R web scraping analysis: Bicycle theft in the United Kingdom

From 1st November 2019 to 31 st October 2022

Author: Samuel Lam

Date: 10th November 2022

Table of Contents

1. Introduction

1.1 Background

Bicycle is now one of the most important transportation tools in the world in terms of its cost-saving, environmental-friendly nature, as well as its advantages on health. The benefits are exceptionally considerable when it comes to the problem of energy shortage.

During the nearly 5 years of cycling to school when I was young, I was fortunate enough the bicycle never got stolen. While there were a few times it's obvious someone attempted to break the chain lock but failed. I can definitely understand how frustrated the owners would be if their bicycles got stolen. It is understandable that many bicycles may not value much and police may not put in too much resources to investigate each case (But make no mistake, some bicycles with expensive gears could cost over 10,000 pounds).

With the powerful internet resources nowadays, owners of bicycles can now report their lost on different websites with an extra hope for recovery. One of these in the UK is the Stolen Bikes UK. Thanks to the hard

work of the founder John Moss, the website now contains more than 50,000 reported stolen bicycles across the UK since year 2012. It is encouraging when some of the reported stolen bicycles had been recovered. The website also provides data on some bicycle thefts in which the owners did not report to the police.

As one may notice that there is already official statistics provided by the Office for National Statistics (ONS, the UK recognized national statistical institute), which is available here. Currently the latest available dataset is the "Year ending March 2020 edition of this dataset", which means that there is a time gap between the official statistics and the recent situation.

The aim of this analysis is to provide an overview across the past three years from 1st November 2019 to 31st October 2022, and to reveal any possible trend or pattern of the thefts and other insightful information.

All the data preparation, processing, analysis and visualization will be done in R Programming with R Studio.

1.2 Terminology and definitions

- Nomenclature of Territorial Units for Statistics (NUTS) areas: The 12 areas of NUTS level 1 (hereinafter referred to as "regions") in this document follows the standard in the Office for National Statistics.
- Seasons definition is used according to the meteorological calendar from the Met Office, which is defined as: spring (March, April, May), summer (June, July, August), autumn (September, October, November), and winter (December, January, February).

2. Data Preparation

2.1 Data Source

- Stolen Bikes UK, https://stolen-bikes.co.uk
- The region (except Scotland) populations were obtained from the Census 2021 result from the Office for National Statistics and the Northern Ireland Statistics and Research Agency: https://www.ons.gov.uk/census
- Scotland population estimation was obtained from the Office for National Statistics, as result from Scotland's Census 2022 is not yet available. Please click here for information.

2.2 Method of data collection

- R programming is used for convenient and efficient data manipulation. High quality visualization could also be generated at the same time.
- Web Scraping was used in this webpage. There are two levels of data. The first level contains a thumbnail list of 12 bicycle thefts per page, with some basic information of each case. The second level is accessed when clicking into each case, with more detailed information provided.
- To avoid unnecessary traffic on the website, it is observed that for thefts occurring after 1st November 2019, it sits at page 1690 (with the most recent case in page 1 and in total 4300 pages).

2.3 Data limitations

- Sample size: since there are also other similar websites for reporting, the dataset cannot represent a full picture of all internet-reported bicycle thefts.
- Data format: some of the data were input by bicycle owners' typing but not from selection fields, which resulted in inconsistent data format. For instance, inconsistent date format, and date with/without time or time only. Data processing could not fully retrieve all the data in a useful way for analysis.

2.4 Data collection

2.4.1 Libraries loading

```
library(rvest)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()         masks stats::filter()
## x readr::guess_encoding() masks rvest::guess_encoding()
## x dplyr::lag()            masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(stringr)
library(readr)
```

2.4.2 Web Scraping

```
url = "https://stolen-bikes.co.uk/stolen-bikes/page/%d/"

#For 3 years data, it ends at page 1690
stolen <- map_df(1:1690, function(i) {
  page <- read_html(sprintf(url, i))

#Create temporary variable for the next step
  temp_bike <- page %>%
    html_nodes("h3") %>% html_text()

  temp_location <- page %>%
    html_nodes(".listmeta") %>% html_nodes("a") %>% html_text()

  temp_link <- page %>%
    html_nodes("h3") %>% html_nodes("a") %>% html_attr("href")

#Extract details from the 2nd level of the website
  temp_details <- as.character(lapply(temp_link, function(x)
    x %>% read_html() %>% html_nodes(".bikemeta") %>% html_text()))

  temp_bike_des <- as.character(lapply(temp_link, function(x)
    x %>% read_html() %>% html_nodes(".bikedescription") %>% html_text()))
```

```
  temp_theft_des <- as.character(lapply(temp_link, function(x)
    x %>% read_html() %>% html_nodes(".theftdescription") %>% html_text()))

  tibble(
    bike_name = temp_bike[1:(length(temp_bike) - 3)],
    location1 = temp_location[seq(1, length(temp_location), 2)],
    location2 = temp_location[seq(2, length(temp_location), 2)],
    link = temp_link,
    details = temp_details,
    bike_des = temp_bike_des,
    theft_des = temp_theft_des
  )
})

#Save and keep the original dataset
write_csv(stolen, "bike_stolen.csv")
```

- It took a few hours to finish the web scraping as there were 1,690 level 1 pages, and 1,690 * 12 = 20,280 level 2 pages.

3. Data Processing (Cleaning and transformation)

3.1 Checking of duplicate rows

- Check for any duplicate rows due to the long time web scraping and possibly update of the web pages.
- Use a new variable "stolen2" for data cleaning, transformation and analysis, hold the original variable "stolen" for comparison in case it is needed.

```
stolen2 <- read_csv("bike_stolen.csv")
```

```
## Rows: 20280 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): bike_name, location1, location2, link, details, bike_des, theft_des
##
## 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.
```

```
stolen2[duplicated(stolen2), ]
```

```
## # A tibble: 14 x 7
##    bike_name                       locat~1 locat~2 link  details bike_~3 theft~4
##    <chr>                           <chr>   <chr>   <chr> <chr>   <chr>   <chr>
##  1 Intersens  (custom made/assemb~ London  London  http~ "Colou~ "\n   ~ "\n   ~
##  2 Ribble R872 Disc                London  London  http~ "Colou~ "\n   ~ "chara~
##  3 Voodoo Bucca                    Somers~ South ~ http~ "Colou~ "\n   ~ "\n   ~
##  4 Moustache XROAD FS 7 VERT       London  London  http~ "Colou~ "\n   ~ "\n   ~
##  5 BMC SLR02                       London  London  http~ "Colou~ "\n   ~ "\n   ~
##  6 Trek emonda sl6                 London  London  http~ "Colou~ "\n   ~ "\n   ~
##  7 Carrera bicycles Crossfire 2 21 London  London  http~ "Colou~ "\n   ~ "\n   ~
##  8 Btwin Tilt 120                  London  London  http~ "Colou~ "\n   ~ "\n   ~
##  9 Whyte Hardtail MTB 805          Edinbu~ Scotla~ http~ "Colou~ "\n   ~ "\n   ~
## 10 Carrera bicycles Vengeance E M~ Dorset  South ~ http~ "Colou~ "\n   ~ "\n   ~
## 11 Trek Domane SL5 2020            Bristol South ~ http~ "Colou~ "\n   ~ "\n   ~
## 12 OFO just one style              East S~ South ~ http~ "Colou~ "\n   ~ "\n   ~
```

```
## 13 Boardman Bikes Junior Hybrid S~ London  London  http~ "Colou~ "\n   ~ "\n   ~
## 14 Claud Butler levante            Oxford~ South ~ http~ "Colou~ "\n   ~ "\n   ~
## # ... with abbreviated variable names 1: location1, 2: location2, 3: bike_des,
## #   4: theft_des
```

```
stolen2 <- stolen2[!duplicated(stolen2), ]
```

3.2 Character encoding

- Convert all characters to "UTF-8" encoding.

```
for (i in 1:7) {
  stolen2[, i] <-
    sapply(stolen2[, i], iconv, from = "UTF-8", to = "UTF-8", sub = " ")
}
```

3.3 Data columns split

- As observed from the head and tail rows, the column "details" contains 6 data elements separated by "\n", which are:
  - Colour
  - Frame number
  - Stolen date
  - Stolen location
  - Crime reference number
  - Reward if it is recovered
- Split them into different columns.
- There is an extra column created after the last "\n", thus the "extra" column is also created for that.

```
head(stolen2$details)
```

```
## [1] "ColourBlack/Lithium Grey \r\n                              Frame Number1046368 \r\n
## [2] "ColourBlack, White and Yellow \r\n                             Frame NumberGW9E3731 \r\n
## [3] "ColourGrey / Silver \r\n                             Frame NumberUnknown \r\n
## [4] "ColourGold \r\n                                Frame NumberEN17467 \r\n
## [5] "ColourBlack/White with Teal Trim \r\n                              Frame NumberBY2B8FU \r\n
## [6] "ColourRed & Grey \r\n                                  Frame NumberAL19050270 \r\n
```

```
tail(stolen2$details)
```

```
## [1] "Colouroak green \n                               Frame NumberWSBC604448000N \n
## [2] "ColourBlue and green \n                                Frame NumberUnknown \n
## [3] "ColourSilver \n                                Frame NumberAA605 13994 \n
## [4] "ColourOrange \n                                Frame NumberWSBC602055442K \n
## [5] "ColourGrey / Green \n                                Frame NumberHidden \n
## [6] "ColourRed \n                                 Frame NumberEM2NS17014370560017 \n
```

```
stolen2 <-
  separate(stolen2, col = details, into = c("colour", "frame_number", "location",
    "date_time", "crime_ref_num", "reward_GBP", "extra"), sep = "\\n")
```

3.4 Unnecessary columns removal

- Content of the column "location" is the same as "location1" and "location2", thus it is removed.

```
stolen2 <- subset(stolen2, select = -location)
```

- Check if the "extra" column is empty//NULL, delete the column if TRUE.

```
all(is.na(stolen2$extra) |
      stolen2$extra == "" | is.null(stolen2$extra) | is.na(stolen2$extra))
```

```
## [1] TRUE
```

```
stolen2 <- subset(stolen2, select = -extra)
```

3.5 Data cleaning for remaining columns

3.5.1 Unnecessary characters removal

- Remove the column names that appear in the columns' values.

```
stolen2$colour <- sub("Colour", "", stolen2$colour)
stolen2$frame_number <- sub("Frame Number", "", stolen2$frame_number)
stolen2$date_time <- sub("Stolen When", "", stolen2$date_time)
stolen2$crime_ref_num <- sub("Crime Reference Number", "", stolen2$crime_ref_num)
stolen2$reward_GBP <- sub("Reward", "", stolen2$reward_GBP)
stolen2$bike_des <- sub("Bike Description", "", stolen2$bike_des)
stolen2$theft_des <- sub("Theft Description", "", stolen2$theft_des)
```

- Trim white space.

```
for (i in 1:11) {
  stolen2[, i] <- lapply(stolen2[, i], str_squish)
}
```

3.5.2 "colour" column cleaning

- Remove any single letter and non-letter of the "colour" column.

```
stolen2$colour <- tolower(stolen2$colour)
stolen2$colour <- gsub("[^[:alpha:]]|*\\b[[:alpha:]]\\b*", " ", stolen2$colour)
```

3.5.3 "location1" and "location2" columns cleaning

- Clean the location1 and location2 based on the UK regions (NUTS 1 areas).

```
nuts1 <- c("North East", "North West", "Yorkshire and The Humber", "East Midlands",
           "West Midlands", "East of England", "London", "South East", "South West",
           "Scotland", "Wales", "Northern Ireland")
```

- It is observed that the following "location2" values don't match the region list.

```
stolen2[!tolower(stolen2$location2) %in% tolower(nuts1), "location2"]
```

```
## # A tibble: 2,245 x 1
##    location2
##    <chr>
##  1 Tyne and Wear
##  2 Nottinghamshire
##  3 Essex
##  4 Wiltshire
##  5 Worcestershire
##  6 Lincolnshire
##  7 Lincolnshire
##  8 Hertfordshire
##  9 West Sussex
```

```
## 10 Essex
## # ... with 2,235 more rows
```

- While the "location1" in those rows all match the region list, which indicates location1 and location2 were input in the wrong order.

```
all(pull(stolen2[!tolower(stolen2$location2) %in% tolower(nuts1), "location1"]) %in% nuts1)
```

```
## [1] TRUE
```

- Swap the values of "location1" and "location2" in those rows.

```
stolen2[!(tolower(stolen2$location2) %in% tolower(nuts1)), c("location2", "location1")] <-
  stolen2[!(tolower(stolen2$location2) %in% tolower(nuts1)), c("location1", "location2")]
```

- " (England)" is added after the following regions, and columns are renamed.

```
location_add <-
  c("North East", "North West", "East Midlands", "West Midlands", "South East",
    "South West")

stolen2[stolen2$location2 %in% location_add, "location2"] <-
  paste(pull(stolen2[stolen2$location2 %in% location_add, "location2"]), "(England)")
```

- Rename the columns for a more readable name.

```
stolen2 <- stolen2 %>% rename(region = location2, city = location1)
```

3.5.4 Missing values

- Fill in the missing values with "0" or "Unknown".

```
stolen2$frame_number[stolen2$frame_number == ""] <- "Unknown"
stolen2$crime_ref_num[stolen2$crime_ref_num == ""] <- "Unknown"
stolen2$reward_GBP[stolen2$reward_GBP == ""] <- 0
stolen2$theft_des[stolen2$theft_des == "character(0)"] <- "Unknown"
```

3.5.5 "date_time" column cleaning

- Remove the time in am or pm as most of the values don't contain time.

```
stolen2$date_time[str_detect(stolen2$date_time, "\\s?[-:0-9.]+\\s*[ap]\\.?m\\.?") &
                    !is.na(stolen2$date_time)] <-
  stolen2$date_time[str_detect(stolen2$date_time, "\\s?[-:0-9.]+\\s*[ap]\\.?m\\.?") &
                      !is.na(stolen2$date_time)] %>%
  gsub("\\s?[-:0-9.]+\\s*[ap]\\.?m\\.?", "--", .)
```

