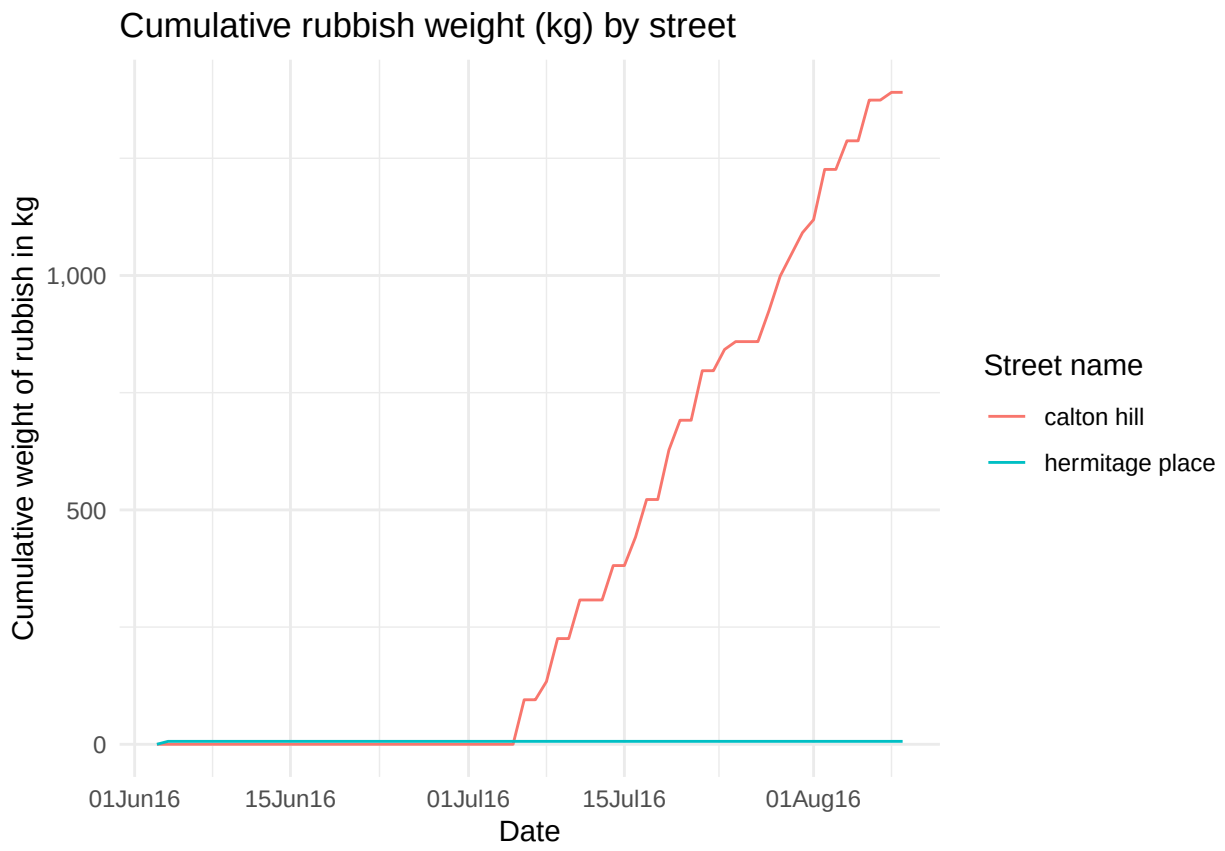
Seems to be a common pattern for most of the streets in the rate of change. Pinpointed 2 locations, Calton Hill and Hermitage Place that seem to have a bit of a different pattern to look at in isolation:

```
combined_bin_osm %>%
  filter(street_name %in% c("calton hill", "hermitage place")) %>%
  ggplot(aes(x = date, y = cumul_total_weight_kg, color = street_name)) +
  geom_line() +
  scale_y_continuous(trans ="log10") +
  scale_x_date(date_labels = "%d%b%y") +
  labs(y = "Log transform of cumulative weight of rubbish in kg", x = "Date", title = "Log transform of
  theme_minimal()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```




```
combined_bin_osm %>%
  filter(street_name %in% c("calton hill", "hermitage place")) %>%
  ggplot(aes(x = date, y = cumul_total_weight_kg, color = street_name)) +
  scale_x_date(date_labels = "%d%b%y") +
  labs(y = "Cumulative weight of rubbish in kg", x = "Date", title = "Cumulative rubbish weight (kg) by
  geom_line() +
  theme_minimal() +
  scale_y_continuous(labels = scales::comma)
```




```
ggsave("plot_images/weight_over_time_subset.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
combined_bin_osm %>%
  filter(street_name == "hermitage place") %>%
  st_drop_geometry() %>%
  select(street_name, date, total_weight_kg, cumul_total_weight_kg) %>%
  head(5)
```

```
##       street_name       date total_weight_kg cumul_total_weight_kg
## 1 hermitage place 2016-06-03            0.0                   0.0
## 2 hermitage place 2016-06-04            6.3                   6.3
## 3 hermitage place 2016-06-05            0.0                   6.3
## 4 hermitage place 2016-06-06            0.0                   6.3
## 5 hermitage place 2016-06-07            0.0                   6.3
```

For Hermitage Place there was only a small bit of rubbish on the 2nd day of having the sensor (it is a little out of city centre) or the sensor perhaps didn't work after this day. Similarly for Calton Hill perhaps there was an issue with the sensor as after the start of July the rate of increase increases sharply.