Plastic Pollution in Oceans
Group 2 Report - CMM507

ALEXANDER RITCHIE, *1911218@rgu.ac.uk*; GEORGIOS ORFANAKIS, *1903446@rgu.ac.uk*;KAREN JEWELL, *1415410@rgu.ac.uk*;ROSHI SHRESTHA, *1903445@rgu.ac.uk*;STUART WATT, *1501869@rgu.ac.uk*

April 7, 2020

Objective

- To understand the composition of plastic pollutants in the ocean
- To understand the sources of plastic pollutants
- To understand how plastic pollution gets distributed across the oceans

1 Problem Statement

H1 = The % of plastic pollution remains constant over time.
H0 = The % of plastic pollution does not remain constant over time.

1.1 Overview

Marine pollution is a major global issue which impacts on environment, economy and human health. Although marine pollution is caused by many different materials, plastics consist of 60-80% of the marine litter (Derraik, 2002; Reisser, 2015, Barboza et al., 2019).
Synthetic organic polymer derived from polymerisation of monomers extracted from oil and gas make up the plastics (Derraik, 2002, Rios et al., 2007). The lightweight feature and its durability make it very suitable to make a range of products that we use in our everyday life (Barnes et al., 2009; Sivan 2011). These same features have been a major cause of pollution due to overuse and non-managed waste disposal system worldwide with plastic contributing to the 10% of the waste generated worldwide (Barnes et al., 2009). Due to its buoyancy, plastic debris can be dispersed over long distances and they can persist for a long time. Although, plastic litter has been a major cause of marine pollution for a while, its seriousness has only been realised recently. Jambeck et al., (2015) reported that in 2010 alone, between 4.8 million to 12.7 million metric tons of plastics entered the ocean. Plastics are now everywhere in the marine environment and urgent action is required to mitigate this problem and reduce the harmful impact (Rios et al., 2007; Rochman et al., 2015).

1.2 Motivation

Impact on marine life Plastics in ocean is one of the many forms of human impact that threatens marine life. There is still very little information available on the impact of plastic pollution on the ocean's ecosystem. Due to the realisation on impact of human on climate and environment, there has been a lot of awareness

activities to reduce the impact of pollution. Ban on single use plastic bags are being applied to many countries in order to protect the environment.
Over 700 marine wildlife species are affected due to entanglement in plastic ropes and materials and ingestion of plastics in the ocean (Gall and Thompson, 2015). Over 340 species of marine animals were found to be entangled (Kuhn et al., 2015). Reducing plastic waste is a major challenge worldwide. It is almost impossible to estimate the number of marine animals affected by marine pollution globally due to the vastness of the ocean. However, studies carried out on the gut contents of thousands of seabirds, found the significant increase in the ingestion of plastics during the 10-15 years interval (Robards et al., 1995). This result might correlate to the rapid increase of plastic production and plastic use globally. In a study carried out over fourteen years, Moser and Lee (1992) found that more 50% of the seabird species contained plastic particles in the gut which increased over time. This could be due the increase in plastic availability over time. Entanglement in plastic debris is another cause of marine life suffering. Discarded fishing gear and floating mastic masses in ocean are serious threat to marine animals. Some animals such as seals are attracted to the floating plastics where they get entangled and get suffocated. Harmful effect of litter on marine life has been reviewed extensively (Ryan, 2015; Kuhn et al., 2015; Gall, and Thompson 2015; Williams and Rangel-Buitragen, 2019). Floating plastics over long distances can disperse alien species as well as some pathogens. Drifting plastic debris are also the source of alien species introduction and thus affecting the native marine biodiversity (Gregory, 2009; Kiessling et al., 2015).
Impact on environment and human health Plastic debris floating in the oceans and the littering the coastal areas are not a pleasant sight. Masses of plastic accumulation and discarded objects made from plastics are found everywhere nowadays.
Over time plastic disintegrates into small microplastics which are easily consumed by fish and they enter the food chain. Plastics have been found in a third of fish caught in the UK which included the popular fishes such as cod, haddock and mackerel. Impact of plastic entering the human food chain and the effects of it are still to be studied. Plastic toxicity and the occurrence of microplastics and nanoplastics in the water supply can also be a direct impact on human health in addition to the contamination in seafood (Rochman et al., 2015; Markic et al., 2019).
Reducing plastic pollution has recently been a global aim. Research in plastic pollution in marine environment has played a big role in reducing it and raising awareness all over the world. In order to understand the plastic pollution in marine environments and its effect in long term, it is essential to keep collecting data on patterns of marine debris around the world. Effective monitoring of plastic debris is very essential in order to reduce the abundance of plastic debris everywhere. In addition, monitoring the type, frequency and the source of the litter is also important for prevention initiative of marine pollution. Most of the monitoring are done by surveys looking at frequencies of beach litter collected by organisations and volunteers (Coe and Rodgers, 1997). Most abundant litter can be found close to urban areas where beach visitor numbers are higher (Garrity and Levings, 1993).

1.3 Objectives

The main objectives of this project can be outlined as follows:

2 Research

Things we found citation example [**8489087**].
Sources of pollution: 10 river dataset, 50km2 coastline dataset, pollution density and body of water dataset....

3 Methods

This paper is conducted using secondary data collection methods only. The authors did not collect or create any new data using primary methods.

3.1 Dataset Description

- The data was taken from marine debris tracker (marinedebris.engr.uga.edu/newmap/) between 2010 till February 19th 2020. The time of 2010 was chosen as there was no data before that time.
- The dataset was composed by combining the multiple csv files gathered from the marine debris tracker into a single set after this was done the date data type was renamed "Time".
- The dataset created from the combined csv files contain more than 360000 rows of data and consists of the folowing variables.
  - ListID is the ID code for the list
  - ListName is the name of the list
  - ItemID is the ID code given to the item of debris
  - ItemName is the name we give to item of debris
  - LogID is the ID code given to the location of the debris
  - Latitude, Longitude and Altitude are the coordinates of the location where the observation was made
  - Quantity is the number of pieces of debris in the observation.
  - Error radius is the radius around the observation site within the error for reasonable doubt.
  - Location is the area the observation of debris was made in.
  - Description is the description of the area the debris was found in.
  - MaterialID is the ID code of the material that the debris was composed of.
  - Material Description is the description given to the material that composes the debris.
  - Time is the time that the observation was made.
  - There were a number of problems with the dataset namely;
    - There were a number of cases of missing data in the dataset.
    - data anomalies (lat/long values don't match named regions)
    -

3.2 Dataset Pre-processing

Everything below is from Stuart's RNW file

Logged marine debris is available for download *here*. I'm importing data from 2010 till Feb 19th 2020. There doesn't seem to be data before 2010. The data is reported marine debris. DataImport

I'm going to replace the column for time as a date data type, renaming it as simply "Time": DateParsing

Wrangling A quick look at the data:

```
data

## # A tibble: 363,368 x 15
##    ListID ListName ItemID ItemName LogID Latitude Longitude Altitude Quantity
##     <int> <fct>     <int> <fct>    <int>    <dbl>     <dbl>    <dbl>    <dbl>
##  1     22 Marine ~    183 Lumber/~   322     28.0     -82.8     -8.8        1
##  2     22 Marine ~    183 Lumber/~   323     28.0     -82.8     -8.7        1
##  3     22 Marine ~    187 Cigaret~   324     28.0     -82.8      0.2        2
##  4     22 Marine ~    181 Bottle ~   325     28.0     -82.8      0.4        1
##  5     22 Marine ~    181 Bottle ~   326     28.0     -82.8      0.4        1
##  6     22 Marine ~    181 Bottle ~   327     28.0     -82.8      1.4        1
```

```
## 7      22 Marine ~    174 Aerosol~   328     28.0     -82.8      1.9        1
## 8      22 Marine ~    207 Straws     329     28.0     -82.8      2.6        1
## 9      22 Marine ~    185 Disposa~   330     28.0     -82.8      2          1
## 10     22 Marine ~    202 Plastic~   331     28.0     -82.8      3          1
## # ... with 363,358 more rows, and 6 more variables: `Error Radius` <dbl>,
## #   Location <fct>, Description <chr>, `Material ID` <int>, `Material
## #   Description` <fct>, Time <dttm>
```

Let's first check for missing values: MissingValues

```
## [1] "Location"    "Description"
```

So only these columns contain missing values. We will use an explicit missing value for the location factor:

Lets see the amount of unique values for each column: UniqueValueCount

```
##               ListID            ListName              ItemID
##                    1                   1                  55
##             ItemName               LogID            Latitude
##                   55              363368              142707
##            Longitude            Altitude            Quantity
##               136490              135214                 496
##         Error Radius            Location         Description
##                18374                1458                8494
##          Material ID Material Description                Time
##                    8                   8              248436
```

Both "ListID" and "ListName" don't give us any information, so we will remove them both.