- Convert the "date_time" column's values into standard format.

```
stolen2$date_time[!is.na(dmy(stolen2$date_time))] <-
  dmy(stolen2$date_time)[!is.na(dmy(stolen2$date_time))] %>% as.character()
```

- Use the "as_date()" function to try converting the remaining values, save the results into a new column. The unconverted values remain in the "date_time" column for reference.

```
stolen2 <- stolen2 %>% mutate(date = as_date(stolen2$date_time))
```

3.5.6 "reward_GBP" column cleaning

- Extract numbers from the string.

```
stolen2$reward_GBP <- parse_number(stolen2$reward_GBP)
```

3.5.7 bike_des" (bike description) and the "theft_des" (theft description) columns cleaning

- Remove the html tags.

```
stolen2$bike_des <- gsub("<[^>]+>", " ", stolen2$bike_des)
stolen2$theft_des <- gsub("<[^>]+>", " ", stolen2$theft_des)
```

3.5.8 Columns reorder

```
stolen2 <- stolen2[, c(1, 5, 6, 12, 7, 2, 3, 8, 9, 10, 11, 4)]
```

3.5.9 Cleaned dataset saving

```
write_csv(stolen2, "bike_stolen_clean.csv")
```

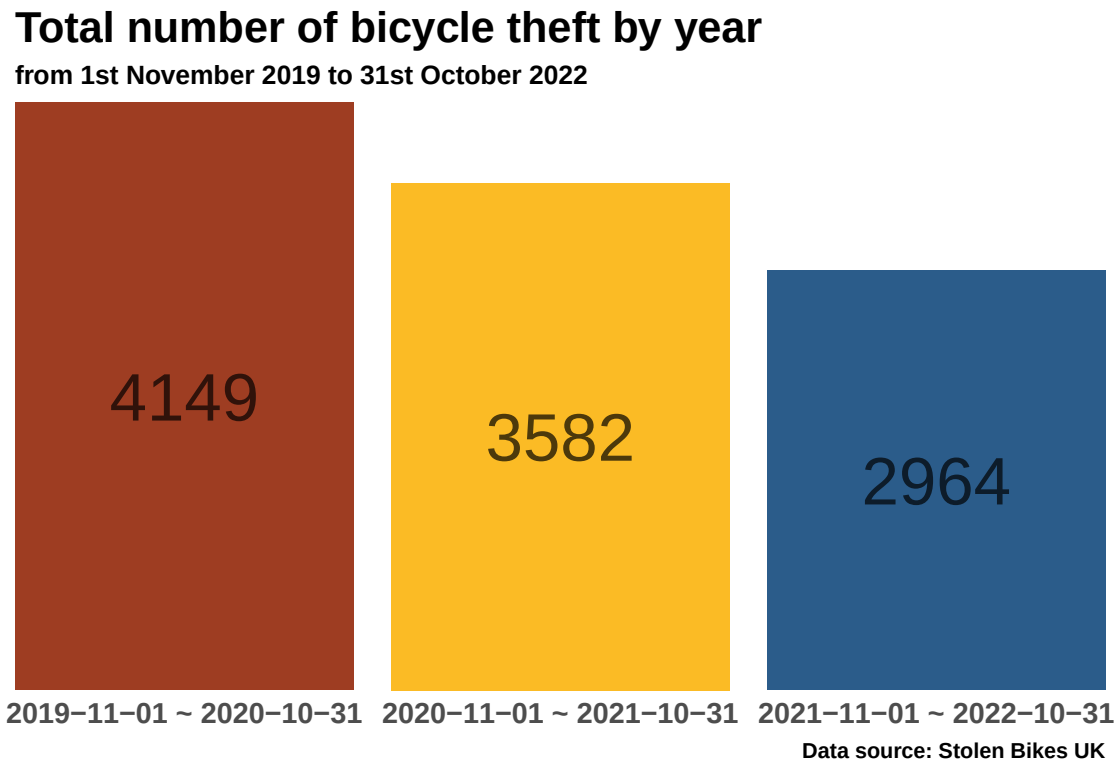
4. Data analysis and visualization

4.1 Number of bicycle theft by time

The following section will examine the number of bicycle thefts based on different time dimensions, from year and then narrow down to day.

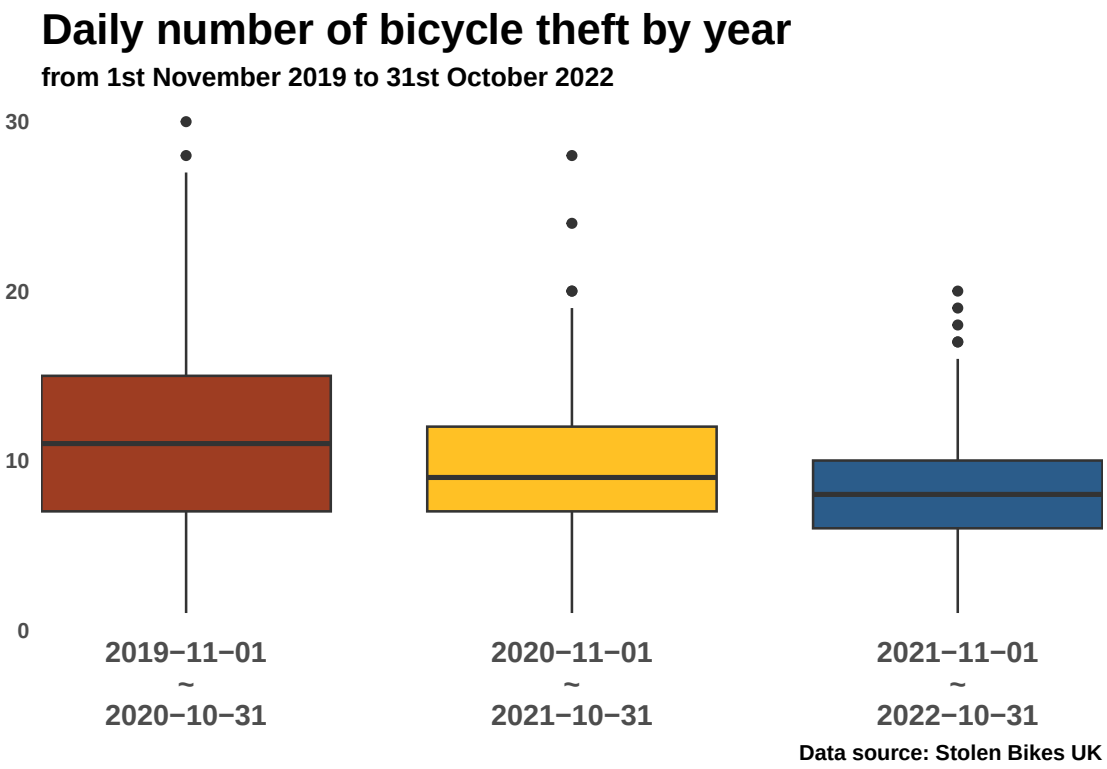
4.1.1 Number of bicycle theft by year

```
year_count <- tibble(
  year = c("2019-11-01 ~ 2020-10-31", "2020-11-01 ~ 2021-10-31",
           "2021-11-01 ~ 2022-10-31"),
  freq = c(pull(stolen2 %>% filter(date >= "2019-11-01", date <= "2020-10-31") %>%
                  count()),
           pull(stolen2 %>% filter(date >= "2020-11-01", date <= "2021-10-31") %>%
                  count()),
           pull(stolen2 %>% filter(date >= "2021-11-01", date <= "2022-10-31") %>%
                  count())))

year_count %>% ggplot(aes(x = year, y = freq, fill = freq)) +
  geom_col() + geom_text(aes(label = freq), size = 10,
                         position = position_stack(vjust = .5), alpha = .7) +
  labs(x = NULL, y = NULL, title = "Total number of bicycle theft by year",
       subtitle = "from 1st November 2019 to 31st October 2022",
       caption = "Data source: Stolen Bikes UK") +
  scale_fill_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  scale_x_discrete(expand=c(0, 0)) + scale_y_continuous(expand = c(0,0)) +
  theme_minimal() + theme(text = element_text(face = "bold"), legend.position = "none",
                          plot.title = element_text(size = 18),
                          axis.text.y=element_blank(), panel.grid = element_blank(),
                          axis.text.x = element_text(size = 12))
```



- It is observed that the total number decreases 13.7% in the second year from 4,149 to 3,582, and a further 17.3% reduction from 3,582 to 2,964 in the third year.

```
stolen2 %>% filter(date >= "2019-11-01", date <= "2022-10-31") %>% count(date)%>%
  mutate(year = case_when(date >= "2019-11-01" &
                            date <= "2020-10-31" ~ "2019-11-01 ~ 2020-10-31",
                          date >= "2020-11-01" &
                            date <= "2021-10-31" ~ "2020-11-01 ~ 2021-10-31",
                          date >= "2021-11-01" &
                            date <= "2022-10-31" ~ "2021-11-01 ~ 2022-10-31")) %>%
  ggplot(aes(x = str_wrap(year,10), y = n, fill = year)) + geom_boxplot() +
  scale_fill_manual(values=c("#9e3d22", "goldenrod1", "#2b5c8a")) +
  labs(x = NULL, y = NULL, title = "Daily number of bicycle theft by year",
       subtitle = "from 1st November 2019 to 31st October 2022",
       caption = "Data source: Stolen Bikes UK") +
  scale_x_discrete(expand=c(0, 0)) + scale_y_continuous(expand = c(0,1)) +
  theme_minimal() + theme(text = element_text(face = "bold"), legend.position = "none",
                          plot.title = element_text(size = 18),
                          panel.grid = element_blank(),
                          axis.text.x = element_text(size = 12))
```

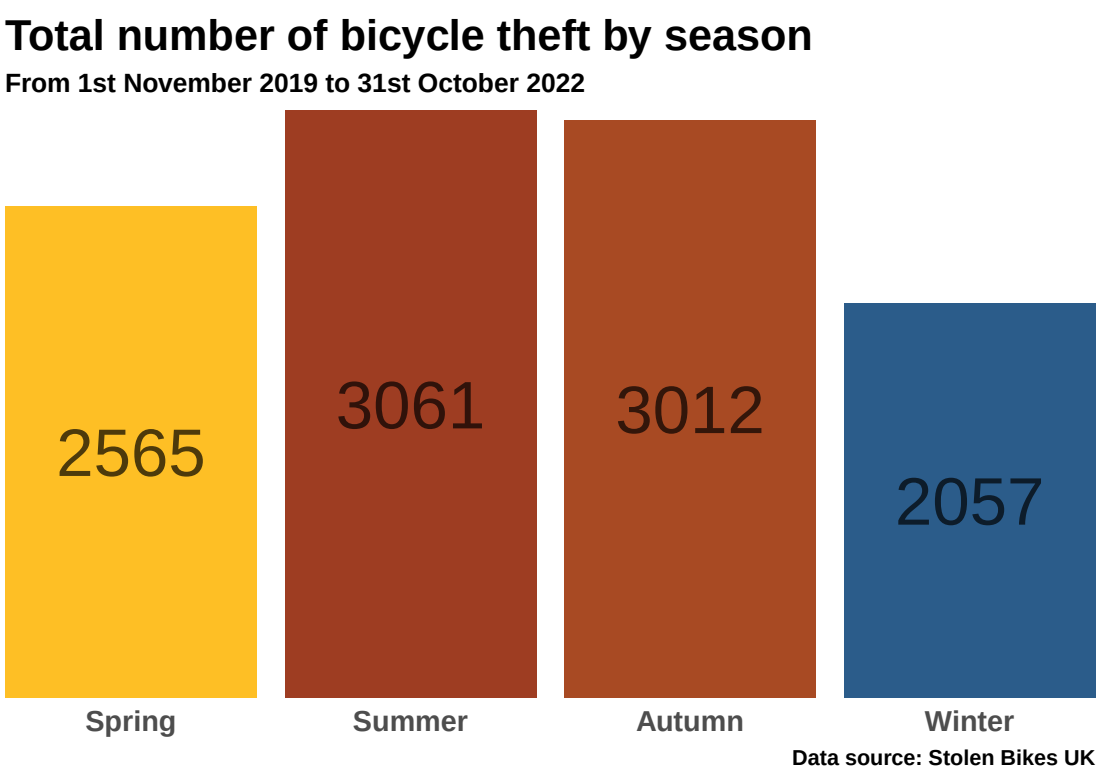


- When looking into the daily number in each year, there are also falling trends for all the maximum (from 30 cases a day to near 20), median (from above 10 cases to below 10), and majority number of cases.

