MA20277 - Coursework 2

24224

Question 1

The novel "Frankenstein" by Mary Shelley tells the story of a young scientist who creates a creature via an experiment and is subsequently horrified by what he has made. Perform an analysis that addresses the following questions:

```
Frankenstein = readLines("Frankenstein.csv", encoding = "UTF-8")
```

a) Which five words, apart from those on the stop list considered in the lectures, appear the most often, and what is their term frequency?

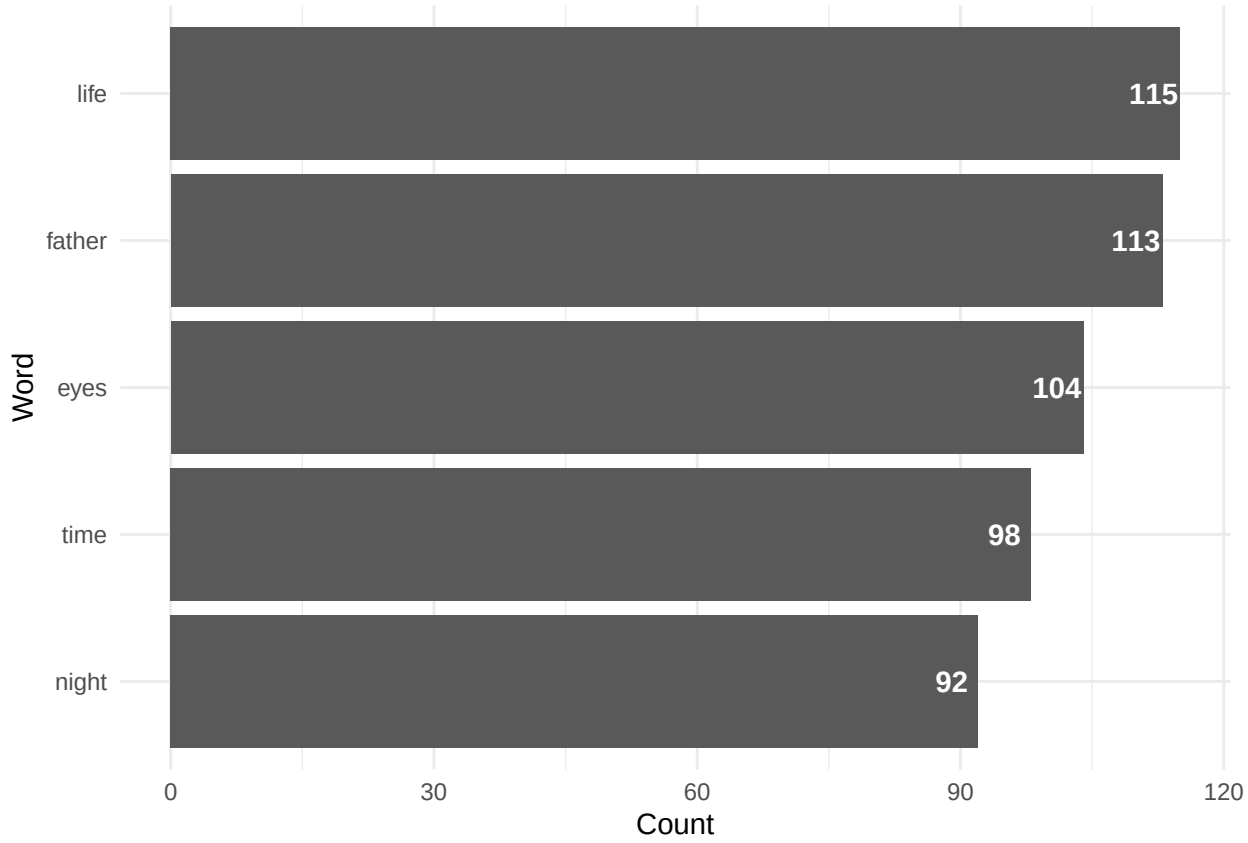
We first clean our dataset so that we get one word per line in a data frame, then remove stop words, which are commonly occurring words that we don't deem useful for frequency analysis.

```
data("stop_words")
# Split data by row
Frankenstein = data.frame(text = Frankenstein)
FS_Clean = Frankenstein %>% unnest_tokens( output=word,input=text)
# Remove _ formatting
FS_Clean$word = gsub("_","",FS_Clean$word)

# Removes contents page
FS_Clean = FS_Clean %>%
  slice((71):nrow(.))

# Remove stop words
FS_NoStopWords = FS_Clean %>% anti_join(stop_words)
```