Lets see if there are any "ItemNames" associated with more than one "Material Descriptions".

```
## # A tibble: 1 x 2
##   ItemName          n
##   <chr>         <int>
## 1 Rubber Gloves     2
```

So rubber gloves are associated with two material descriptions, but otherwise a one to many relationship exists between "Material Description" and "ItemName".

```
## # A tibble: 2 x 2
##   `Material Description` Quantity
##   <fct>                     <dbl>
## 1 PLASTIC                    2114
## 2 RUBBER                      155
```

It seems that most rubber gloves are classified as plastic rather than rubber. I'm going to search for any extra descriptions given in the observations to try and gain some insight.

```
## # A tibble: 33 x 3
##    `Material Description` ItemName     Description
##    <fct>                  <fct>        <chr>
##  1 PLASTIC                Rubber Glov~ Found on wassaw island Oct. 21 with beac~
##  2 PLASTIC                Rubber Glov~ undefined
```

```
##  3 PLASTIC                Rubber Glov~ undefined
##  4 PLASTIC                Rubber Glov~ thermal
##  5 PLASTIC                Rubber Glov~ Near water
##  6 PLASTIC                Rubber Glov~ Taste of Omaha Cleanup
##  7 PLASTIC                Rubber Glov~ Taste of Omaha Cleanup
##  8 PLASTIC                Rubber Glov~ 2 diff kinds
##  9 PLASTIC                Rubber Glov~ undefined
## 10 PLASTIC                Rubber Glov~ Latex
## # ... with 23 more rows
```

All instances of rubber gloves with non-missing descriptions are categorised as plastic. We also see that the descriptions suggest that the categorisation may be innaccurate: the last two instances here have "Balloon" in the extra descriptions... why aren't they categorised as such?

3.3 Distribution of observed debris:

MaterialQuantities

```
## Error:  <text>:4:0:  unexpected end of input
## 2:  group_by(‘Material Description‘) %>%
## 3:  summarise(Quantity = sum(Quantity)) %>%
##  ^
```

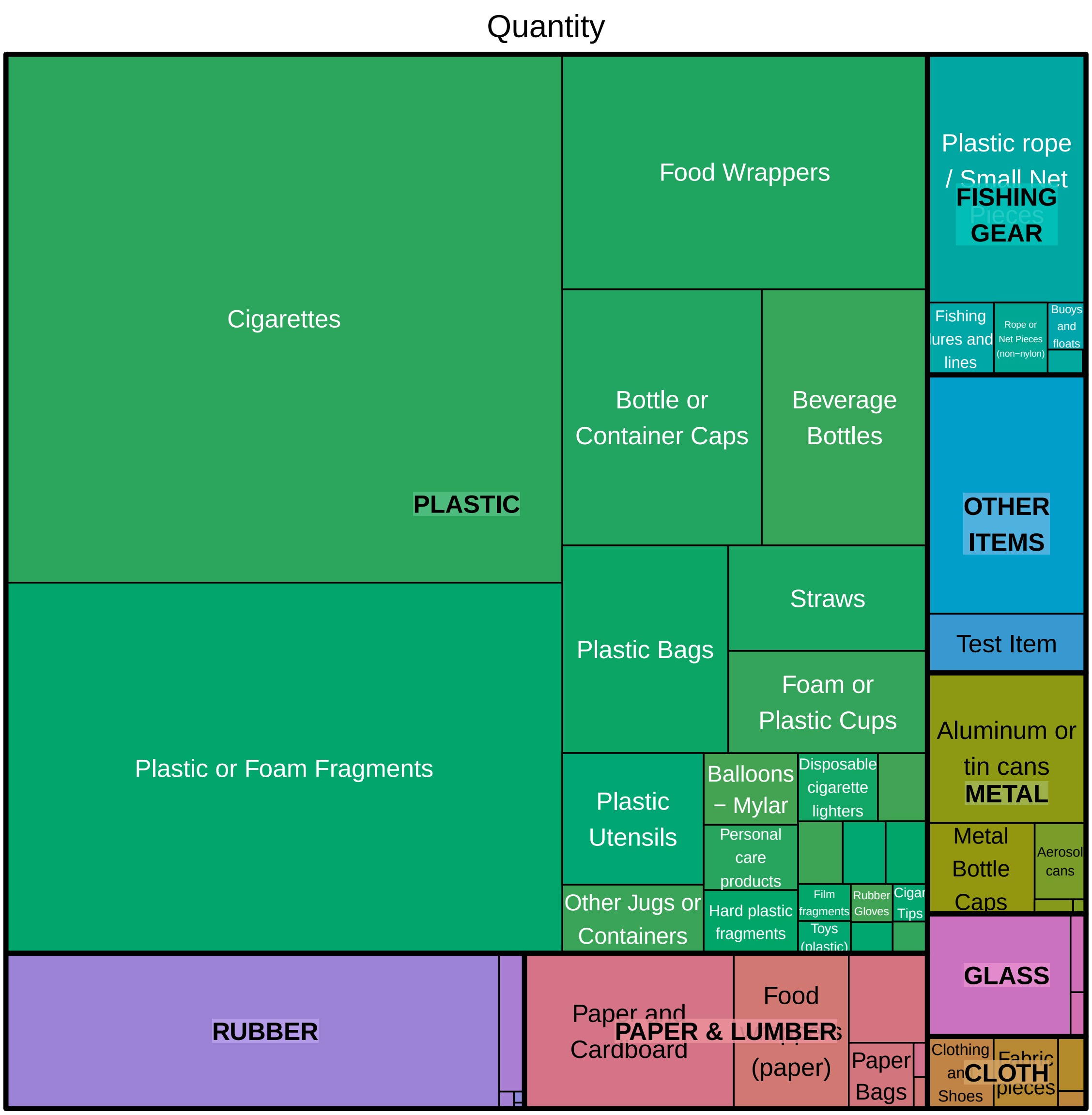
```
## Error:  ‘data‘ must be a data frame, or other object coercible by ‘fortify()‘, not an
S3 object with class uneval## Did you accidentally pass ‘aes()‘ to the ‘data‘ argument?
```

Figure 1: Material Quantities

So the most populated material class is Plastic. Note that this does not necessarily mean that plastic is the largest quantity of debris, just that the individual number of items categorised is largest.