4.1.2 Number of bicycle theft by season

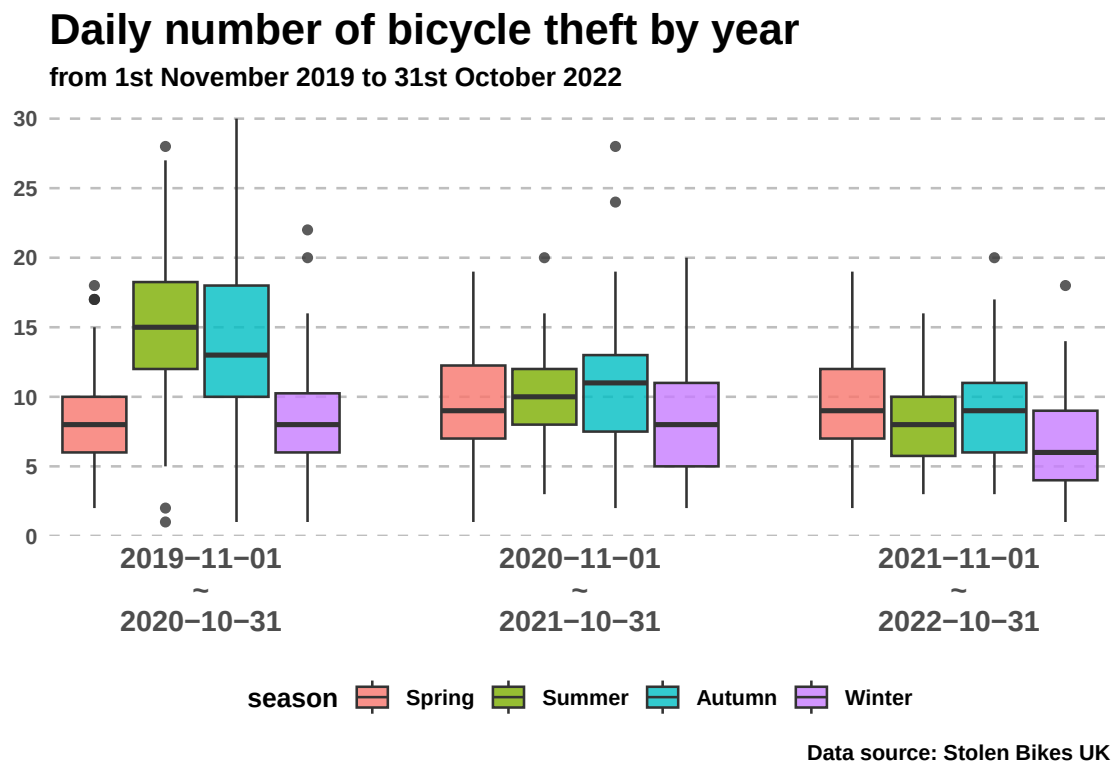
```
stolen2 <-
  stolen2 %>% mutate(
    season = case_when(
      between(month(date), 3, 5) ~ "Spring",
      between(month(date), 6, 8) ~ "Summer",
      between(month(date), 9, 11) ~ "Autumn",
      month(date) %in% c(12, 1, 2) ~ "Winter"
    ) )

stolen2$season <-
  factor(stolen2$season, levels = c("Spring", "Summer", "Autumn", "Winter"))

stolen2 %>% filter(date >= "2019-11-01", date <= "2022-10-31") %>%
  count(season) %>% ggplot(aes(x = season, y= n, fill = n)) + geom_col() +
  geom_text(aes(label = n), size = 10, position = position_stack(vjust = .5),
            alpha = .7) +
  scale_fill_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  labs(x = NULL, y = NULL,
       title = "Total number of bicycle theft by season",
       subtitle = "From 1st November 2019 to 31st October 2022",
       caption = "Data source: Stolen Bikes UK") +
  scale_x_discrete(expand=c(0, 0)) + scale_y_continuous(expand = c(0,1)) +
  theme_minimal() + theme(text = element_text(face = "bold"), legend.position = "none",
```

```
                        plot.title = element_text(size = 18),
                        axis.text.y=element_blank(), panel.grid = element_blank(),
                        axis.text.x = element_text(size = 12))
```



- Summer and autumn make up a relatively larger portion of the number, with the percentage of number held by spring, summer, autumn, winter, are 24%, 28.6%, 28.2%, and 19.2% respectively.

```
stolen2 %>% filter(date >= "2019-11-01", date <= "2022-10-31") %>% group_by(season) %>%
  count(date) %>% mutate( year = case_when(date >= "2019-11-01" &
                             date <= "2020-10-31" ~ "2019-11-01 ~ 2020-10-31",
                           date >= "2020-11-01" &
                             date <= "2021-10-31" ~ "2020-11-01 ~ 2021-10-31",
                           date >= "2021-11-01" &
                             date <= "2022-10-31" ~ "2021-11-01 ~ 2022-10-31")) %>%
  ggplot(aes(x = str_wrap(year,10), y = n, fill = season)) + geom_boxplot(alpha = 0.8) +
  labs(x = NULL, y = NULL, title = "Daily number of bicycle theft by year",
       subtitle = "from 1st November 2019 to 31st October 2022",
       caption = "Data source: Stolen Bikes UK") +
  scale_x_discrete(expand=c(0, 0.4)) + scale_y_continuous(expand = c(0,1),
                                                          breaks = seq(0, 30, 5)) +
  theme_minimal() + theme(text = element_text(face = "bold"), legend.position = "bottom",
                          plot.title = element_text(size = 18),
                          panel.grid = element_blank(),
                          panel.grid.major.y = element_line(linetype = 2,
                                                            colour = "grey"),
                          axis.text.x = element_text(size = 12))
```

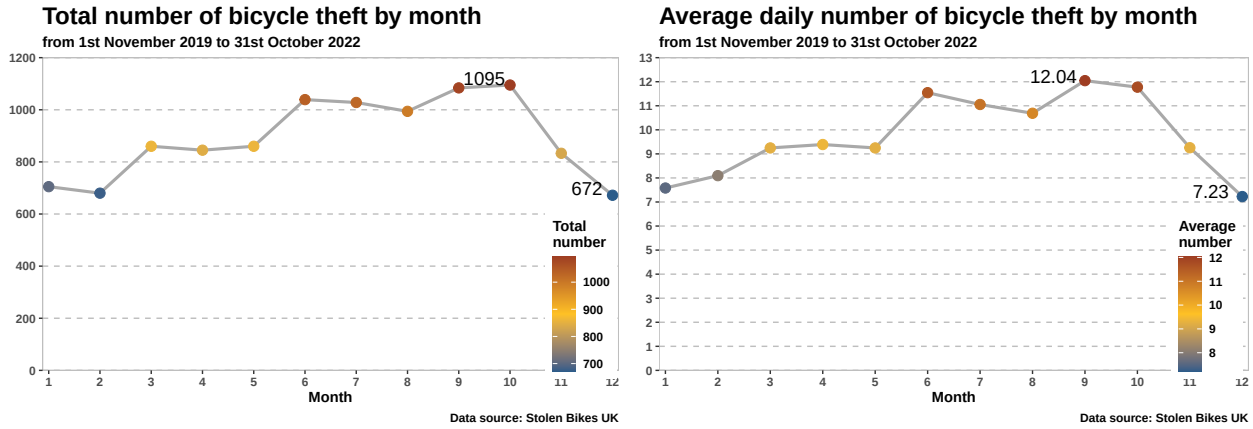


- By separating the seasonal daily number into 3 years, it can be seen that seasonal differences are relatively smaller in the past two years.
- The winter time remains the lowest number in all years, while that in the summer and autumn in the first year are particularly higher.

4.1.3 Number of bicycle theft by month

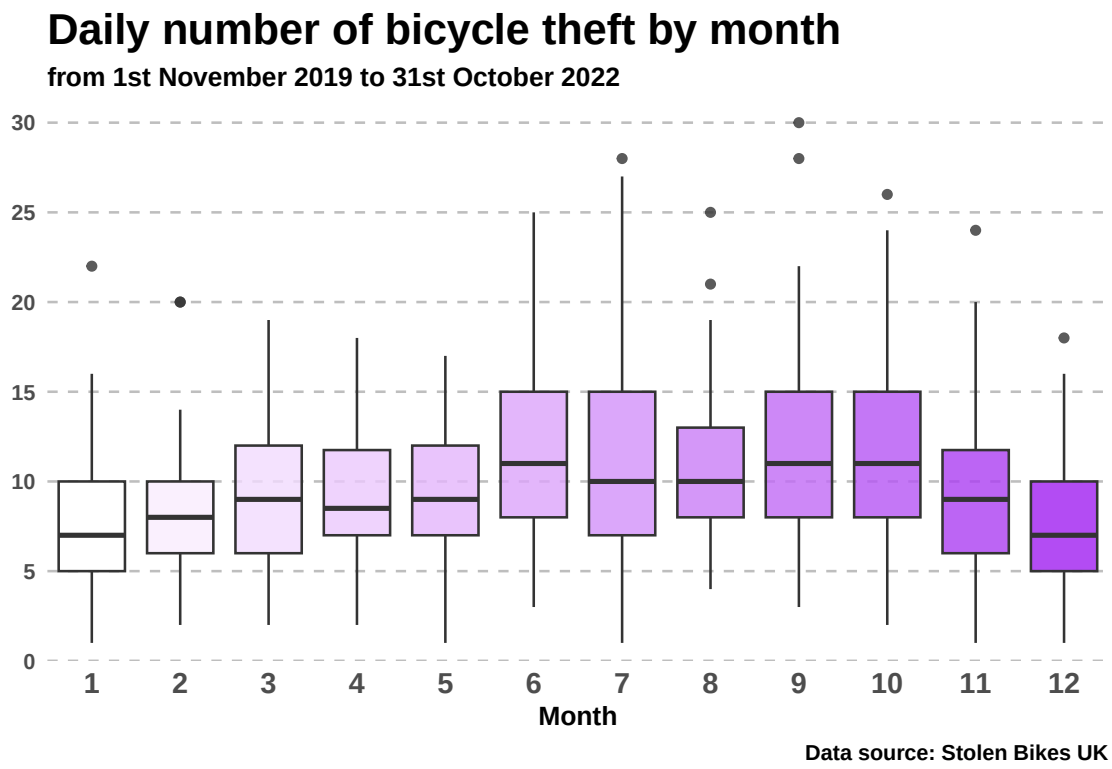
```
stolen2 %>% filter(date >= "2019-11-01", date <= "2022-10-31") %>%
  count(month = month(date)) %>% ggplot(aes(x = month, y = n)) +
  geom_line(size = 1.1, colour = "darkgrey") + geom_point(aes(colour = n), size = 3.3) +
  geom_text(data = . %>% filter(n %in% c(max(n), min(n))),
                            aes(label = n), size = 5, nudge_x = -0.5, nudge_y = 25) +
  scale_colour_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  labs(title = "Total number of bicycle theft by month",
       subtitle = "from 1st November 2019 to 31st October 2022",
       x = "Month", y = NULL, caption = "Data source: Stolen Bikes UK",
       colour = "Total\nnumber") +
  theme(text = element_text(face = "bold"), legend.position = c(0.935, 0.24),
        plot.title = element_text(size = 18),
        panel.background =element_rect(fill = "white", colour = "grey"),
        panel.grid = element_blank(),
        panel.grid.major.y = element_line(linetype = 2, colour = "grey")) +
  scale_x_continuous(expand=c(.01,.01), breaks = seq(1, 12, 1)) +
  scale_y_continuous(expand=c(0,0), breaks = seq(0, 1300, 200)) +
  expand_limits(y = c(0,1200))


stolen2 %>% filter(date >= "2019-11-01", date <= "2022-10-31") %>%
  count(month = month(date)) %>% mutate(avg = n/days_in_month(month)/3) %>%
```

```
  ggplot(aes(x = month, y = avg)) +
  geom_line(size = 1.1, colour = "darkgrey") +
  geom_point(aes(colour = avg), size = 3.3) +
  geom_text(data = . %>% filter(avg %in% c(max(avg), min(avg))),
            aes(label = round(avg,2)), size = 5, nudge_x = -0.6, nudge_y = 0.2) +
  scale_colour_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  labs(title = "Average daily number of bicycle theft by month",
       subtitle = "from 1st November 2019 to 31st October 2022",
       x = "Month", y = NULL, caption = "Data source: Stolen Bikes UK",
       colour = "Average\nnumber") +
  theme(text = element_text(face = "bold"), legend.position = c(0.93, 0.24),
        plot.title = element_text(size = 18),
        panel.background =element_rect(fill = "white", colour = "grey"),
        panel.grid = element_blank(),
        panel.grid.major.y = element_line(linetype = 2, colour = "grey")) +
  scale_x_continuous(expand=c(.01,.01), breaks = seq(1, 12, 1)) +
  scale_y_continuous(expand=c(0,0), breaks = seq(0, 13, 1)) +
  expand_limits(y = c(0,13))
```



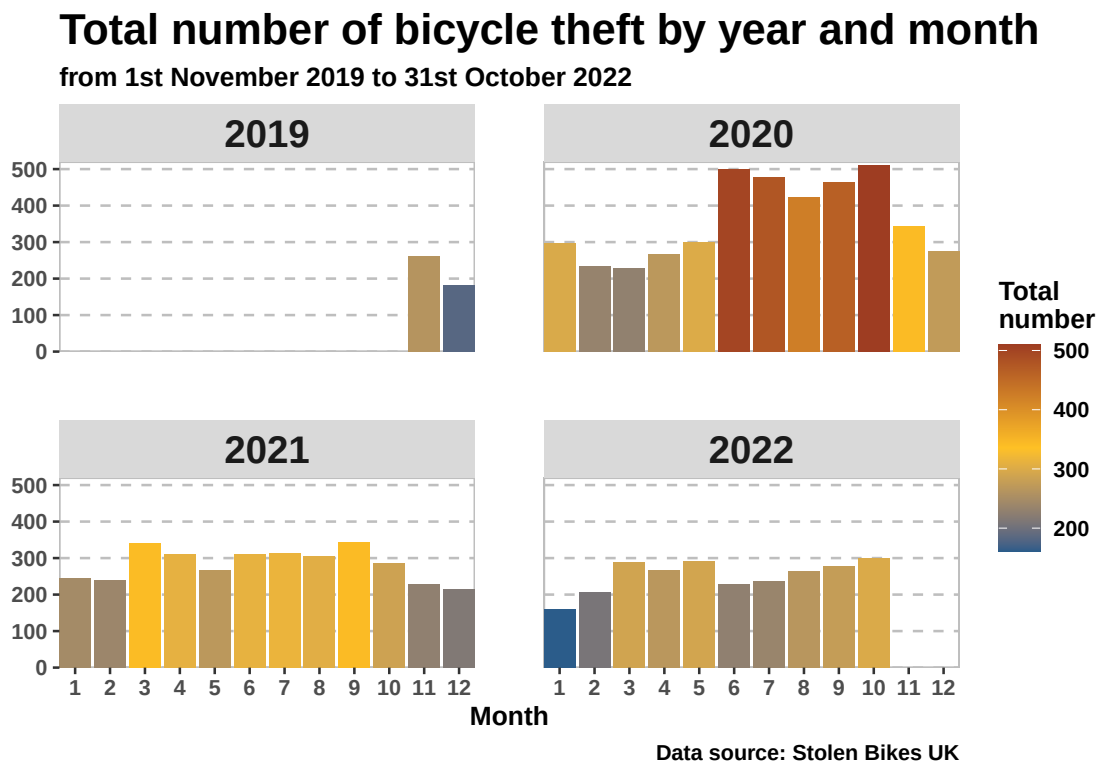
- Similar to the seasonal number, most cases line between June to October.
- There is a rising trend from the start of the year to October, followed by a sharp drop in November and December.

```
stolen2 %>% filter(date >= "2019-11-01", date <= "2022-10-31") %>%
  group_by(month = month(date)) %>% count(date) %>%
ggplot(aes(x = month, y = n, fill = month, group = month)) + geom_boxplot(alpha = 0.8) +
 scale_fill_gradient(low = "white", high = "purple") +
  labs(title = "Daily number of bicycle theft by month",
       subtitle = "from 1st November 2019 to 31st October 2022",
       x = "Month", y = NULL, caption = "Data source: Stolen Bikes UK") +
  scale_x_continuous(expand=c(.01,.01), breaks = seq(1, 12, 1)) +
  scale_y_continuous(expand = c(0,1), breaks = seq(0, 30, 5)) +
  theme_minimal() + theme(text = element_text(face = "bold"), legend.position = "none",
                          plot.title = element_text(size = 18),
                          panel.grid = element_blank(),
                          panel.grid.major.y = element_line(linetype = 2,
                                                            colour = "grey"),
                          axis.text.x = element_text(size = 12))
```