Now we have removed stop words, cleaned the text to remove the preamble, and removed _ formatting, we are ready to proceed with analysis. We analyse the total frequency of each word, then plot the top 5 most frequent words:

```
# Analyse frequency of words overall
FS_Freq = FS_NoStopWords %>%
count( word, sort=TRUE ) %>%
mutate( 'term frequency' = n / sum(n), rank = row_number() )

# Plot total frequency of words overall
FS_Freq %>%
slice_head( n=5) %>%
mutate( word = reorder(word,n) ) %>%
ggplot( aes( x=n, y=word ) ) + geom_col() +
```

```
labs( x="Count", y="Word" ) +
  geom_text(aes(label=n), fontface = "bold", nudge_x = -3, color = "white") +
theme( axis.title=element_text(size=17), axis.text=element_text(size=15) ) +
  theme_minimal()
```



We see all top 5 words have around the same frequency, with the top three, 'life', 'father' and 'eyes', sitting above 100. 'Time' and 'night' sit below 100 with values of 98 and 92 respectively.

b) Which three words are the most specific to Chapter 14?

We consider our Frankenstein book to be a corpus composed of chapters. We are then able to calculate the inverse document frequency for each word in each chapter to find the three most specific words to chapter 14. This is given by the largest tf.idf.

```
# Find chapter numbers
FS_Clean = FS_Clean %>%
mutate( chapter = cumsum( str_detect(
word, regex("chapter", ignore_case = TRUE)
) ) ) %>% filter( chapter > 0 )

# Compute frequency of words by chapter
freq_by_chapter = FS_Clean %>%
  group_by(chapter, word) %>%
  summarise(word_count = n()) %>%
  ungroup()
```

```
# Calculate IDF
freq_by_chapter.idf = freq_by_chapter %>%
bind_tf_idf( word, chapter, word_count) %>%
arrange( desc(tf_idf) ) %>%
mutate( idf=round(idf,2) )

freq_by_chapter.idf_C14 = freq_by_chapter.idf %>% filter(chapter == 14)
kable(head( freq_by_chapter.idf_C14[c(1,2,6)],3 ))
```

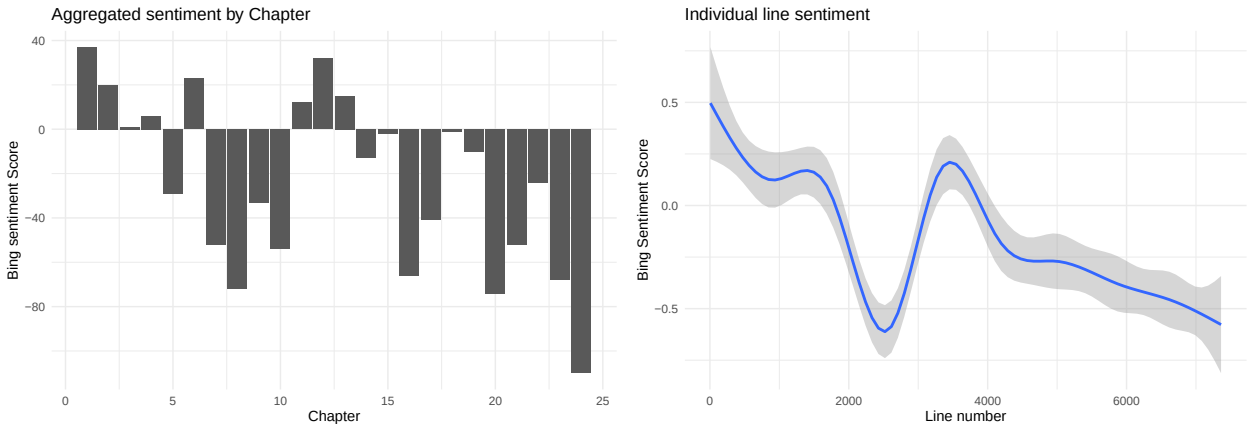
chapter	word	tf_idf
14	turk	0.0154108
14	felix	0.0134447
14	safie	0.0118322