A tree map of material quantities:

```
## Warning in png("plots/treemap.png"):  unable to open file 'plots/treemap.png' for writing
## Warning in png("plots/treemap.png"):  opening device failed
## Error in png("plots/treemap.png"):  unable to start png() device
```




```
## $tm
##    Material Description                       ItemName  vSize vColor
## 1             CLOTH                 Clothing and Shoes   6336      1
## 2             CLOTH                     Fabric pieces   6172      1
## 3             CLOTH                Gloves (non-rubber)    653      1
## 4             CLOTH                               <NA>  15168      4
## 5             CLOTH                    Towels or rags   2007      1
## 6      FISHING GEAR                  Buoys and floats   2397      1
## 7      FISHING GEAR        Crab/Lobster/Fish trap parts   1185      1
## 8      FISHING GEAR           Fishing lures and lines   6391      1
## 9      FISHING GEAR                      Fishing nets    104      1
## 10     FISHING GEAR                               <NA>  67442      6
## 11     FISHING GEAR    Plastic rope / Small Net Pieces  52221      1
## 12     FISHING GEAR     Rope or Net Pieces (non-nylon)   5144      1
## 13            GLASS                       Glass Bottle  23298      1
## 14            GLASS                    Glass fragments    907      1
```

```
## 15                GLASS                             Glass Jars    1603      1
## 16                GLASS                                   <NA>   25808      3
## 17                METAL                           Aerosol cans    5201      1
## 18                METAL                   Aluminum or tin cans   31606      1
## 19                METAL    Batteries (acidic and alkaline)     248      1
## 20                METAL                     Metal Bottle Caps   12901      1
## 21                METAL                       Metal fragments     738      1
## 22                METAL                                  <NA>   50694      5
## 23          OTHER ITEMS                                  <NA>   62649      2
## 24          OTHER ITEMS                                 Other   50223      1
## 25          OTHER ITEMS                             Test Item   12426      1
## 26       PAPER & LUMBER                 Food wrappers (paper)   23690      1
## 27       PAPER & LUMBER            Lumber/Building Materials    9335      1
## 28       PAPER & LUMBER                                  <NA>   83006      6
## 29       PAPER & LUMBER                               Pallets     579      1
## 30       PAPER & LUMBER                  Paper and Cardboard   43116      1
## 31       PAPER & LUMBER                            Paper Bags    5659      1
## 32       PAPER & LUMBER                            Paper Cups     627      1
## 33              PLASTIC                      Balloons - Mylar    8993      1
## 34              PLASTIC                      Beverage Bottles   56296      1
## 35              PLASTIC             Bottle or Container Caps   67807      1
## 36              PLASTIC  Chemicals and chemical containers      40      1
## 37              PLASTIC                            Cigar Tips    1734      1
## 38              PLASTIC      Cigarette or tobacco packaging    3728      1
## 39              PLASTIC                            Cigarettes  389889      1
## 40              PLASTIC       Disposable cigarette lighters    7234      1
## 41              PLASTIC                        Film fragments    2591      1
## 42              PLASTIC                             Fireworks    3565      1
## 43              PLASTIC                        Foam fragments    4481      1
## 44              PLASTIC                  Foam or Plastic Cups   27031      1
## 45              PLASTIC                        Food Wrappers  113817      1
## 46              PLASTIC               Hard plastic fragments    7935      1
## 47              PLASTIC                                  <NA> 1098985     25
## 48              PLASTIC Non-food related plastic packaging    1737      1
## 49              PLASTIC             Other Jugs or Containers   12904      1
## 50              PLASTIC               Personal care products    8154      1
## 51              PLASTIC                         Plastic Bags   45711      1
## 52              PLASTIC            Plastic or Foam Fragments  273536      1
## 53              PLASTIC                     Plastic Utensils   24673      1
## 54              PLASTIC                        Rubber Gloves    2114      1
## 55              PLASTIC                       Six-pack rings    1420      1
## 56              PLASTIC                               Straws   27857      1
## 57              PLASTIC                  Styrofoam packaging    3474      1
## 58              PLASTIC                      Toys (plastic)    2264      1
## 59               RUBBER                           Flip-flops    4661      1
## 60               RUBBER                       Latex balloons      84      1
## 61               RUBBER                                  <NA>  106823      5
## 62               RUBBER                     Rubber fragments     335      1
## 63               RUBBER                        Rubber Gloves     155      1
## 64               RUBBER                                Tires  101588      1
##    stdErr vColorValue level        x0          y0            w           h
## 1     6336          NA     2 0.8531943 0.000000000 0.0613238944 0.068397960
## 2     6172          NA     2 0.9145182 0.000000000 0.0597365966 0.068397960
```

```
## 3      653            NA      2 0.9742548 0.000000000 0.0257451956 0.016790928
## 4    15168            NA      1 0.8531943 0.000000000 0.1468056866 0.068397960
## 5     2007            NA      2 0.9742548 0.016790928 0.0257451956 0.051607032
## 6     2397            NA      2 0.9644487 0.719882250 0.0355512621 0.044634504
## 7     1185            NA      2 0.9644487 0.695879799 0.0326828903 0.024002452
## 8     6391            NA      2 0.8531943 0.695879799 0.0616408346 0.068636956
## 9      104            NA      2 0.9971316 0.695879799 0.0028683718 0.024002452
## 10   67442            NA      1 0.8531943 0.695879799 0.1468056866 0.304120201
## 11   52221            NA      2 0.8531943 0.764516755 0.1468056866 0.235483245
## 12    5144            NA      2 0.9148351 0.695879799 0.0496135899 0.068636956
## 13   23298            NA      2 0.8531943 0.068397960 0.1325278552 0.116377542
## 14     907            NA      2 0.9857222 0.068397960 0.0142778314 0.042053558
## 15    1603            NA      2 0.9857222 0.110451518 0.0142778314 0.074323984
## 16   25808            NA      1 0.8531943 0.068397960 0.1468056866 0.116377542
## 17    5201            NA      2 0.9524158 0.198492909 0.0475841776 0.072357240
## 18   31606            NA      2 0.8531943 0.270850149 0.1468056866 0.142522806
## 19     248            NA      2 0.9880316 0.184775502 0.0119684341 0.013717408
## 20   12901            NA      2 0.8531943 0.184775502 0.0992215089 0.086074648
## 21     738            NA      2 0.9524158 0.184775502 0.0356157435 0.013717408
## 22   50694            NA      1 0.8531943 0.184775502 0.1468056866 0.228597454
## 23   62649            NA      1 0.8531943 0.413372956 0.1468056866 0.282506843
## 24   50223            NA      2 0.8531943 0.469406253 0.1468056866 0.226473546
## 25   12426            NA      2 0.8531943 0.413372956 0.1468056866 0.056033297
## 26   23690            NA      2 0.6739071 0.000000000 0.1064756875 0.147289679
## 27    9335            NA      2 0.7803827 0.062416274 0.0728115719 0.084873404
## 28   83006            NA      1 0.4801204 0.000000000 0.3730739096 0.147289679
## 29     579            NA      2 0.8404032 0.000000000 0.0127910788 0.029966022
## 30   43116            NA      2 0.4801204 0.000000000 0.1937866502 0.147289679
## 31    5659            NA      2 0.7803827 0.000000000 0.0600204931 0.062416274
## 32     627            NA      2 0.8404032 0.029966022 0.0127910788 0.032450252
## 33    8993            NA      2 0.6459880 0.269153750 0.0873999284 0.068116327
## 34   56296            NA      2 0.6998035 0.534216090 0.1533908431 0.242960577
## 35   67807            NA      2 0.5150484 0.534216090 0.1847550962 0.242960577
## 36      40            NA      2 0.8523133 0.147289679 0.0008809984 0.030056788
## 37    1734            NA      2 0.8210379 0.177346466 0.0321564428 0.035697582
## 38    3728            NA      2 0.7333879 0.213044048 0.0414821483 0.059493891
## 39  389889            NA      2 0.0000000 0.498869692 0.5150483741 0.501130308
## 40    7234            NA      2 0.7333879 0.272537939 0.0739803294 0.064732138
## 41    2591            NA      2 0.7333879 0.177952479 0.0488790011 0.035091570
## 42    3565            NA      2 0.7748701 0.213044048 0.0396684170 0.059493891
## 43    4481            NA      2 0.8073682 0.272537939 0.0458260791 0.064732138
## 44   27031            NA      2 0.6686979 0.337270077 0.1844964097 0.096991104
## 45  113817            NA      2 0.5150484 0.777176668 0.3381459393 0.222823332
## 46    7935            NA      2 0.6459880 0.147289679 0.0873999284 0.060102642
## 47 1098985            NA      1 0.0000000 0.147289679 0.8531943134 0.852710321
## 48    1737            NA      2 0.7822669 0.147289679 0.0387709646 0.029658619
## 49   12904            NA      2 0.5150484 0.147289679 0.1309396025 0.065239563
## 50    8154            NA      2 0.6459880 0.207392320 0.0873999284 0.061761429
## 51   45711            NA      2 0.5150484 0.337270077 0.1536495297 0.196946013
## 52  273536            NA      2 0.0000000 0.147289679 0.5150483741 0.351580013
## 53   24673            NA      2 0.5150484 0.212529242 0.1309396025 0.124740835
## 54    2114            NA      2 0.7822669 0.176948297 0.0387709646 0.036095751
## 55    1420            NA      2 0.8210379 0.147289679 0.0312754443 0.030056788
```

```
## 56   27857           NA     2 0.6686979 0.434261181 0.1844964097 0.099954910
## 57    3474           NA     2 0.8145385 0.213044048 0.0386558431 0.059493891
## 58    2264           NA     2 0.7333879 0.147289679 0.0488790011 0.030662800
## 59    4661           NA     2 0.4565915 0.016149814 0.0235289246 0.131139865
## 60      84           NA     2 0.4703235 0.000000000 0.0097968867 0.005676085
## 61  106823           NA     1 0.0000000 0.000000000 0.4801204039 0.147289679
## 62     335           NA     2 0.4565915 0.000000000 0.0137320379 0.016149814
## 63     155           NA     2 0.4703235 0.005676085 0.0097968867 0.010473729
## 64  101588           NA     2 0.0000000 0.000000000 0.4565914792 0.147289679
##      color
## 1  #C08446
## 2  #B78933
## 3  #BC863D
## 4  #D3A362
## 5  #B38B2B
## 6  #00A6AB
## 7  #00A89E
## 8  #00A7A7
## 9  #00A899
## 10 #00C1BA
## 11 #00A7A2
## 12 #00A894
## 13 #CA71BF
## 14 #D26FB0
## 15 #CE6FB8
## 16 #E68ECF
## 17 #7A9C28
## 18 #8D9813
## 19 #809B21
## 20 #93960F
## 21 #87991A
## 22 #A1B453
## 23 #5BB5E2
## 24 #009EC8
## 25 #3799CF
## 26 #D07871
## 27 #D47481
## 28 #EC929B
## 29 #D27777
## 30 #D57387
## 31 #D3757D
## 32 #D5728D
## 33 #44A352
## 34 #36A459
## 35 #22A560
## 36 #00A666
## 37 #00A66D
## 38 #3DA456
## 39 #2CA55D
## 40 #11A664
## 41 #00A66A
## 42 #00A770
## 43 #41A354
```