- As observed from the daily number by month, there are more higher value outliers from July to November, which means the daily number of cases could be diverging in these months, up to 30 cases a day.

```
stolen2 %>% filter(date >= "2019-11-01", date <= "2022-10-31") %>%
  group_by(year = year(date)) %>% count(month = month(date)) %>%
  ggplot(aes(x = month, y = n)) +
  geom_col(aes(fill = n), size = 1.5) +
  scale_fill_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  labs(title = "Total number of bicycle theft by year and month",
       subtitle = "from 1st November 2019 to 31st October 2022",
       x = "Month", y = NULL, caption = "Data source: Stolen Bikes UK",
       fill = "Total\nnumber") +
  theme(text = element_text(face = "bold"),
        plot.title = element_text(size = 18),
        strip.text = element_text(size=16),
        panel.background =element_rect(fill = "white", colour = "grey"),
        panel.grid = element_blank(), panel.spacing = unit(2, "lines"),
        panel.grid.major.y = element_line(linetype = 2, colour = "grey")) +
  scale_x_continuous(expand=c(0,0), breaks=seq(1,12,1)) +
  scale_y_continuous(expand=c(0,0,0,10)) + expand_limits(y = 0) +
  facet_wrap(~year)
```

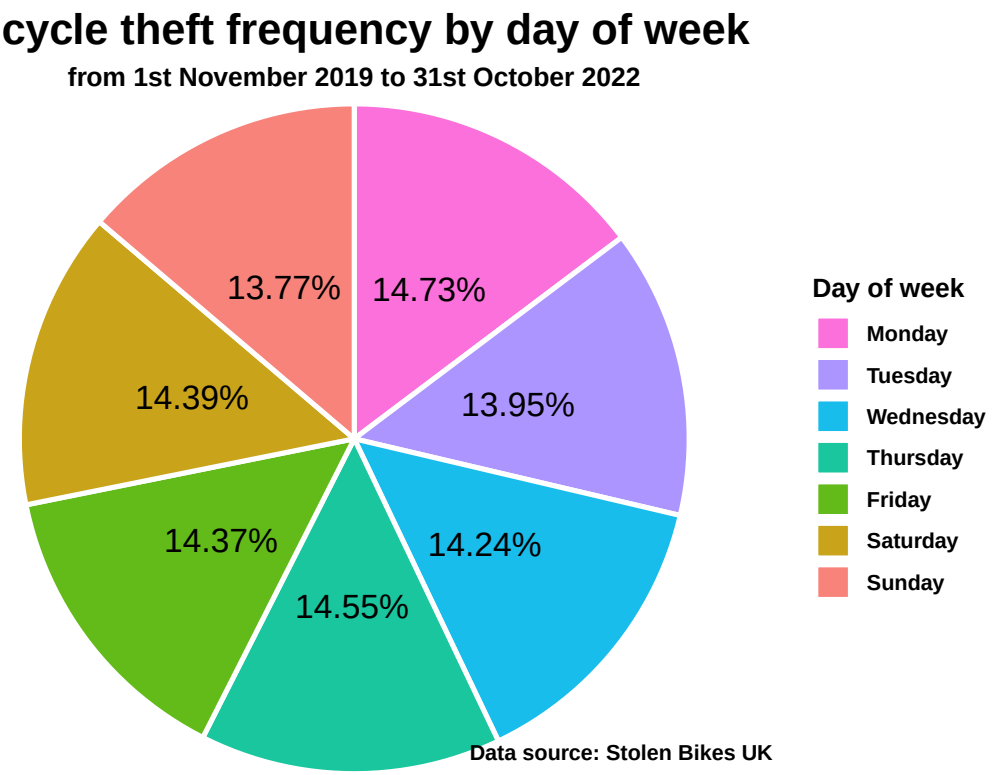


- Comparing the monthly number across different years, year 2020 generally has got higher values than that of 2021 and 2022. And numbers in 2021 and 2022 are relatively steady throughout the year.
- There is also a rising trend in the last 5 months (June to October, 2022).

4.1.4 Number of bicycle theft by day of week

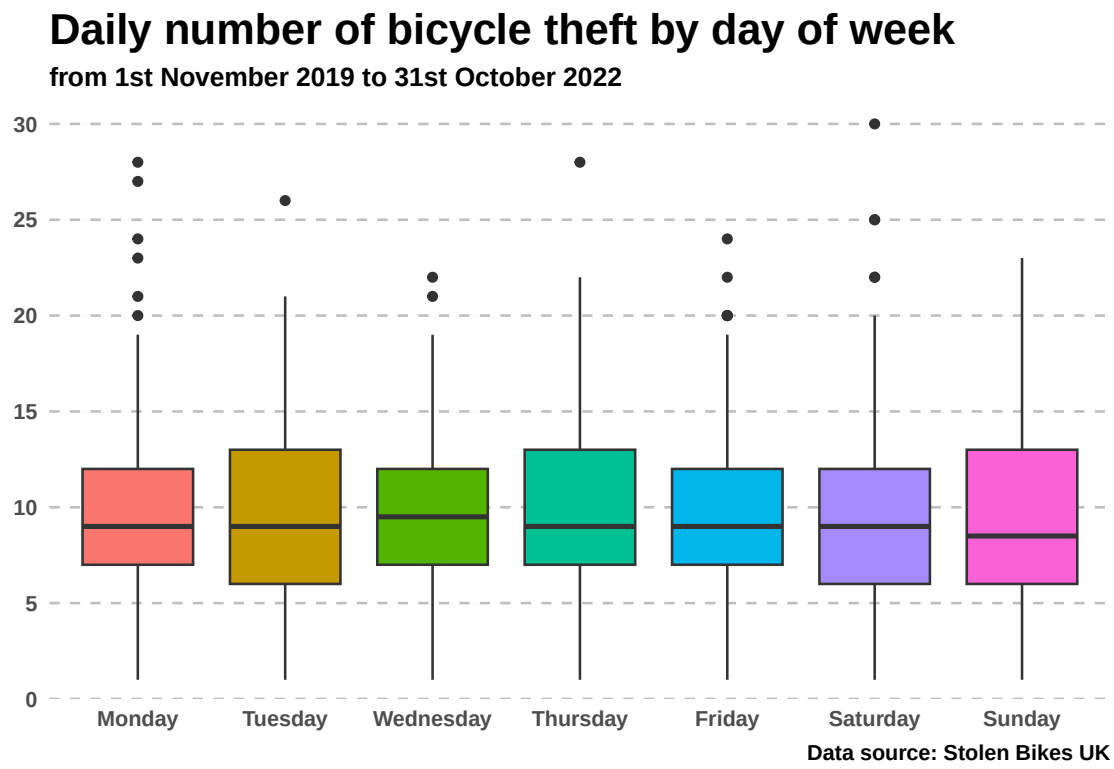
```
week_case <- stolen2 %>% filter(date >= "2019-11-01", date <= "2022-10-31") %>%
  group_by(weekday = weekdays(date)) %>% count(weekday)  %>%
  mutate(weekday =  factor(weekday, levels = c("Sunday", "Saturday", "Friday",
    "Thursday", "Wednesday", "Tuesday", "Monday"))) %>% arrange(desc(weekday))
week_case <- week_case %>%  mutate(percent = n/sum(week_case$n)*100)

week_case %>% ggplot(aes(x = "", y = percent, fill = weekday)) +
  geom_bar(width = 1, size = 1, stat = "identity", colour = "white", alpha = 0.9) +
  geom_text(aes(y = percent/2 + c(0, cumsum(percent)[-length(percent)]),
                label = paste0(round(percent,2),"%")), size=5) +
  coord_polar("y", start=0) +
  labs(title = "Bicycle theft frequency by day of week",
       subtitle = "from 1st November 2019 to 31st October 2022",
       caption = "Data source: Stolen Bikes UK",
       fill = "Day of week") +
  theme_minimal()+
  theme(text = element_text(face = "bold"),
    axis.title = element_blank(),
    axis.text = element_blank(),
    panel.border = element_blank(),
    panel.grid = element_blank(),
    axis.ticks = element_blank(),
    plot.title = element_text(size=18, hjust = 0.5),
```

```
    plot.subtitle = element_text(hjust = 0.5, margin=margin(0,0,-30,0)),
    plot.caption = element_text(margin = margin(-50,0,0,0))) +
  guides(fill = guide_legend(reverse = TRUE))
```



- The numbers of cases distribute quite evenly in all days, with Monday the highest and Sunday the lowest.

```
stolen2 %>% filter(date >= "2019-11-01", date <= "2022-10-31") %>%
  group_by(weekday = weekdays(date)) %>% count(date) %>%
  mutate(weekday =  factor(weekday, levels = c("Monday", "Tuesday", "Wednesday",
                       "Thursday", "Friday", "Saturday", "Sunday"))) %>%
ggplot(aes(x = weekday, y = n, fill = weekday, group = weekday)) + geom_boxplot() +
  labs(title = "Daily number of bicycle theft by day of week",
       subtitle = "from 1st November 2019 to 31st October 2022",
       x = NULL, y = NULL, caption = "Data source: Stolen Bikes UK") +
  scale_y_continuous(expand = c(0,1), breaks = seq(0, 30, 5)) +
  theme_minimal() + theme(text = element_text(face = "bold"), legend.position = "none",
                          plot.title = element_text(size = 18),
                          panel.grid = element_blank(),
                          panel.grid.major.y = element_line(linetype = 2,
                                                            colour = "grey"))
```



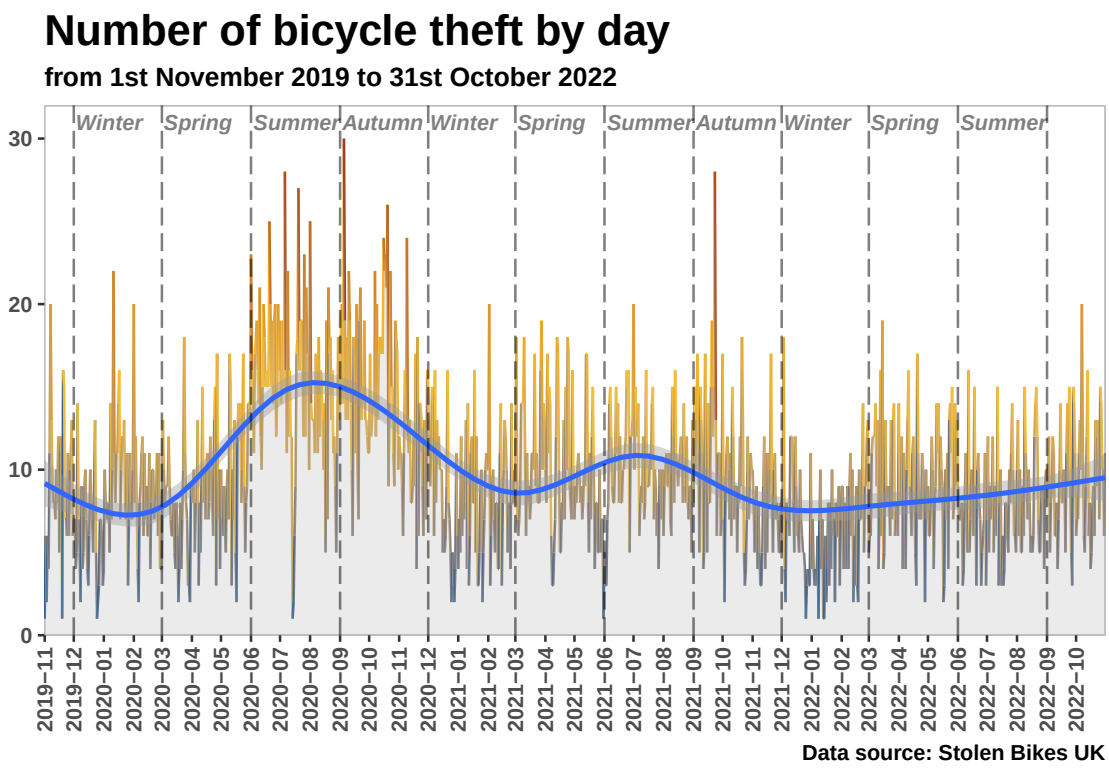
- As noticed from the daily numbers, Monday and Saturday have got some particularly higher values than other days.