We find the words with the highest IDF value in chapter 14 are Turk, Felix, and Safie. These are all character names within the novel Frankenstein.

c) How does the emotional intent evolve throughout the book?

We care to include stop words in our analysis to ensure we include all data in our analysis. We opt to use the Bing sentiment Lexicon as this has a more expansive list of words, however a benefit of using the AFINN dataset over this is that it assigns a weight to words, instead of just a positive or negative sentiment. We first plot a sentiment by chapter graph demonstrating the overall sentiment for each chapter. We also plot the sentiment over each individual line in order to get a more continuous representation of sentiment throughout the book.

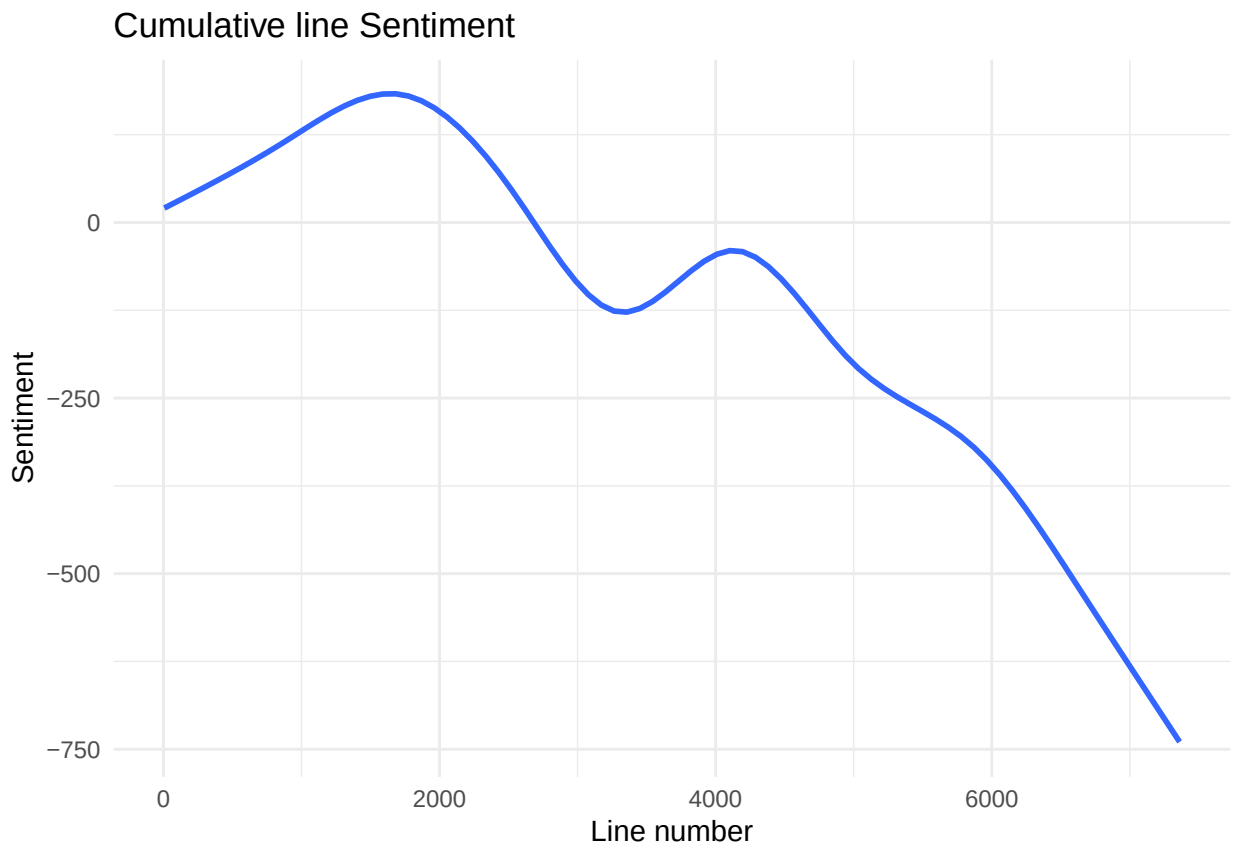
```
# Read in data
Bing_Sentiment_Lexicon = read.csv("Bing Sentiment Lexicon.csv")

freq_by_chapter %>%
inner_join( Bing_Sentiment_Lexicon ) %>% # Join Bing Lexicon
group_by( chapter ) %>%
count( chapter, sentiment ) %>%
pivot_wider( names_from=sentiment, values_from=n, values_fill = 0 ) %>%
  mutate( sentiment = positive - negative ) %>%
ggplot( aes( x=chapter, y=sentiment ) ) +
geom_col() + labs( x="Chapter", y="Bing sentiment Score",
                   title = "Aggregated sentiment by Chapter") +
theme_minimal()

Frankenstein %>% mutate(line = row_number()) %>%  # Identify row number
unnest_tokens( word, text ) %>%
mutate( word = gsub( "_", "", word ) ) %>%
  inner_join( Bing_Sentiment_Lexicon ) %>% # Join Bing Lexicon
  mutate(sentiment_Int = ifelse(sentiment == "positive", 1,-1)) %>%
mutate( sentiment_score = cumsum( sentiment_Int ) ) %>%
  group_by(line) %>% summarise(sentiment_score = sum(sentiment_Int)) %>%
ggplot( aes( x=line, y=sentiment_score ) ) +
geom_smooth() + labs( x="Line number", y="Bing Sentiment Score",
                      title = "Individual line sentiment" ) +
theme_minimal()
```