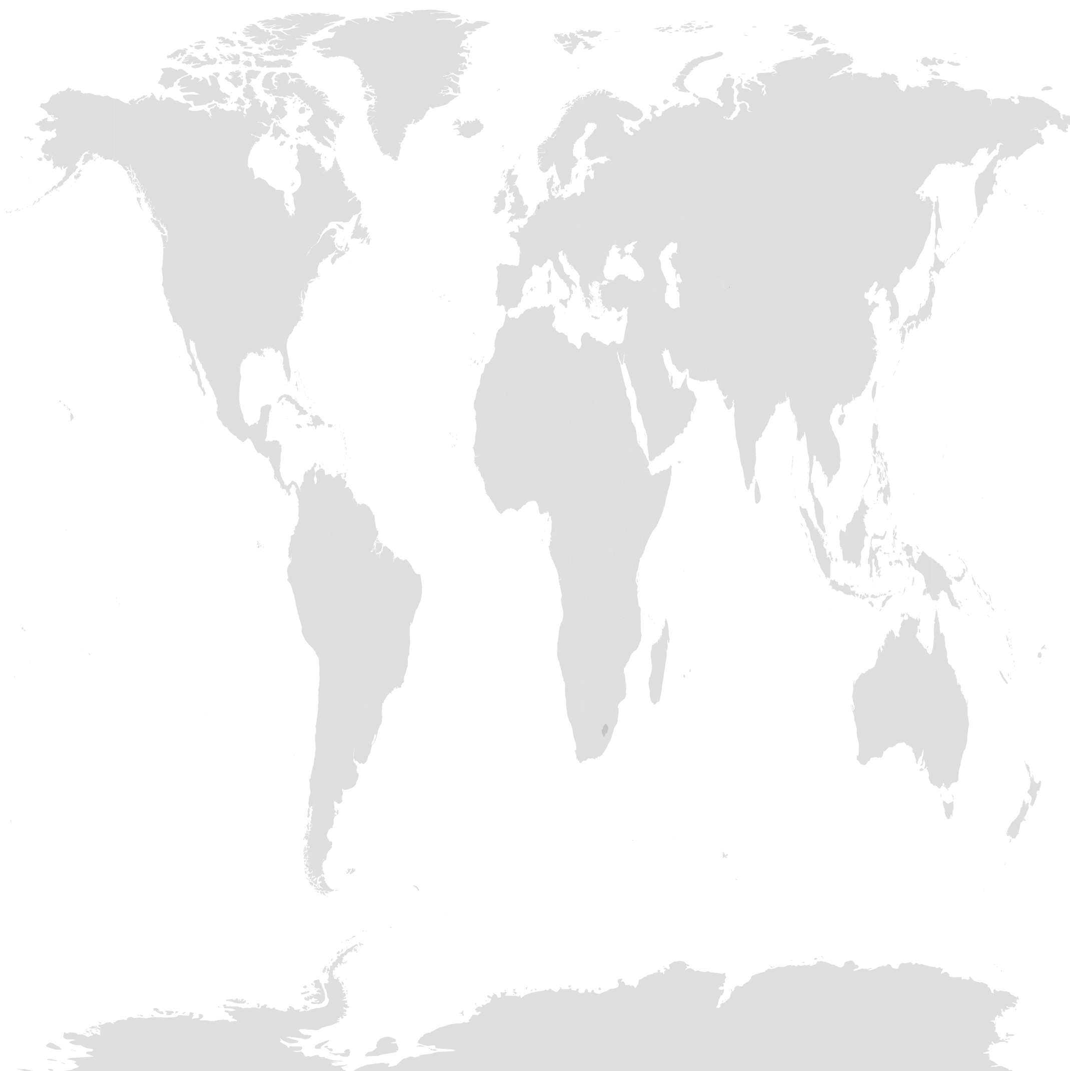
```
## 44 #34A45A
## 45 #20A561
## 46 #00A668
## 47 #53BF82
## 48 #00A76E
## 49 #39A458
## 50 #27A55E
## 51 #0BA665
## 52 #00A66B
## 53 #00A772
## 54 #40A454
## 55 #30A55B
## 56 #19A562
## 57 #00A669
## 58 #00A770
## 59 #AA7ED3
## 60 #9585D6
## 61 #B79FEB
## 62 #A480D4
## 63 #8D88D7
## 64 #9D83D5
##
## $type
## [1] "index"
##
## $vSize
## [1] "Quantity"
##
## $vColor
## [1] NA
##
## $stdErr
## [1] "Quantity"
##
## $algorithm
## [1] "pivotSize"
##
## $vpCoorX
## [1] 0.02812148 0.97187852
##
## $vpCoorY
## [1] 0.01406074 0.93593926
##
## $aspRatio
## [1] 1.023733
##
## $range
## [1] NA
##
## $mapping
## [1] NA NA NA
##
## $draw
```

```
## [1] TRUE
## null device
##           1
```

Cigarettes are the most common item recorded. Perhaps some of the debris is not actually from the sea, but rather from people littering by the coastline? Does debris littered on the coastline end up in the oceans?

We have locational data, so lets check for any geographical observation bias.

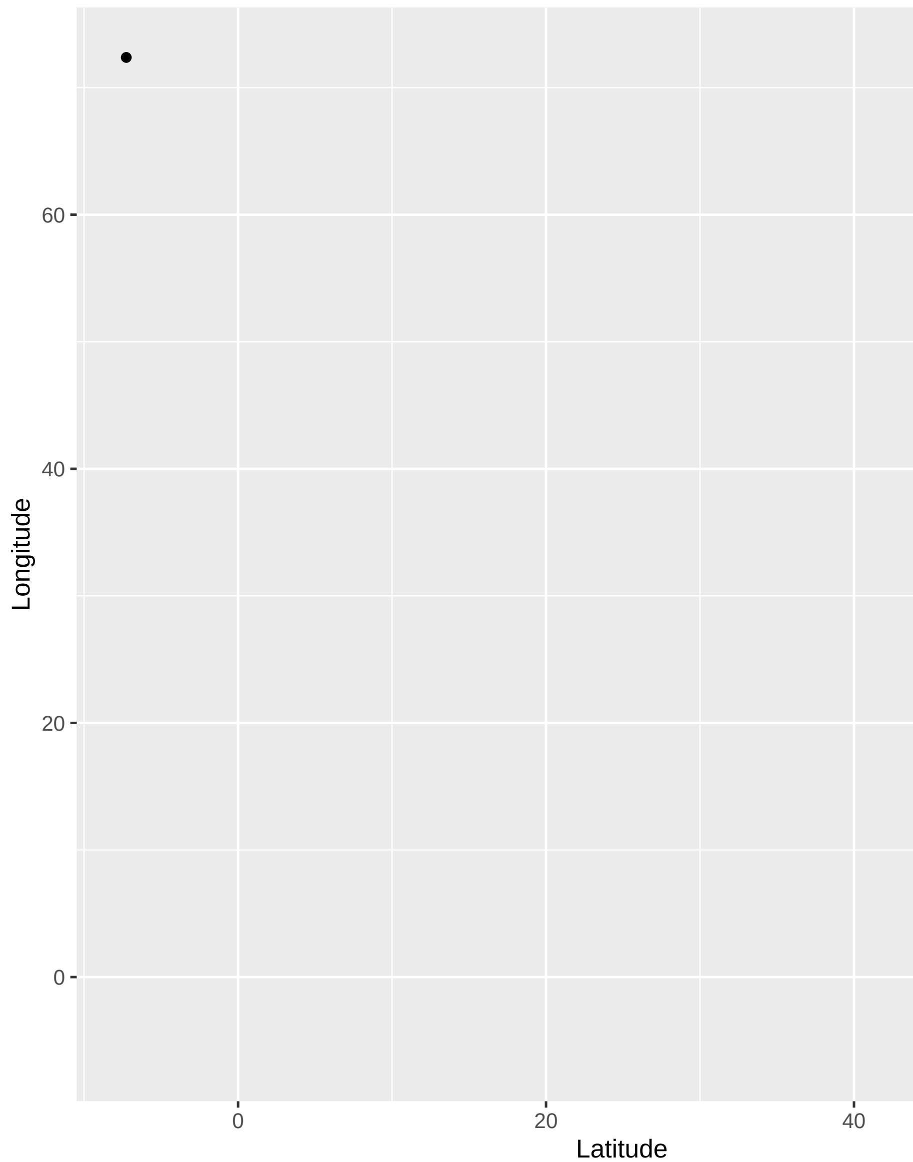
```
## Warning:  Computation failed in `stat_binhex()`:
## Package `hexbin` required for `stat_binhex`.
## Please install and try again.
```