4.1.5 Number of bicycle theft by day

```
stolen2 %>% filter(date >= "2019-11-01", date <= "2022-10-31") %>% count(date) %>%
  ggplot(aes(x = date, y = n)) + geom_line(aes(colour = n)) +
  geom_area(aes(group = 1), fill = "grey", alpha = 0.3) +
  scale_colour_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  labs(title = "Number of bicycle theft by day",
       subtitle = "from 1st November 2019 to 31st October 2022",
       x = NULL, y = NULL, caption = "Data source: Stolen Bikes UK") +
  theme(text = element_text(face = "bold"), legend.position = "none",
        plot.title = element_text(size = 18), axis.text.x = element_text(angle=90,
                                                                         vjust=0.5),
        panel.background =element_rect(fill = "white", colour = "grey"),
        panel.grid = element_blank()) +
  scale_x_date(expand=c(0,0), date_breaks = "1 month", date_labels = "%Y-%m") +
  scale_y_continuous(expand=c(0,0,0,1)) + geom_smooth(method = "gam") +
  geom_vline(xintercept = seq(as.Date("2019-12-1"), as.Date("2022-10-31"),
                              by = "3 month"), linetype = 5, alpha = 0.5) +
  annotate("text", x = seq(as.Date("2019-12-1"), as.Date("2022-10-31"),
                           by = "12 month"), y = 31, label = "Winter",
           hjust = -0.03, fontface = "bold.italic", colour = "black", alpha = 0.5,
           size = 3.2) +
  annotate("text", x = seq(as.Date("2020-3-1"), as.Date("2022-10-31"),
                           by = "12 month"), y = 31, label = "Spring",
           hjust = -0.03, fontface = "bold.italic", colour = "black", alpha = 0.5,
           size = 3.2) +
```

```
  annotate("text", x = seq(as.Date("2020-6-1"), as.Date("2022-10-31"),
                           by = "12 month"), y = 31, label = "Summer",
           hjust = -0.03, fontface = "bold.italic", colour = "black", alpha = 0.5,
           size = 3.2) +
  annotate("text", x = seq(as.Date("2020-9-1"), as.Date("2021-10-31"),
                           by = "12 month"), y = 31, label = "Autumn",
           hjust = -0.03, fontface = "bold.italic", colour = "black", alpha = 0.5,
           size = 3.2)
```

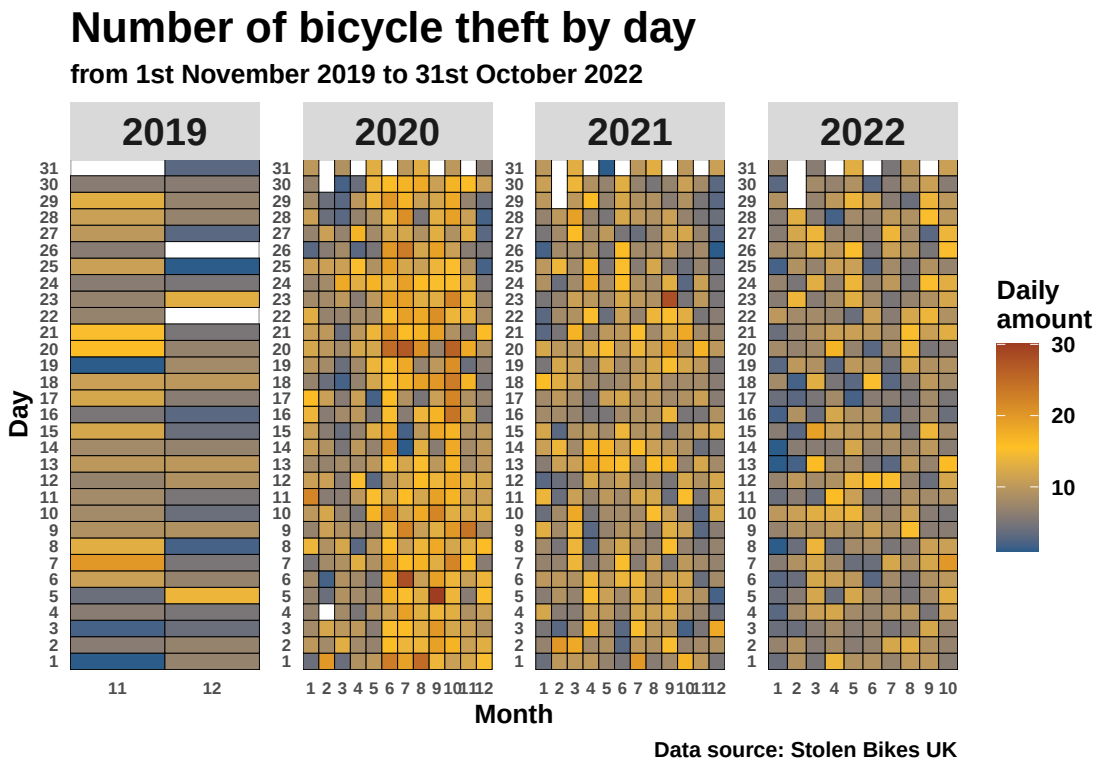
```
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```



- When reviewing the number of cases by day, it can also be seen that larger numbers happened in summer and autumn in each year, with relatively fewer in year 2021 than that in 2020.
- Winter remains the fewest.

```
stolen2 %>% filter(date >= "2019-11-01", date <= "2022-10-31") %>% count(date) %>%
  ggplot(aes(x = month(date) ,y = day(date), fill = n)) +
  geom_tile(colour = "black") + facet_wrap(~year(date), nrow = 1, scales = "free") +
  scale_fill_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  labs(title = "Number of bicycle theft by day",
       subtitle = "from 1st November 2019 to 31st October 2022",
       x = "Month", y = "Day", fill = "Daily\namount",
       caption = "Data source: Stolen Bikes UK") +
  theme(text = element_text(face = "bold"), legend.position = "right",
        plot.title = element_text(size = 18), axis.ticks = element_blank(),
        axis.text = element_text(size = 7), strip.text = element_text(size=16),
        panel.background =element_rect(fill = "white", colour = "grey"),
        panel.grid = element_blank()) +  scale_y_continuous(expand=c(0,0),
                                                            breaks=seq(1,31,1)) +
```

```
  scale_x_continuous(expand = c(0,0), breaks=seq(1,12,1))
```



- It is observed that most cases happened in the second half of year 2020.
- Generally year 2022 (until October) has fewer number than 2021.

4.2 Number of bicycle theft by location

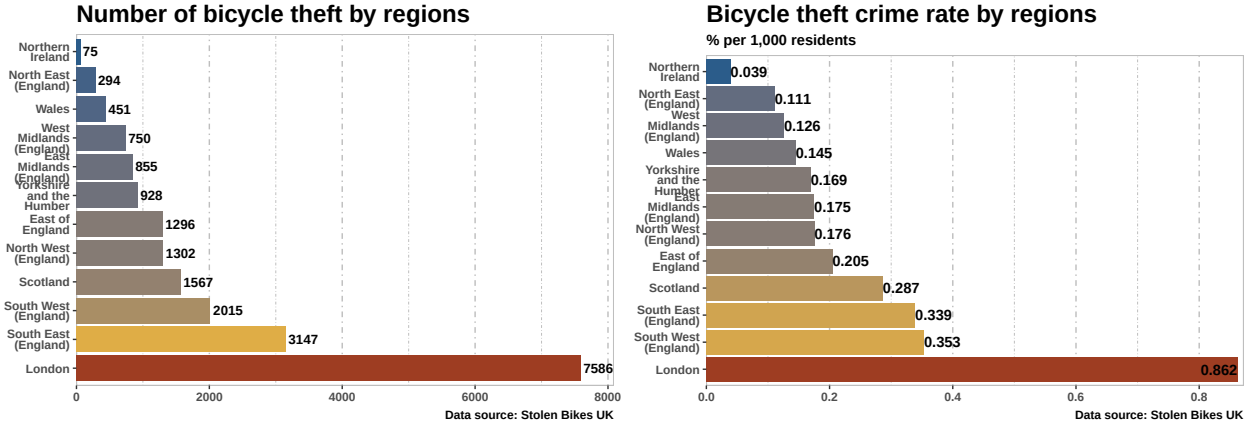
This section will examine the number of bicycle theft based on region and city.

4.2.1 Number of bicycle theft by regions

- Populations of each region were obtained from the Office for National Statistics and the Northern Ireland Statistics and Research Agency).

```
uk_population <-
  data.frame(
    area = c("London", "South East (England)", "West Midlands (England)",
      "South West (England)", "North West (England)", "Northern Ireland",
      "East of England", "Scotland", "Wales", "East Midlands (England)",
      "Yorkshire and the Humber", "North East (England)"),
    population = c(8799800, 9278100, 5950800, 5701200, 7417300, 1903175,
      6334500, 5466000, 3107500, 4880200, 5480800, 2647100)) %>%
  arrange(area)

stolen2 %>% count(region) %>% arrange(desc(n)) %>%
  ggplot(aes(y = fct_inorder(str_wrap(region,10)), x = n, fill = n)) + geom_col() +
  geom_text(aes(label = n), hjust=-0.1, fontface = "bold", size = 3.5) +
  scale_fill_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  labs(x = NULL, y = NULL,
       title = "Number of bicycle theft by regions",
```

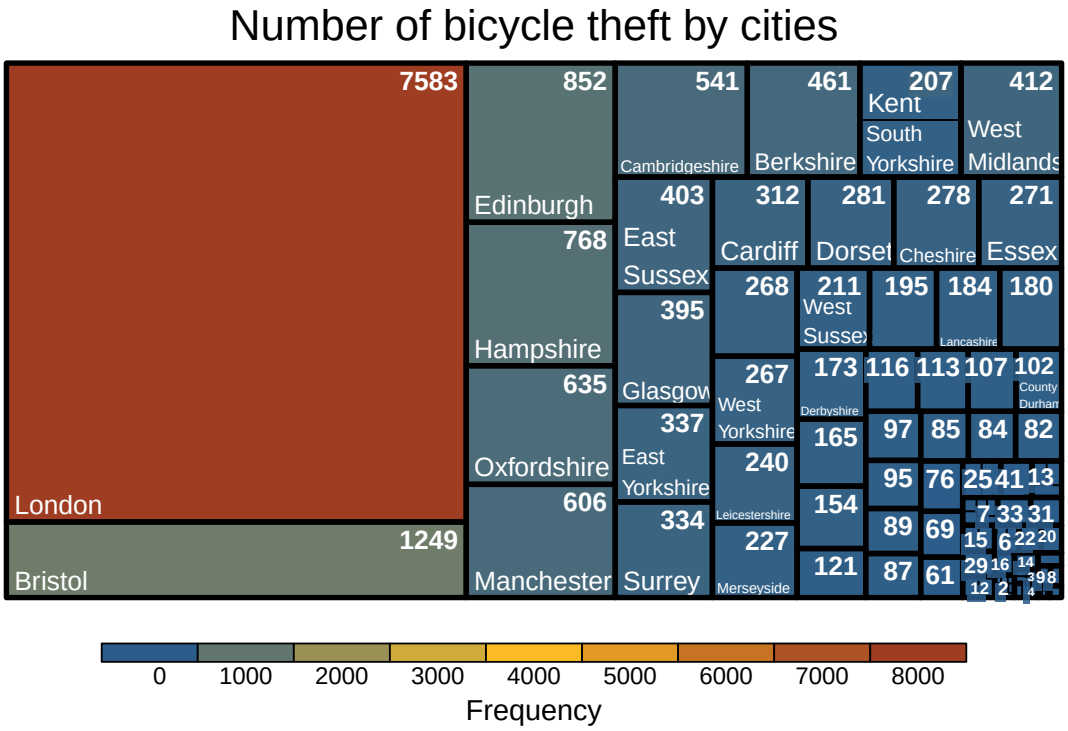
```
       caption = "Data source: Stolen Bikes UK") +
  theme(text = element_text(face = "bold"), legend.position = "none",
        plot.title = element_text(size = 18),
        axis.text.y = element_text(lineheight = 0.75),
        panel.background =element_rect(fill = "white", colour = "grey"),
        panel.grid = element_blank(), panel.grid.major.x = element_line(colour = "grey",
                                                                         linetype = 4),
        panel.grid.minor.x = element_line(colour = "grey", linetype = 4)) +
  scale_x_continuous(expand=c(0,0,0,500))

stolen2 %>% count(region) %>% arrange() %>%
  mutate(percent_in_pop = n / uk_population[, "population"] * 1000) %>%
  arrange(desc(percent_in_pop)) %>%
  ggplot(aes(y = fct_inorder(str_wrap(region,10)), x = percent_in_pop,
             fill = percent_in_pop)) +
  geom_col() + geom_text(aes(label = round(percent_in_pop,3)), hjust="inward",
                         fontface = "bold", size = 4) +
  scale_fill_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  labs(x = NULL, y = NULL,
       title = "Bicycle theft crime rate by regions", subtitle = "% per 1,000 residents",
       caption = "Data source: Stolen Bikes UK") +
  theme(text = element_text(face = "bold"), legend.position = "none",
        plot.title = element_text(size = 18),
        axis.text.y = element_text(lineheight = 0.75),
        panel.background =element_rect(fill = "white", colour = "grey"),
        panel.grid = element_blank(), panel.grid.major.x = element_line(colour = "grey",
                                                                         linetype = 4),
        panel.grid.minor.x = element_line(colour = "grey", linetype = 4)) +
  scale_x_continuous(expand=c(0,0,0,0.01))
```



- Obviously London has got the highest number and higher than other regions by factors of 2.4 to over 100.
- But when population is taken into consideration, with the number of cases per 1,000 residents, the factors decrease to 2.4 to 22. South West (England) becomes the second place. Northern Ireland remains with the lowest value.

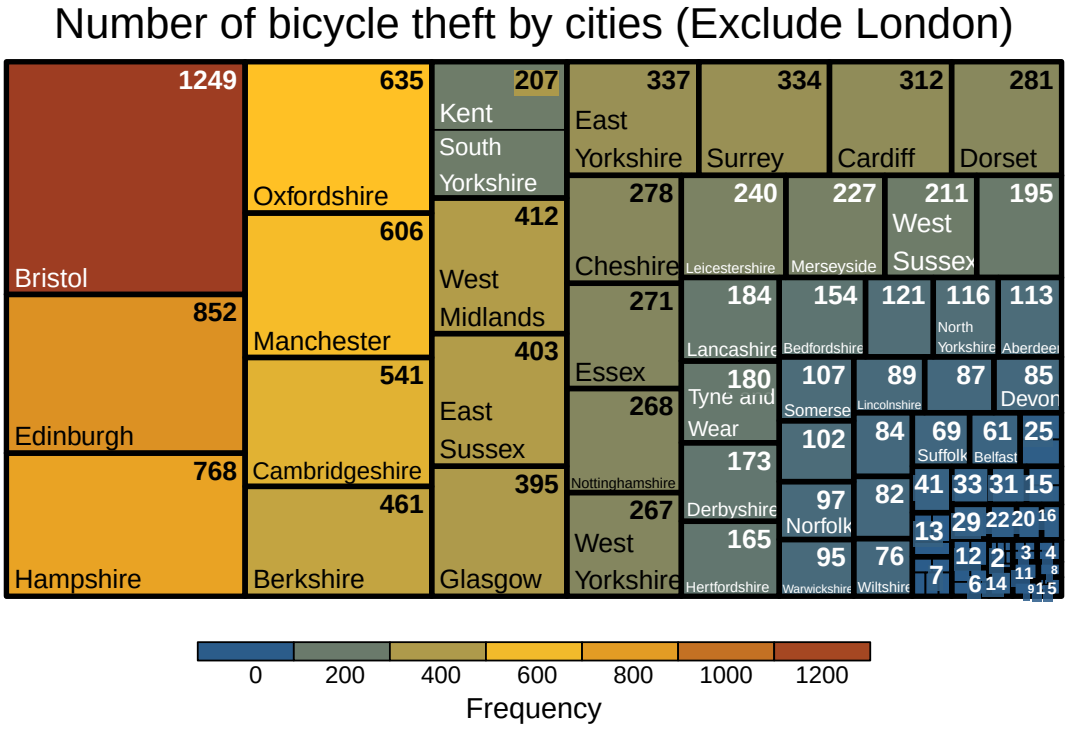
4.2.2 Number of bicycle theft by cities