We first analyse each chapter separately to find which chapters have an overall positive or negative sentiment, and also analyse by line number. We see chapter 24 has the most negative sentiment, and that chapter 1 has the most positive sentiment. By analysing by line number we find an interesting break in the trend from lines 2,500 to 3,500, where the sentiment suddenly becomes more positive. The sharp decrease in sentiment around lines 2,000 to 2,500 corresponds to the time in the book where Justine Moritz is put on trial for murder and is subsequently sentenced to death.

By analysing the cumulative sentiment by line number, we hope to see similar trends, and see exactly where the overall sentiment of the book changes from positive to negative.

```
Frankenstein %>% mutate(line = row_number()) %>%
unnest_tokens( word, text ) %>%
mutate( word = gsub( "_", "", word ) ) %>%
  inner_join( Bing_Sentiment_Lexicon ) %>%
  mutate(sentiment_Int = ifelse(sentiment == "positive", 1,-1),
         sentiment_score = cumsum( sentiment_Int )) %>%
ggplot( aes( x=line, y=sentiment_score ) ) +
 geom_smooth() + labs( x="Line number", y="Sentiment",
                       title = "Cumulative line Sentiment" ) +
  theme_minimal()
```



We see the overall sentiment of the book changing from positive to negative around line 2,300. It does not become positive again after this time. Overall, the data shows us a negative trend in sentiment as the book progresses, however has moments where the sentiment changes sharply, perhaps to keep the reader hooked.