There seems to be a strong bias towards North America in our dataset. We will try a logarithmic plot to see things more clearly:

```
## Warning:  Computation failed in `stat_binhex()`:
## Package `hexbin` required for `stat_binhex`.
## Please install and try again.
```

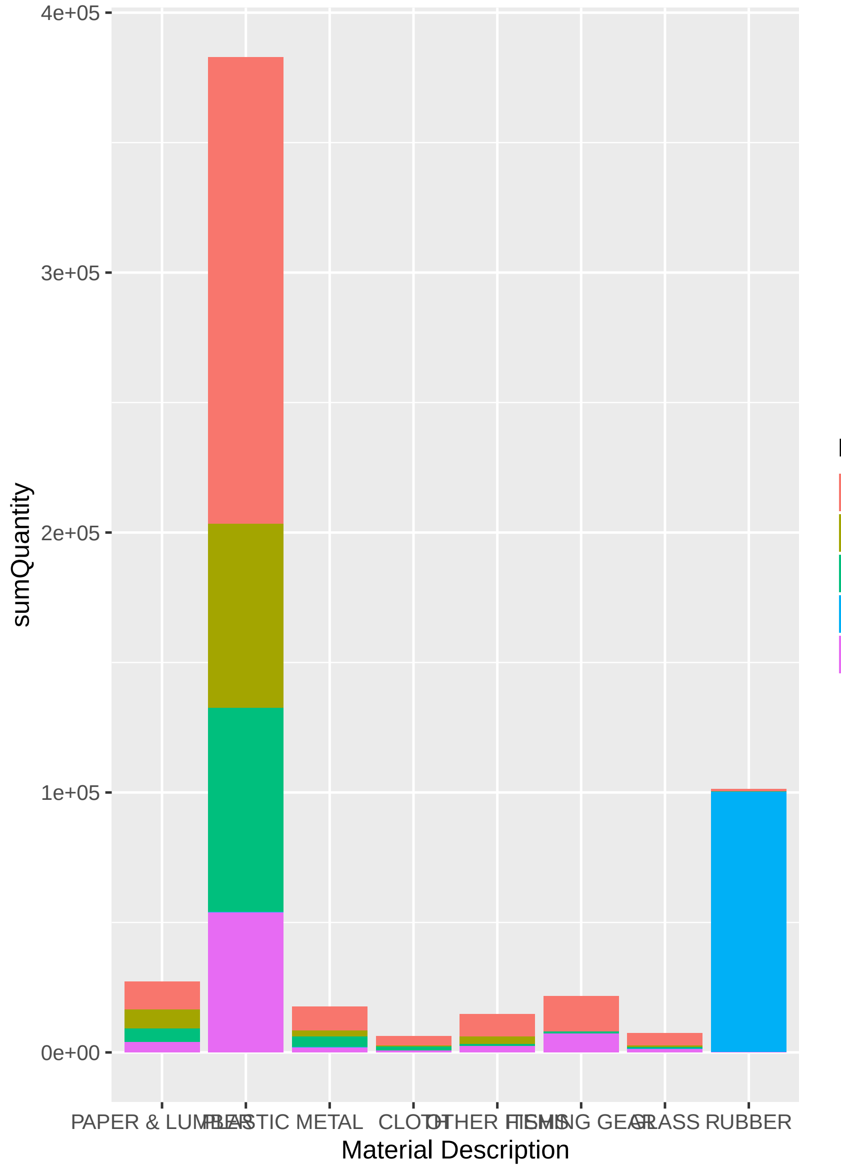
We need to know how reliable the location data is. I'm going to filter for "united kingdom" in the loca-



tion field and plot the raw coordinates.

We have a outliers here. Maybe a difference in standards used for Longitude and Latitude? Some systems put the Latitude origin close to the UK.

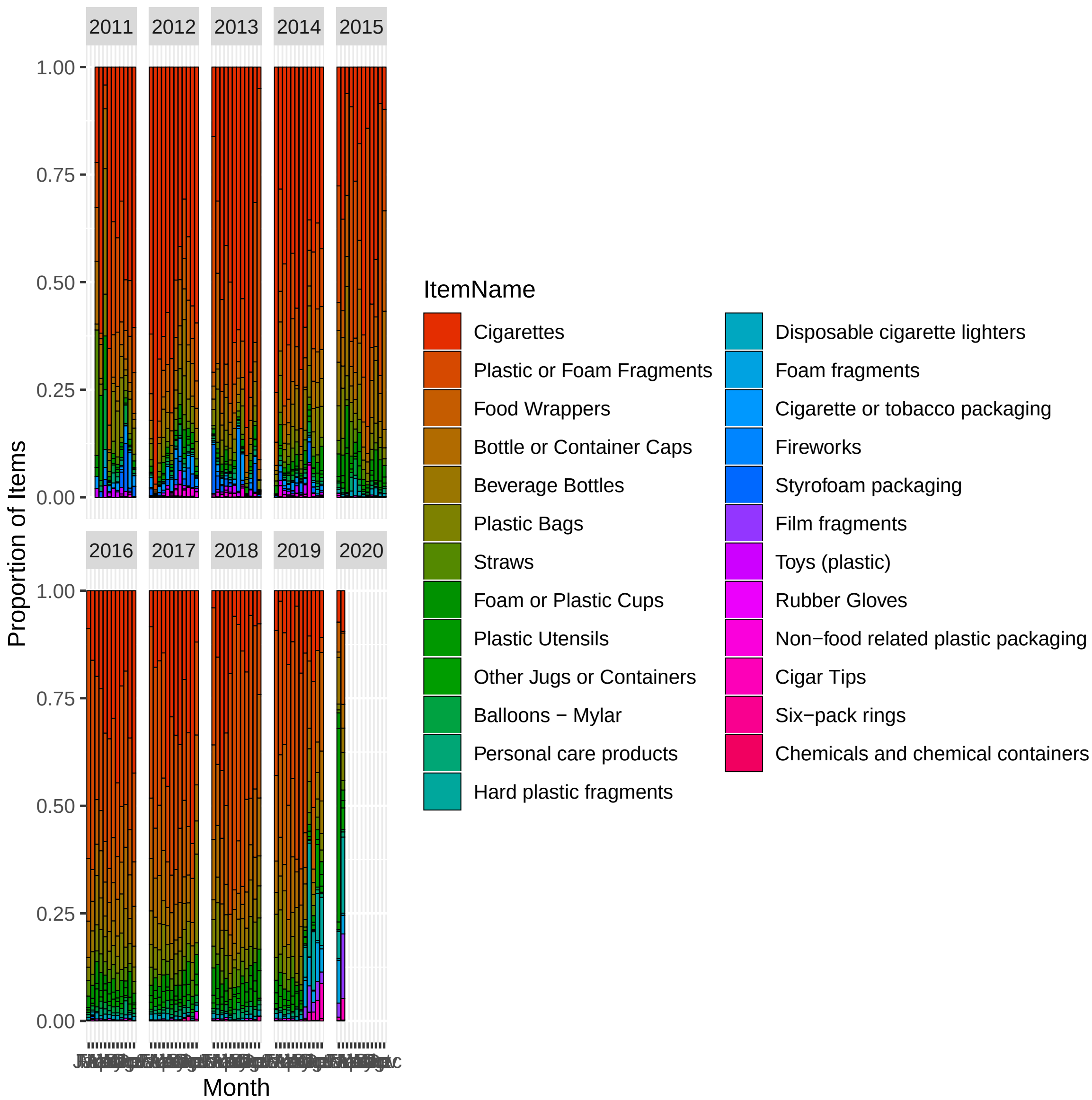
Questions Distribution of plastic by location. Are the distributions of plastic fairly constant for the loca-