```
stolen2 %>% count(city) %>%
  treemap::treemap(index = c("n", "city"),
                   vSize = "n",
                   vColor = "n",
                   palette = c("#2b5c8a", "goldenrod1", "#9e3d22"),
                   type = "manual",
                   align.labels = list(c("right", "top"), c("left", "bottom")),
                   overlap.labels=0.2,
                   title = "Number of bicycle theft by cities",
                   fontsize.title = 18,
                   title.legend = "Frequency")
```



- Among the cities, London still has the greatest number.
- As expected from the previous bar charts, cities from the South East (England), South West (England), Scotland, North West (England) occupy the higher rankings.

```
stolen2 %>% filter(city != "London") %>% count(city) %>%
  treemap::treemap(index = c("n", "city"),
                   vSize = "n",
                   vColor = "n",
                   palette = c("#2b5c8a", "goldenrod1", "#9e3d22"),
                   type = "manual",
                   align.labels = list(c("right", "top"), c("left", "bottom")),
                   overlap.labels=0.2,
                   title = "Number of bicycle theft by cities (Exclude London)",
                   fontsize.title = 18,
                   title.legend = "Frequency")
```



- When taking out London from the list, the differences among different cities are more obvious by the colour representation.

4.3 Colours of stolen bicycles

Let's take a look into the colours of the stolen bicycles.

4.3.1 Colours of stolen bicycle by word frequency

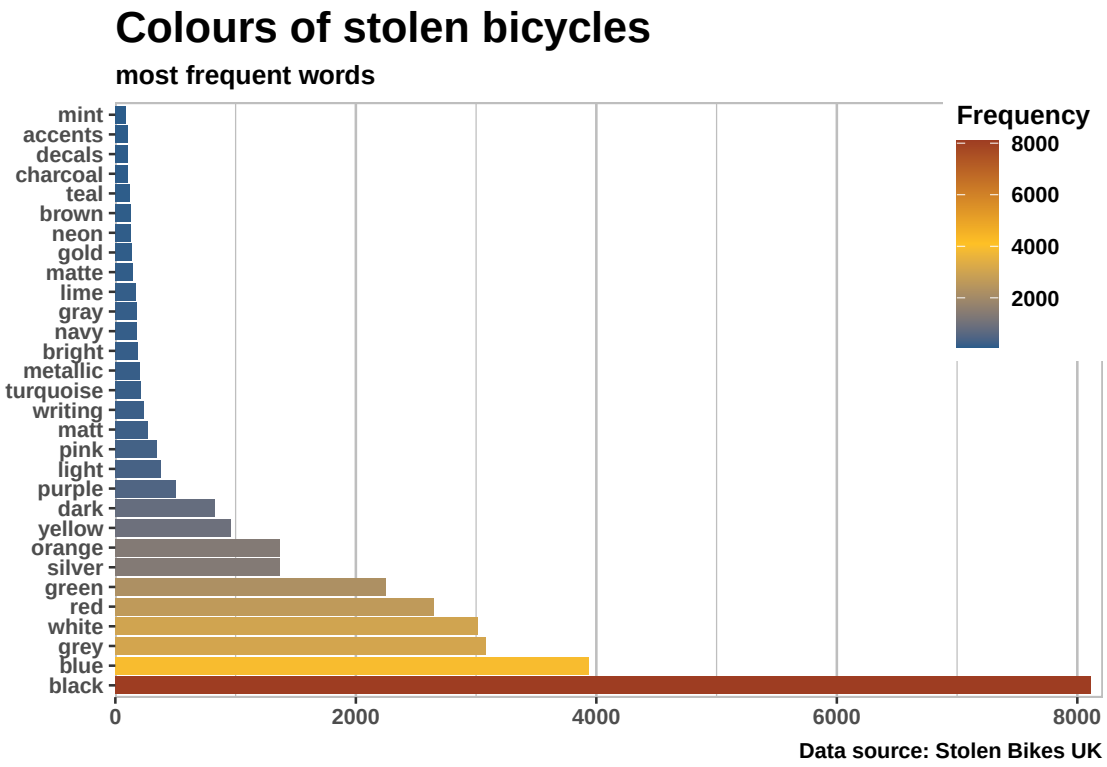
- Create function "word_summary" for counting words in the colour description.

```
word_summary <- function (data, exclude = NA, remove = "[?]") {
  tibble(text = data) %>%
    mutate(text = tolower(text)) %>%
    mutate(text = str_remove_all(text, remove)) %>%
    mutate(word = str_split(text, "\\s+")) %>%
    unnest(cols = c(word)) %>%
    filter(!word %in% exclude) %>%
    count(word) %>%
    mutate(percent = n / sum(n)) %>%
    rename(freq = n) %>%
    arrange(desc(freq))
  }
```

- Count the words in the "colour" column and omit some unrelated words that may appear.
- Plot chart with the top 30 words that appear the most frequent.

```
colour_word <-
  word_summary(stolen2$colour, c("into", "frame", "and", "for", "the", "in",
    "was", "from", "to", "of", "on", "it", "my", "at", "with", ""))
```

```
colour_word[1:30,] %>% ggplot(aes(x = freq, y = fct_inorder(word), fill = freq)) +
  geom_col() + labs(x = NULL, y = NULL,
                    title = "Colours of stolen bicycles",
                    subtitle = "most frequent words", fill = "Frequency",
       caption = "Data source: Stolen Bikes UK") +
  scale_fill_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  scale_x_continuous(expand=c(0, 0, 0, 100)) +
  theme(text = element_text(face = "bold"), legend.position = c(0.92,0.795),
        plot.title = element_text(size = 18),
        panel.background =element_rect(fill = "white", colour = "grey"),
        panel.grid = element_blank(), panel.grid.major.x = element_line(colour = "grey"),
        panel.grid.minor.x = element_line(colour = "grey"))
```

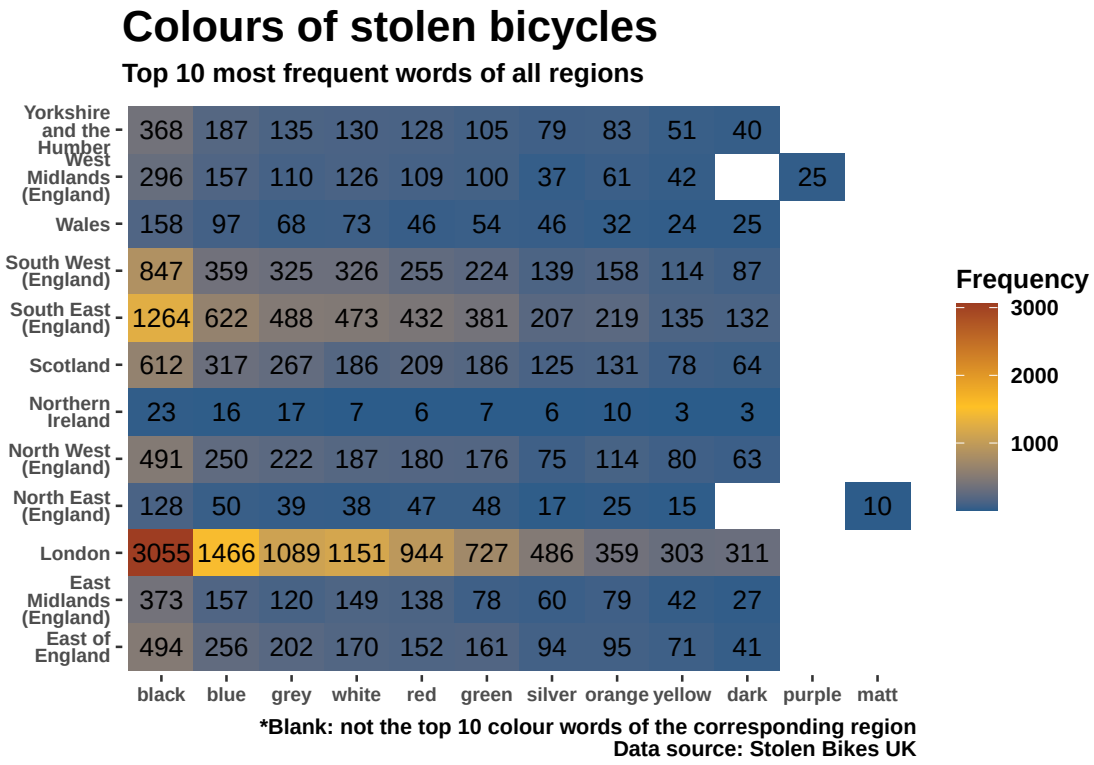


- Apparently black colour is the most popular in the thefts.
- But it is important to take into account that this may be just because more black colour bicycles exist in the country and it just reflects the popularity among the buyers but not the thieves.
- Combining data from the sales market is necessary for further verification.

4.3.2 Top colours of stolen bicycle by regions

```
for (i in 1:12) {
  if (i == 1) {colour_nuts1 <- tibble()}
  colour_nuts1 <-
    rbind(colour_nuts1, (word_summary(
      stolen2$colour[stolen2$region ==unique(stolen2$region)[i]],
      c("into", "frame", "and", "for", "the", "in", "was", "from", "to", "of",
        "on", "it", "my", "at", "with", "")) %>%
        mutate(area = unique(stolen2$region)[i]))[1:10, ])
```

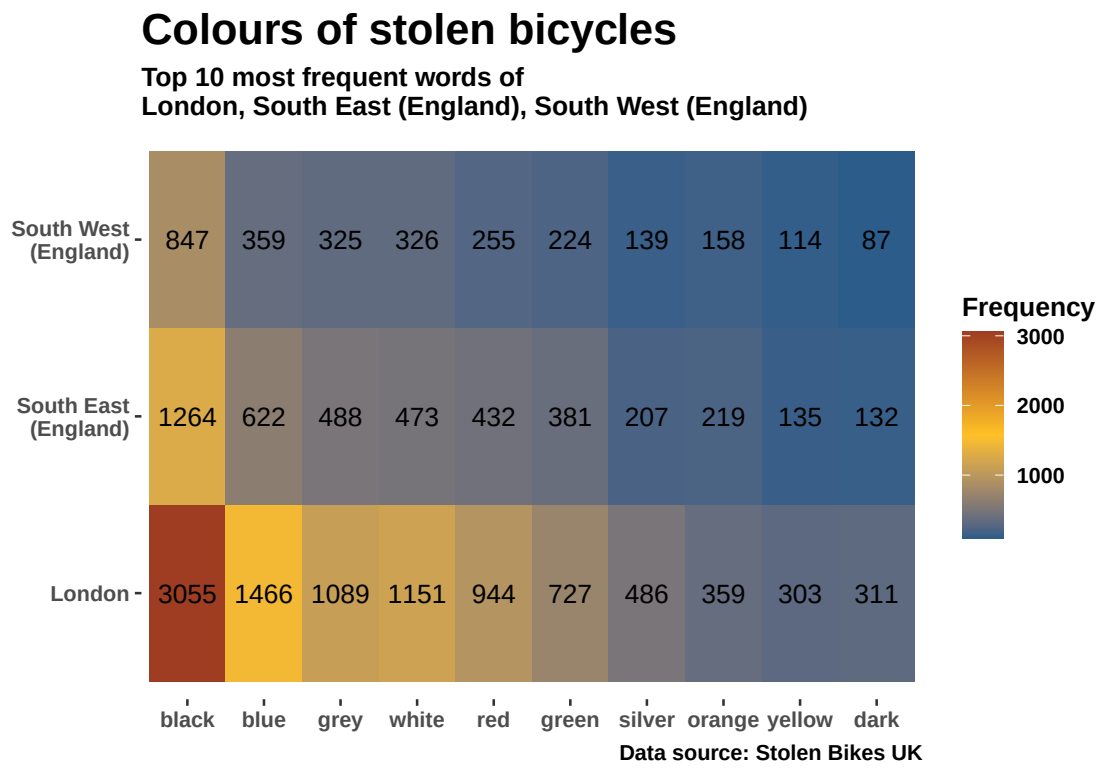
```
}

colour_nuts1$word <- factor(colour_nuts1$word, levels = colour_word$word)