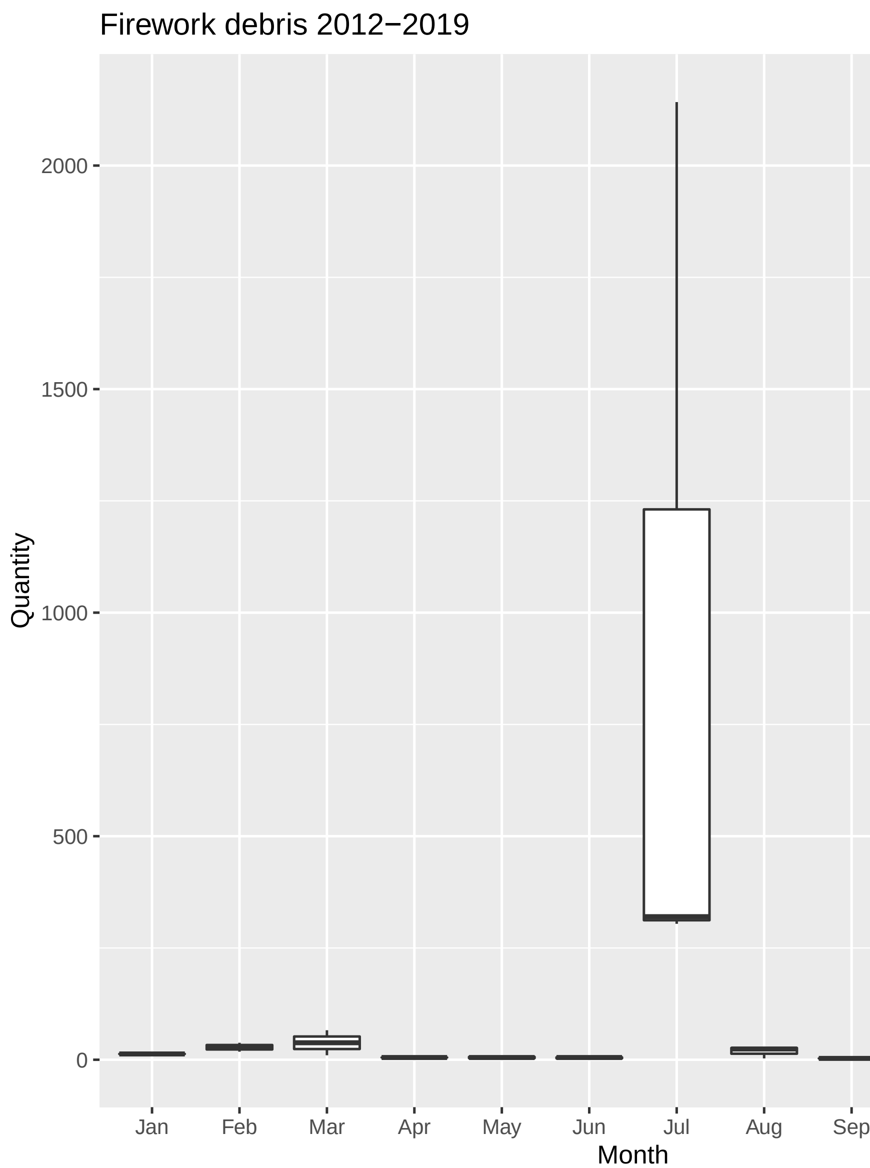
tions with the most observations? Let's look:

We see that the Location "unknown" has the most plastic... note that this is distinct from "(Missing)", which was our original NA values. Maybe we should merge these.

Question: Are observed plastic item proportions time invariant?

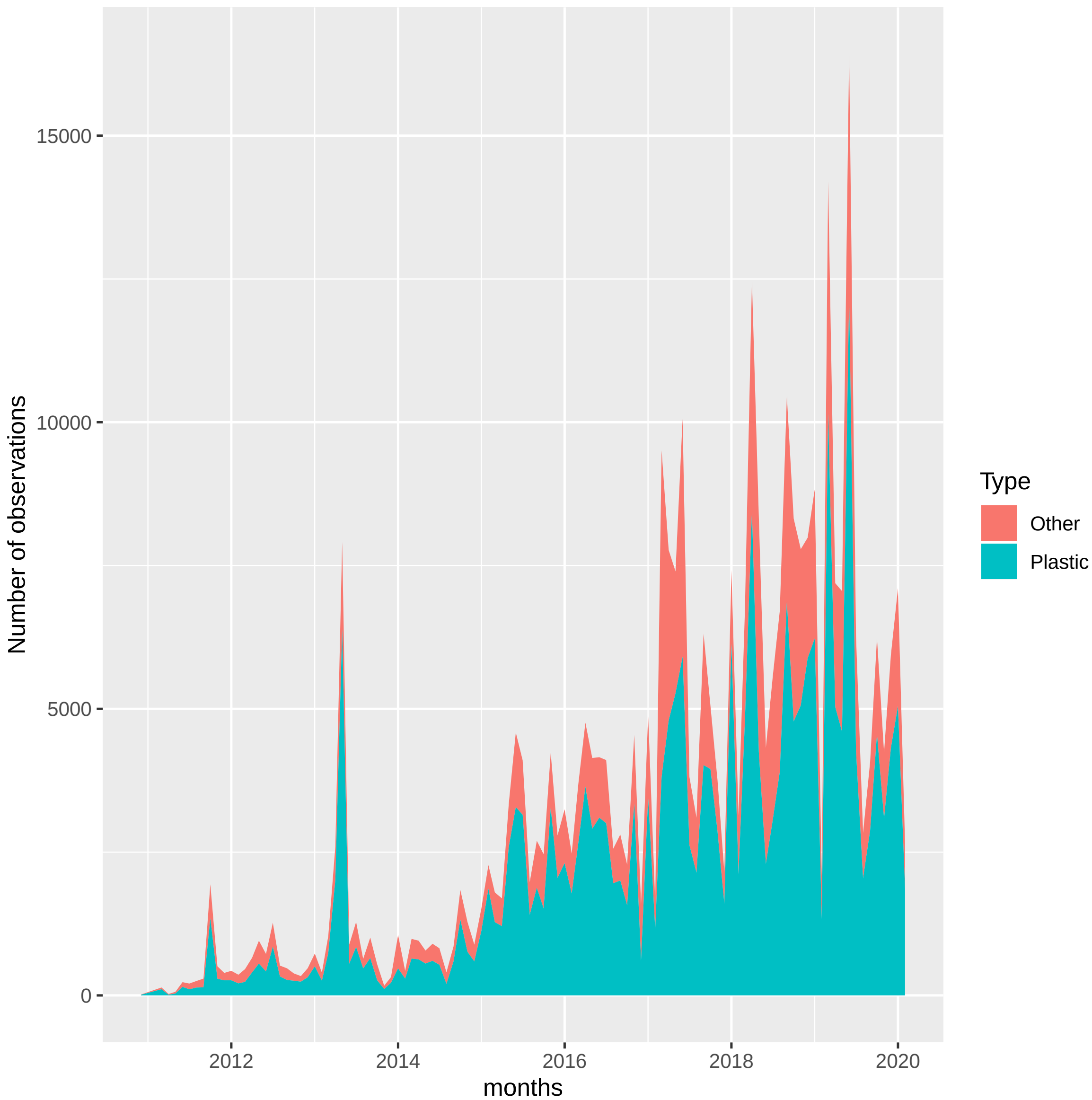


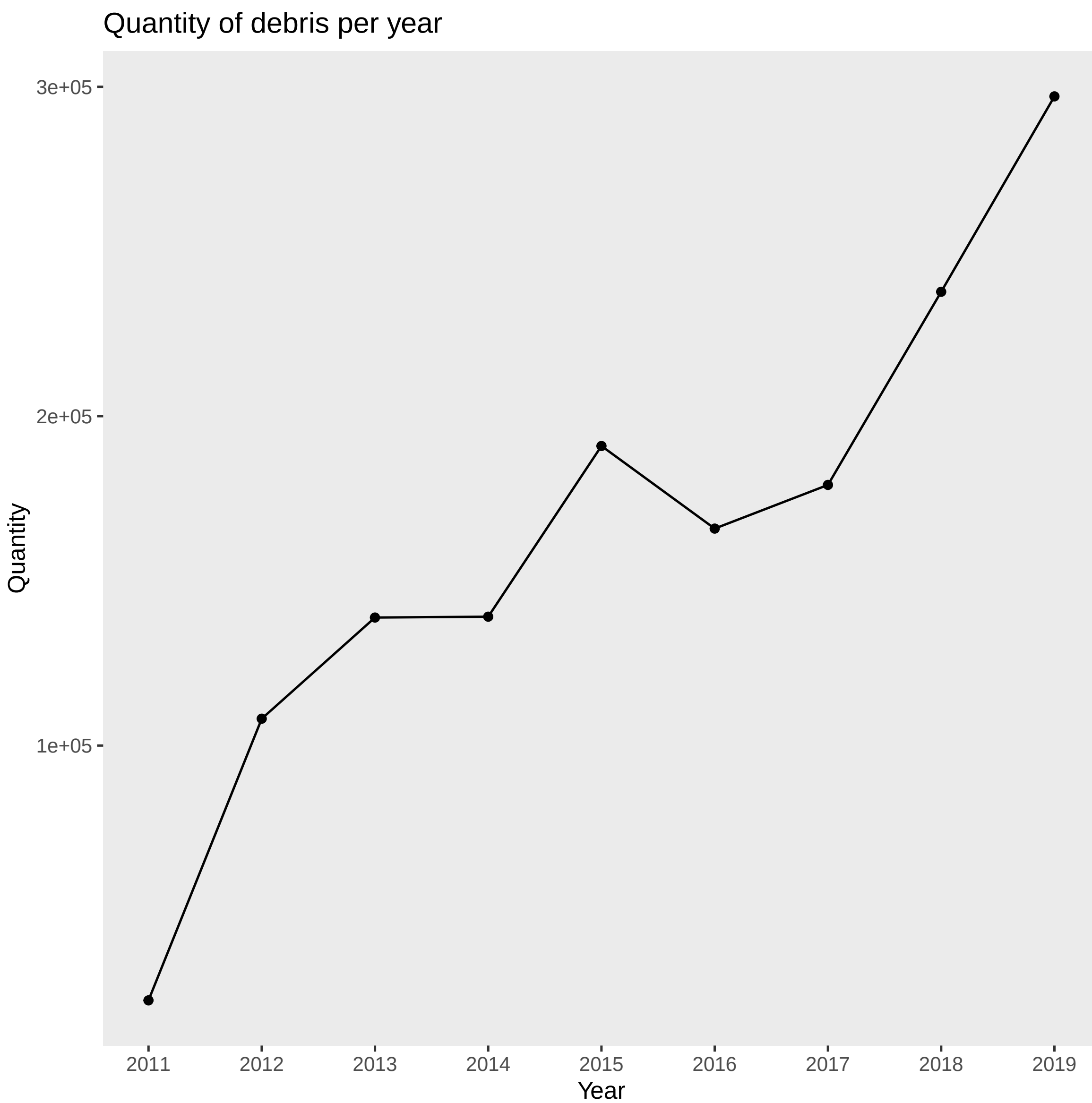

```
## Warning in grDevices::png(..., res = dpi, units = "in"):  unable to open file 'plots/pastic_debr
for writing
## Warning in grDevices::png(..., res = dpi, units = "in"):  opening device failed
## Error in grDevices::png(..., res = dpi, units = "in"):  unable to start png() device
```



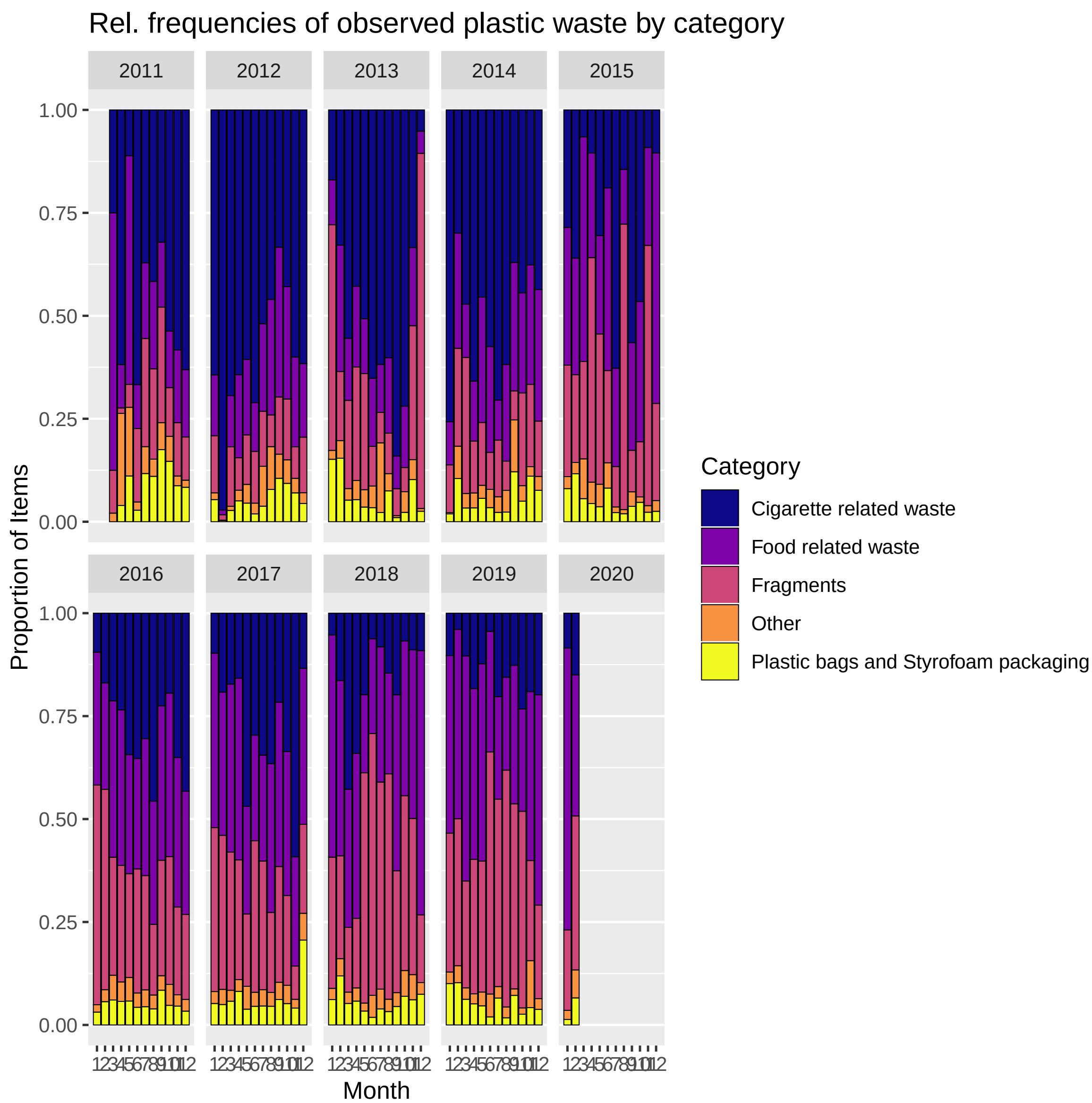
4th July and Firework link? (Karen's Idea)

```
## Saving 7 x 7 in image
## Warning in grDevices::png(..., res = dpi, units = "in"):  unable to open file 'plots/fireworks.p
for writing
## Warning in grDevices::png(..., res = dpi, units = "in"):  opening device failed
## Error in grDevices::png(..., res = dpi, units = "in"):  unable to start png() device
```


15000
10000
5000
0
Number of observations
2012
2014
2016
2018
2020
months
Type
Other
Plastic




```
## Saving 7 x 7 in image
## Warning in grDevices::png(..., res = dpi, units = "in"):  unable to open file 'plots/observation
for writing
## Warning in grDevices::png(..., res = dpi, units = "in"):  opening device failed
## Error in grDevices::png(..., res = dpi, units = "in"):  unable to start png() device
```



Recategorisation

After the issues with the dataset that were identified in the section above, it was decided that it would be best to transform the dataset in the following ways:

- reclassified some labels because variation was too high (there were too many labels)
- The values of the missing data were removed.
- It was decided that subsets that were not needed were removed while retaining the necessary subsets.

4 Exploration

Here we describe the things we found...

4.1 Proportion Trends

How pollutant proportions change over time.

Cigarette butts proportions and raw counts decrease over time: possibly less people smoking, or moving to vaping

General pollution count going down over time?

Old pollutants fall away (cigarette butts) but new ones are introduced

4.2 Event-Driven Pollution

Fireworks found in July and North-America only: possibly 4th July celebrations

4.3 Location-Driven Pollution

Rubber found in Indoneasia only: possibly a recording bias.

Certain classes are found in certain regions only: not because they don't exist elsewhere but because of recording bias focus in those areas

4.4 Item Pairing

(e.g. are 6-pack beer rings observed at the same time as fireworks?)

5 Predictive Modelling

The authors of this report built a model to predict the proportion of plastics given Month and Location. This would give more accurate predictions as opposed to a simple linear model, given we know that event-driven pollution will determine different pollutants are different times.