colour_nuts1 %>% ggplot(aes(word, str_wrap(area,10))) + geom_tile(aes(fill = freq)) +
  geom_text(aes(label = freq)) +
  scale_fill_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
labs(x = NULL, y = NULL, fill = "Frequency",
     title = "Colours of stolen bicycles",
     subtitle = "Top 10 most frequent words of all regions",
     caption = str_wrap("*Blank: not the top 10 colour words of the corresponding
                        region Data source: Stolen Bikes UK",65)) +
  theme(text = element_text(face = "bold"),
        axis.text.x = element_text(size = 8),
        axis.text.y = element_text(size = 8, lineheight = 0.75),
        plot.title = element_text(size = 18), panel.background = element_blank())
```



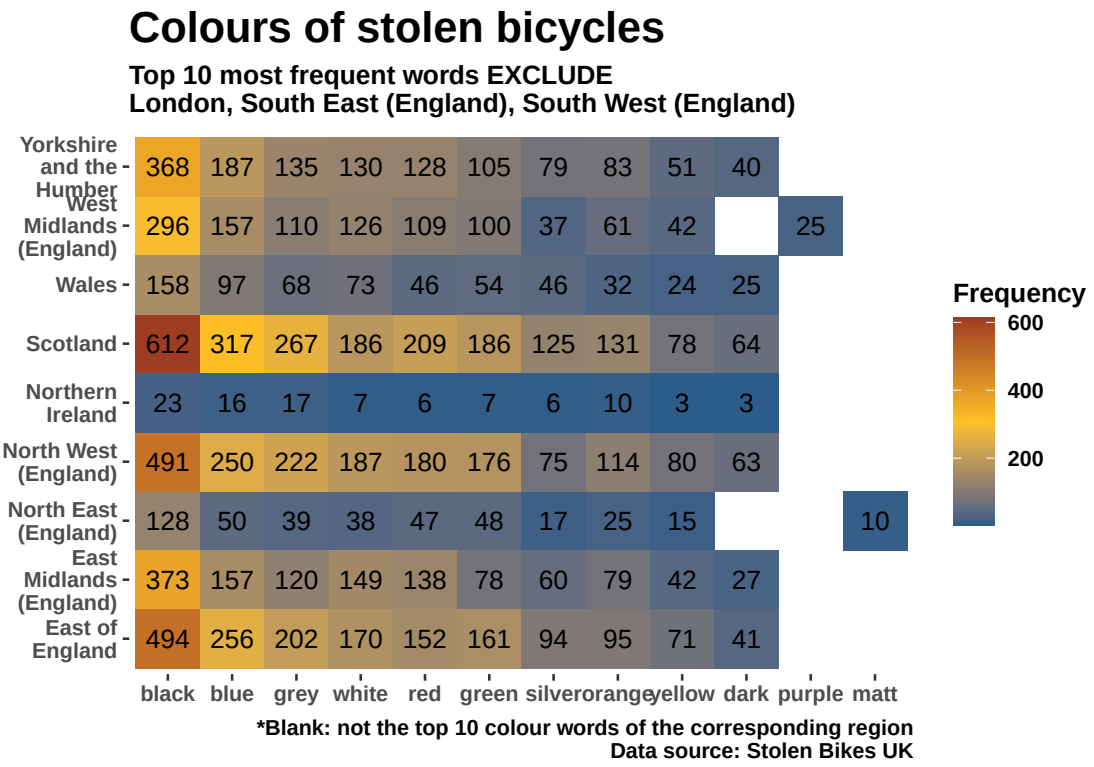
- The top colours among different regions are similar to the previous chart.
- With no surprise that London has got the greatest numbers in all colours across all regions.

```
colour_nuts1 %>% filter(area %in% c("London", "South East (England)",
                                    "South West (England)")) %>%
  ggplot(aes(word, str_wrap(area,10))) + geom_tile(aes(fill = freq)) +
  geom_text(aes(label = freq)) +
  scale_fill_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  labs(x = NULL, y = NULL, fill = "Frequency",
       title = "Colours of stolen bicycles",
subtitle = "Top 10 most frequent words of \nLondon, South East (England), South West (England)",
       caption = "Data source: Stolen Bikes UK") +
```

```
  theme(text = element_text(face = "bold"),
        plot.title = element_text(size = 18),
        panel.background = element_blank())
```




```
colour_nuts1 %>% filter(!(area %in% c("London", "South East (England)",
                                      "South West (England)"))) %>%
  ggplot(aes(word, str_wrap(area,10))) + geom_tile(aes(fill = freq)) +
  geom_text(aes(label = freq)) +
  scale_fill_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  labs(x = NULL, y = NULL, fill = "Frequency",
       title = "Colours of stolen bicycles",
subtitle = "Top 10 most frequent words EXCLUDE \nLondon, South East (England), South West (England)",
       caption = str_wrap("*Blank: not the top 10 colour words of the corresponding
                          region Data source: Stolen Bikes UK",65)) +
  theme(text = element_text(face = "bold"),
        plot.title = element_text(size = 18),
        panel.background = element_blank())
```



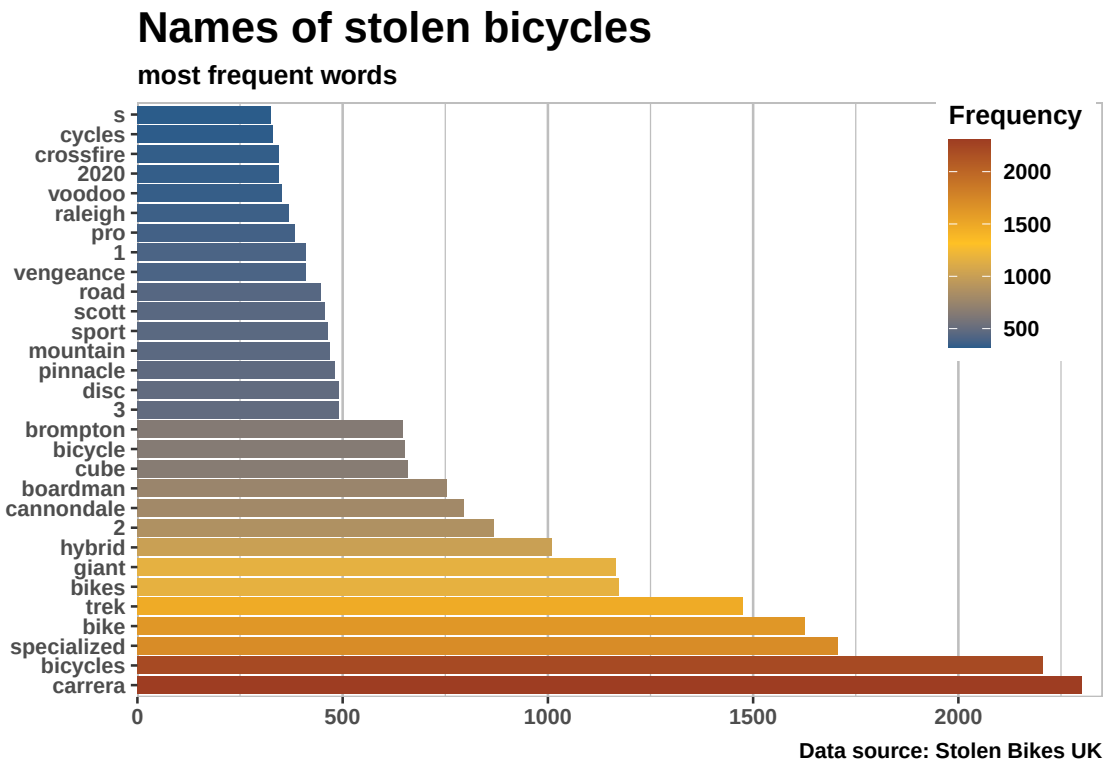
- Separate the top 3 regions with the highest number of word frequency from other regions for a better colour representation.

4.4 Names of stolen bicycles

The names may provide information on the manufacturer, brand, model, or other relevant details.

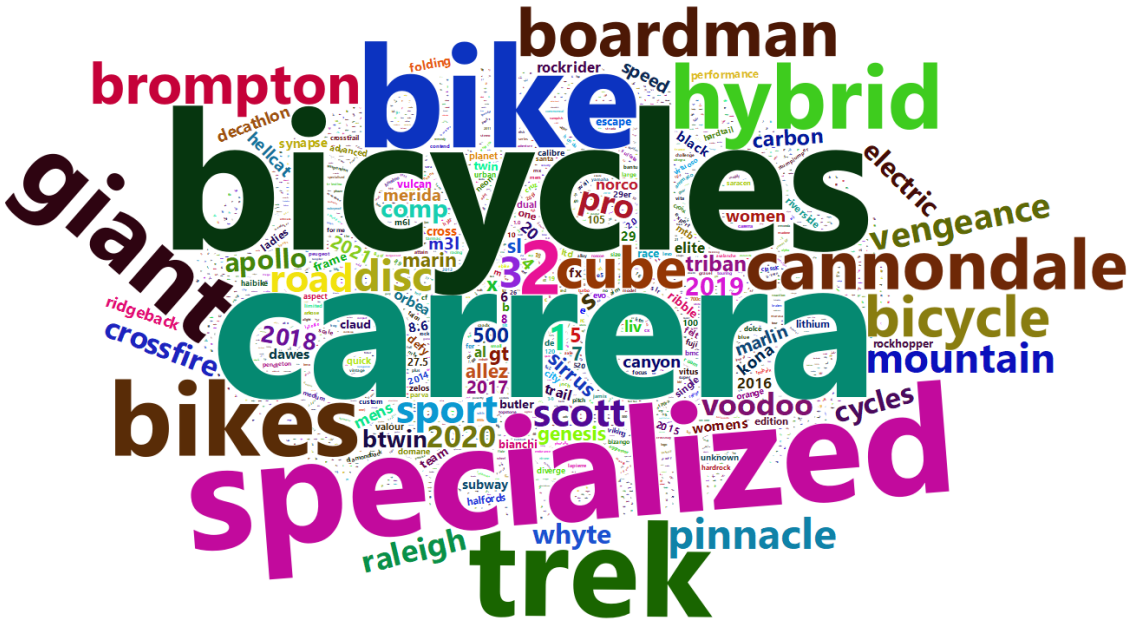
4.4.1 names of stolen bicycle by word frequency

```
bikename_word <- word_summary(stolen2$bike_name)

bikename_word[1:30,] %>% ggplot(aes(x = freq, y = fct_inorder(word), fill = freq)) +
  geom_col() + scale_fill_gradientn(colours = c("#2b5c8a", "goldenrod1", "#9e3d22")) +
  labs(x = NULL, y = NULL,
       title = "Names of stolen bicycles", subtitle = "most frequent words",
       caption = "Data source: Stolen Bikes UK", fill = "Frequency") +
  scale_x_continuous(expand=c(0, 0, 0, 50)) +
  theme(text = element_text(face = "bold"), legend.position = c(0.91, 0.795),
        plot.title = element_text(size = 18),
        panel.background =element_rect(fill = "white", colour = "grey"),
        panel.grid = element_blank(), panel.grid.major.x = element_line(colour = "grey"),
        panel.grid.minor.x = element_line(colour = "grey"))
```



- It is not surprising that some of the big bicycle brands are found in the list, with "Carrera" comes in first place, followed by "Specialized", "Trek", "Giant" and "Cannondale".
- For the bicycle type, "hybrid" is at a top rank. "Disc" may refer to disc brakes (compared to rim brakes, a type of brake technology in the wheel). "Mountain" and "sport" bicycle also appear as one of the most frequent words.

```
wordcloud2::wordcloud2(bikename_word, size = 1.3)
```



* The wordcloud provides more figures on the less frequent words, with the size of each word relative to the number of frequency.

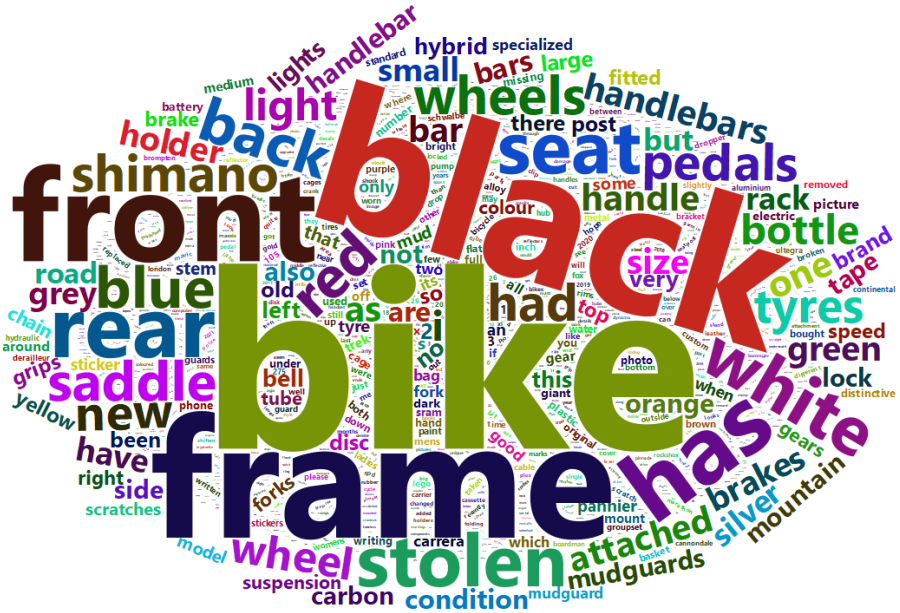
4.5 Description of stolen bicycles

More details of the stolen bicycles are provided in this section by the owners.

4.5.1 Description of stolen bicycle by word frequency

- Count the words in the "bike_des" column and omit some unrelated words that may appear.

```
bike_word <- word_summary(stolen2$bike_des,
                          c("a", "and", "for", "the", "in", "was", "from", "to", "of",
                            "on", "it", "my", "at", "with", "unknown", "is", "", "or",
                            "by", "be"), "[[:punct:]]")

wordcloud2::wordcloud2(bike_word, size = 1.6)
```



* It can be seen that except the names and colours, more description on different parts such as “frame”, “wheels”, “saddle”, “seat”, “pedals”, “handlebar”, e.t.c. are found. * Some bicycles may also have equipment like “bell”, “sticker”, and “lights”, or special systems like “suspension” and “hydraulic”.

4.6 Description of theft

Details of the thefts are also vital clues for recovering the bicycles.

4.6.1 Description of theft by word frequency

- Count the words in the “theft_des” column and omit some unrelated words that may appear.

```
theft_word <- word_summary(stolen2$theft_des,
                           c("a", "and", "for", "the", "in", "was", "from", "to", "of",
                             "on", "it", "my", "at", "with", "unknown", "is", "", "or",
                             "by", "be"), "[[:punct:]]")

wordcloud2::wordcloud2(theft_word, size = 2.8)
```

* Except the top words "bike" and "stolen", it seems many of the bicycles in the thefts were already "locked", "park" "outside" on the "road"/"street", near or in "garage"/"garden"/"house"/"building"/"station", but were being "cut", and higher chance at "morning", "night", or "overnight".

4.7 Average reward of bicycle recovery

Finally let's take a look on the average reward provided in different locations (If you want to be a bounty hunter :))

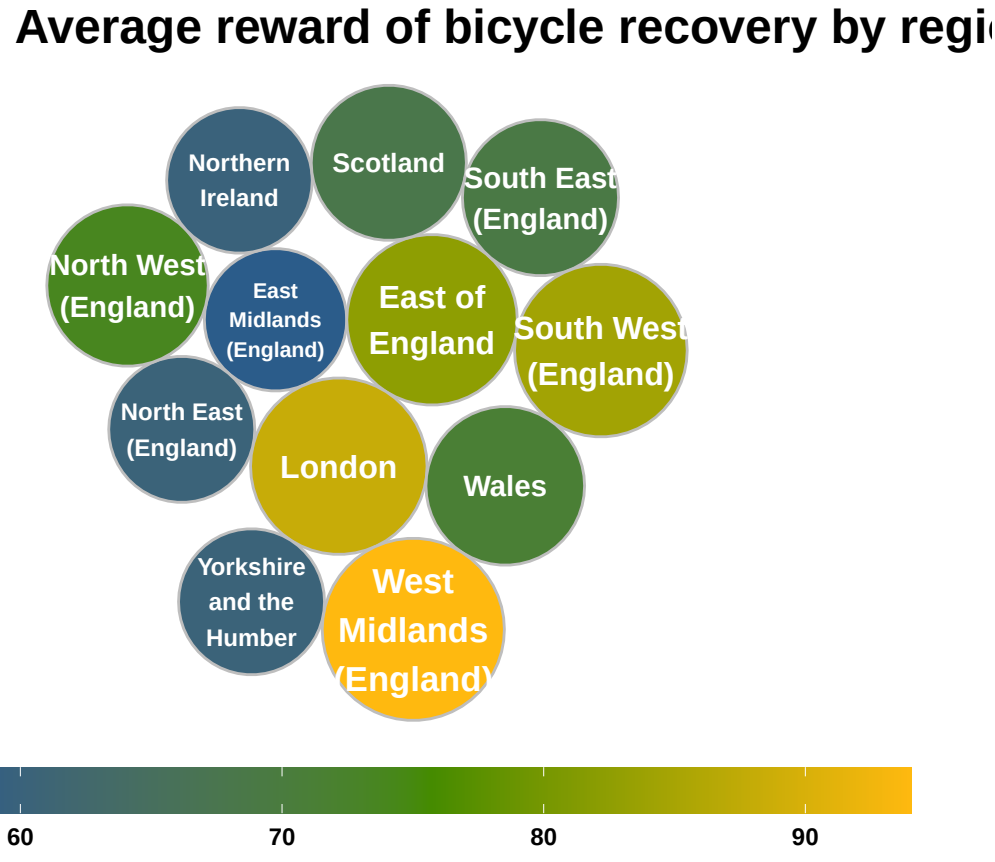
```
stolen2 %>% filter(reward_GBP >= 2e4) %>%  arrange(-reward_GBP) %>% print(n =Inf)
```

```
## # A tibble: 11 x 13
##    bike_name     colour frame~1 date       date_~2 city  region crime~3 reward~4
##    <chr>         <chr>  <chr>   <date>     <chr>   <chr> <chr>  <chr>      <dbl>
##  1 TFL 221       red    221240  NA         9292    Lond~ London 345678~ 1   e215
##  2 Pinnacle Hyb~ black  Unknown NA         Overni~ Manc~ North~ awaiti~ 7.39e  9
##  3 Trek Mountai~ black~ 1037u0~ 2017-08-19 2017-0~ Berk~ South~ 431702~ 7.38e  9
##  4 Ghost Miss 1~ white~ Wcr110~ 2017-08-19 2017-0~ Berk~ South~ 431702~ 7.38e  9
##  5 Honda 2018    black  123456  NA         Last y~ Lond~ London 12345   1.23e  8
##  6 Cube Reactio~ deser~ 58132-~ 2021-08-19 2021-0~ Glas~ Scotl~ Ab0584~ 5.84e  6
##  7 Oxylane Tilt~ grey   000001~ 2022-02-28 2022-0~ Lond~ London Unknown 4.21e  6
##  8 Aist Bicycle~ black~ Yp1902~ 2021-02-05 2021-0~ Lond~ London 190245~ 1.90e  6
##  9 Carrera bicy~ black  1537658 NA         at ur ~ West~ South~ 357634~ 1   e  6
## 10 Cube sterieo  grey   Unknown NA         today   Coun~ North~ Unknown 1   e  6
## 11 Eddy Merckx ~ pink   4356756 NA         yester~ Lond~ London 4654    1.4 e  5
## # ... with 4 more variables: bike_des <chr>, theft_des <chr>, link <chr>,
## #   season <fct>, and abbreviated variable names 1: frame_number, 2: date_time,
## #   3: crime_ref_num, 4: reward_GBP
```

- It appears that some of the reward amounts are unusually large that they might be wrongly input, only values below GBP 20,000 are considered for the following visuals.

4.7.1 Average reward of bicycle recovery by regions

```
library(packcircles)

reward_region <- (stolen2 %>% filter(reward_GBP < 2e4) %>% select(region, reward_GBP) %>%
                    group_by(region) %>% summarize(average_reward = mean(reward_GBP),0))
packing2 <- circleProgressiveLayout(reward_region$average_reward, sizetype="area")
reward_region <- cbind(reward_region, packing2)
reward_region.gg <- circleLayoutVertices(packing2, npoints=50)
reward_region.gg$value <- rep(reward_region$average_reward, each = 51)

ggplot() + geom_polygon(data = reward_region.gg,
                        aes(x, y, group = id, fill = value),
                        colour = "grey") +
  geom_text(data = reward_region,
            aes(x, y, size = average_reward, label = str_wrap(region, 10)),
            colour = "white", fontface = "bold") +
  labs(title = "Average reward of bicycle recovery by regions",
       fill = "Average\nreward\n(GBP)") +
  scale_size_continuous(range = c(2.7287,4.5)) + theme_void() +
  theme(text = element_text(face = "bold"),
        plot.title = element_text(size = 18),
        legend.position = "bottom",
        legend.key.width = unit(5, "line")) +
  coord_equal() +
  scale_fill_gradientn(colours = c("#2b5c8a", "chartreuse4", "darkgoldenrod1")) +
  guides(colour = guide_legend("value"), size = "none")
```

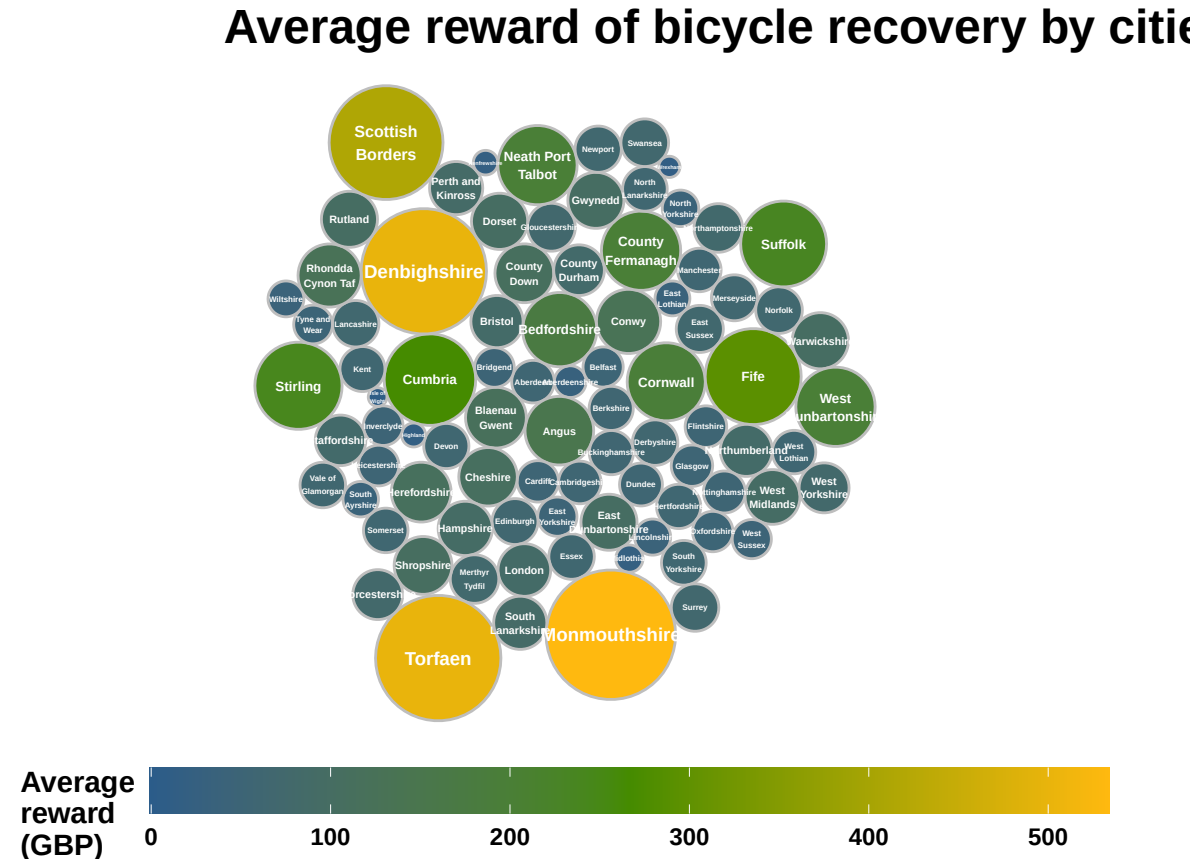


- The range of average reward is not much, running from nearly GBP60 to more than GBP90.
- West Midlands (England) is at the top, London and South West (England) come afterwards.

4.7.2 Average reward of bicycle recovery by cities

```
reward_city <- (stolen2 %>% filter(reward_GBP < 2e4) %>% select(city, reward_GBP) %>%
  group_by(city) %>% summarize(average_reward = mean(reward_GBP)))
packing <- circleProgressiveLayout(reward_city$average_reward, sizetype="area")
reward_city <- cbind(reward_city, packing)
reward_city.gg <- circleLayoutVertices(packing, npoints=50)
reward_city.gg$value <- rep(reward_city$average_reward, each = 51)

ggplot() + geom_polygon(data = reward_city.gg,
                        aes(x, y, group = id, fill = value),
                        colour = "grey") +
  geom_text(data = reward_city,
            aes(x, y, size = average_reward, label = str_wrap(city, 10)),
            colour = "white", fontface = "bold") +
  labs(title = "Average reward of bicycle recovery by cities",
       fill = "Average\nreward\n(GBP)") +
  scale_size_continuous(range = c(0.5,2.4)) + theme_void() +
  theme(text = element_text(face = "bold"),
    plot.title = element_text(size = 18),
    legend.position = "bottom",
    legend.key.width = unit(5, "line")) +
  coord_equal() +
```

```
scale_fill_gradientn(colours = c("#2b5c8a", "chartreuse4", "darkgoldenrod1")) +
guides(colour = guide_legend("value"), size = "none")
```



- The range of average reward by cities is relatively higher, from no reward to GBP500 up. The top four cities are Monmouthshire (Wales), Denbighshire (Wales), Torfaen (Wales), and Scottish Borders (Scotland).

5. Key findings and conclusion

Community safety is always one of the vital concerns for people living there. It is encouraging to see the total number of bicycle thefts are dropping in the past three years. But there is the possibility that there may be correlation between this observation and the condition of fewer people who cycle to work due to more employers offering work-from-home to employees.

Summer and autumn time are likely to have more stolen cases, and it usually comes to the greatest in September and October. Although the monthly figures of year 2022 are mostly lower than those in the previous two years, bicycle owners still should not take it lightly as it is observed that year 2022 is having a steady rising trend since June.

There is not really a distinctive "preference" for the thieves to commit a crime on any day of a week, with Monday and Saturday which could have up to 30 cases a day.

In terms of the geographical distribution, London is having the greatest amount of bicycle thefts, and Northern Ireland can be considered as the "safest" region.

Bicycles in black and blue colour are the most popular colours that are found in the thefts, while as mentioned earlier it may just reflect the popularity among the buyers but not the thieves. The rankings of the top 10

frequent colours in all regions are very similar.

Owners of bicycle brands “Carrera”, “Specialized”, “Trek”, “Giant” and “Cannondale” are relatively more often to be the victims. “Hybrid” and “disc” bicycles also frequently appeared in thefts.

In many cases the bicycles had already been locked but still being cut, which indicates owners may need to consider a strong type of lock or lock it in a safer location. Avoid parking the bicycles in the morning, night, or overnight, may also help to reduce the chance of being stolen.

With the aid of web scraping, up-to-date data could be obtained quickly for analysis. Bicycle theft is happening around us thus it is important to understand the situation. Hopefully this can help individuals or even the whole community to get to know the story behind the data and gain useful insight for making immediate decisions for any improvement.