5.1 Description of Model

Georgios' script

```
plasticN <- plastic %>%
  mutate(year = as.integer(year(Time))) %>%
  filter(year > 2010) %>%
  group_by(year, category) %>%
  summarise(`Total Quantity` = sum(Quantity))

  ####
library(dplyr)
df11N <- plasticN  %>%
  filter(year == 2011) %>%
  group_by(year) %>%
  mutate(freq = `Total Quantity` / sum(`Total Quantity`))

df12N <- plasticN  %>%
  filter(year == 2012) %>%
  group_by(year) %>%
  mutate(freq = `Total Quantity` / sum(`Total Quantity`))


df13N <- plasticN  %>%
```

```r
  filter(year == 2013) %>%
  group_by(year) %>%
  mutate(freq = `Total Quantity` / sum(`Total Quantity`))


df14N <- plasticN  %>%
  filter(year == 2014) %>%
  group_by(year) %>%
  mutate(freq = `Total Quantity` / sum(`Total Quantity`))

df15N <- plasticN  %>%
  filter(year == 2015) %>%
  group_by(year) %>%
  mutate(freq = `Total Quantity` / sum(`Total Quantity`))


df16N <- plasticN  %>%
  filter(year == 2016) %>%
  group_by(year) %>%
  mutate(freq = `Total Quantity` / sum(`Total Quantity`))



df17N <- plasticN  %>%
  filter(year == 2017) %>%
  group_by(year) %>%
  mutate(freq = `Total Quantity` / sum(`Total Quantity`))


df18N <- plasticN  %>%
  filter(year == 2018) %>%
  group_by(year) %>%
  mutate(freq = `Total Quantity` / sum(`Total Quantity`))

df19N <- plasticN  %>%
  filter(year == 2019) %>%
  group_by(year) %>%
  mutate(freq = `Total Quantity` / sum(`Total Quantity`))


dfTotN <- rbind(df11N, df12N, df13N, df14N, df15N, df16N, df17N, df18N, df19N)

# plot for observing the data
(time_plotfr2N <- ggplot(dfTotN, aes(x = year, y = freq, color=category, fill = category)) +
  geom_smooth(method="lm") +
  geom_point(size=3) +
  theme_bw() +
  xlab("Years") +
  ylab("freq") +
  ggtitle("portion of plastic") +
  expand_limits(y=0) +
  scale_y_continuous() +
  scale_x_continuous()+
```

```
  theme(legend.position="bottom")+
  theme(legend.text = element_text(size=5, face="bold")))

  ### MODELING with new categorisation


# create train and test set
n <- nrow(dfTotN)  # Number of observations
ntrain <- round(n*0.75)  # 75% for training set
set.seed(314)    # Set seed for reproducible results
tindex <- sample(n, ntrain)   # Create a random index
train_dfTotN <- dfTotN[tindex,]   # Create training set
test_dfTotN <- dfTotN[-tindex,]

#  Pr(>|t|) is the p-value, defined as the probability of observing any value equal or larger than


# linear model on train set
print("train model")
set.seed(1234)
dfTot_train.modelN <- lm(freq ~ year, data = train_dfTotN)
summary(dfTot_train.modelN)

# plotting frequencies according to train data
ggplot(data = train_dfTotN, aes(x = year, y = freq)) +
geom_point() +
stat_smooth(method = "lm", col = "dodgerblue3") +
theme(panel.background = element_rect(fill = "white"),
axis.line.x=element_line(),
axis.line.y=element_line()) +
ggtitle("Linear Model Fitted to Data")


print("PREDICTION")
predN <- predict(dfTot_train.modelN, test_dfTotN)
summary(predN)

# make actuals_predicteds dataframe
actuals_preds <- data.frame(cbind(actuals=test_dfTotN$freq, predicteds=predN))
head(actuals_preds)
# A simple correlation between the actuals and predicted values can be used as a form of accuracy m

correlation_accuracy <- cor(actuals_preds)  # 5.31%
min_max_accuracy <- mean(apply(actuals_preds, 1, min) / apply(actuals_preds, 1, max))
# => 53.73%, min_max accuracy
mape <- mean(abs((actuals_preds$predicteds - actuals_preds$actuals))/actuals_preds$actuals)
# => 99.4%, mean absolute percentage deviation
# Intrestingly enough min_max accuracy and mostly mean absolute percentage deviation score quite we
# but still on a model that can not be trusted.
```

5.2 Model Evaluation

5.3 Model Results

Time does not impact plastic composition.

6 Discussion

7 Conclusion and Future Work

Our hypothesis stands/does not stand.

8 Project Management

8.1 Facilities

Group 2 communicated using a dedicated Slack Channel, Github repository and weekly 1 hour meetings before the wednesday lab. All project documents used and the final report can be accessed from the *Public Github Repository*

8.2 Project Progress

Table 1: Record of Team Meetings

No	Date	Topic	Alex	Georgios	Karen	Roshi	Stuart
1.00	2020-02-05	Group Formation: set up communication channel in Slack and GitHub repository	yes	yes	yes	yes	yes
2.00	2020-02-11	Agreed topic of ”Plastic Pollution”, distributed research activity for week	yes	yes	yes	yes	yes
3.00	2020-02-18	Presented inividuals’ research findings and discussed hypothesis	yes	yes	yes	yes	yes
4.00	2020-02-25	Decided on final dataset to use and hypothesis of ”proportion of marine plastics pollution does not change over time”	yes	yes	yes	yes	yes
5.00	2020-03-04	Presentation draft agreed	yes	yes	yes	yes	yes
6.00	2020-03-10	Distributed section writing activity for week	yes	yes	yes	yes	yes
7.00	2020-03-17						
8.00	2020-03-24						
9.00	2020-03-31						
10.00	2020-04-07						
11.00	2020-04-14						
12.00	2020-04-21						

8.3 Peer-assessment

8.4 Section on figure referencing - keep for referencing

In this project iris was used, the dataset is made of 150 rows and four features.

Table 2: Peer Assessment out of 100

Peer.Review	Alex	Georgios	Karen	Roshi	Stuart
Alex	100	100	100	100	100
Georgios	100	100	100	100	100
Karen	100	100	100	100	100
Roshi	100	100	100	100	100
Stuart	100	100	100	100	100

Notice how we generate graphics within the sweave document. Check the following code, we will create a function that either finds x^2 or x^3 subject to parameters passed in the function

```
# create a vector of doubles
myNumbers <- seq(from=-1,to=1,by=.1)

# function definition
toPower <- function (x,p=2) {
    if (p==2)
        return (x*x)
    else if (p==3)
        return (x*x*x)
    return (x*x)
}

# call function
squared <- toPower(myNumbers)
cubes <- toPower(myNumbers,3)
```

An easy way to check that our function is doing the right calculation is to plot the results. The code below will generate a figure similar to Figure **??**:

```
plot(myNumbers,cubes,type='b',xlab = 'x', ylab = 'x*x',frame=FALSE,col='blue')
```

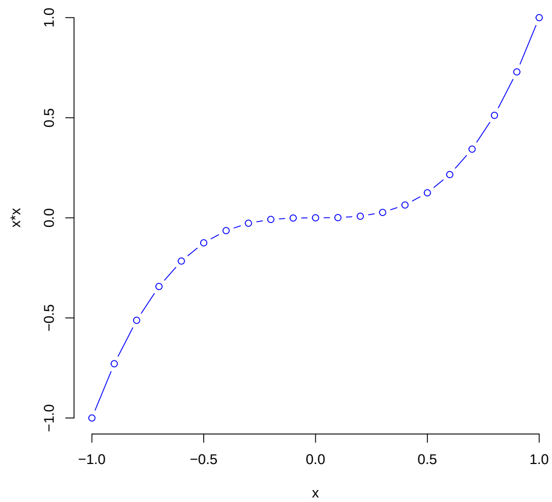


Figure 2: Simple Plot of $f(x) = x^3$ Function

8.5 Experiments

Now we can show how the function $f(x) = x^2$ looks like (Figure **??**)

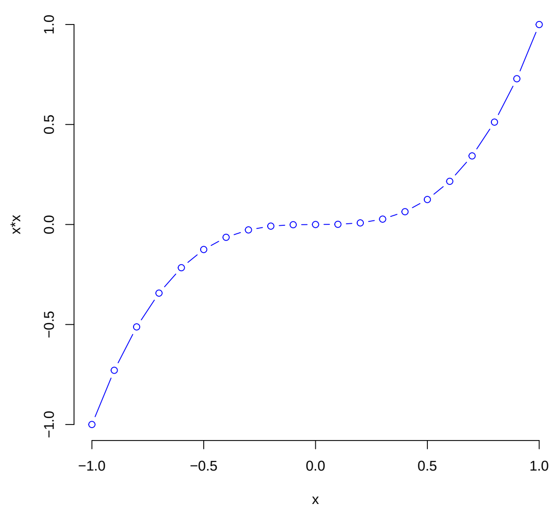


Figure 3: Simple Plot of $f(x) = x^3$ Function

References