Question 2

The grey squirrel is classified as an invasive species in the UK, and it has displaced the native red squirrel across large parts of the UK. A wildlife conservation charity has collected data on reported sightings of grey squirrels for 2020-2022. The charity assured us that the data is representative of the spatial distribution of squirrels across the UK for all years. They ask you to use the data to investigate the following aspects:

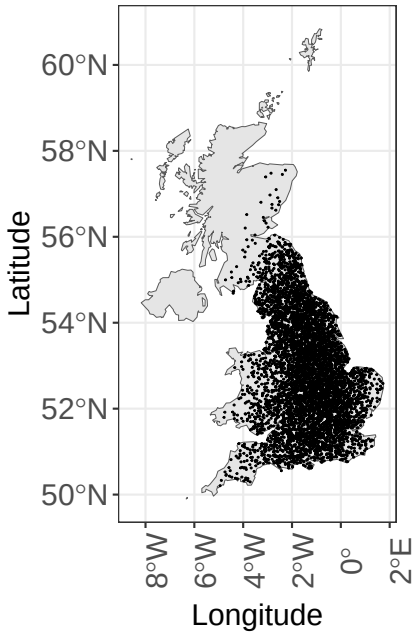
a) What can we say about the spatial distribution of grey squirrels across Great Britain in 2022?

We first read in the data

```
GreySquirrels = read.csv("GreySquirrels.csv")
UK_shp = read_sf("UK Shapefile/UK.shp")
UK_Simplified = st_simplify( UK_shp, dTolerance = 2000, preserveTopology=TRUE )
UK_County = read_sf("UK Shapefile/UK_admin.shp")
```

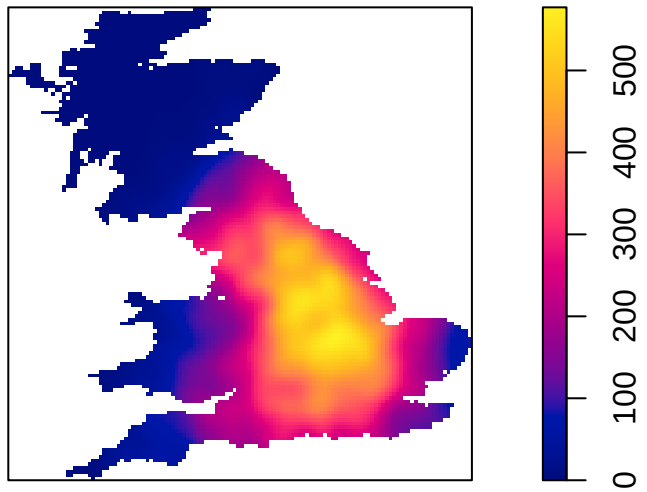
Let's start by conducting some preliminary analysis and identifying the overall distribution of squirrels across the UK, this will help us make an informed decision about what further analysis to conduct.

```
ggplot(UK_Simplified) +   theme_bw() + geom_sf() +
  geom_point(data = filter(GreySquirrels, Year == 2022),aes( x=Lon, y=Lat ), size = 0.05) +
 labs( x="Longitude", y="Latitude" ) +
theme( axis.title=element_text(size=15), axis.text=element_text(size=15),
       axis.text.x = element_text(angle =90))
```



There looks to be an even distribution of Grey Squirrels throughout mainland England, except in areas further west such as Cornwall and Wales. We should analyse the kernel intensity maps given by these points to more accurately gauge the distribution in mainland England.

```
GreySquirrels_2022 = filter(GreySquirrels,Year == 2022)

Squirrels_2022_ppp= ppp(GreySquirrels_2022$Lon,GreySquirrels_2022$Lat,
                        poly = UK_Simplified$geometry[[1]][[1]])
lambdaC_2022 <- density.ppp( Squirrels_2022_ppp, edge=TRUE, sigma =  0.2)
plot(lambdaC_2022, main = "Kernel Intensity map for population in 2022")
```



We find that the population of Grey Squirrels is clustered predominantly in the south eastern region of the UK in 2022. From analysing this cluster, we conclude the point process describing the population is non-homogeneous. In fact, different landscapes and environments provide different challenges for the species to exist, and therefore we would expect non-homogeneity to some extent. An example is Scotland having pine martens (A small ferret-looking animal) which drive off grey squirrels. It would however be reasonable to assume in the populated areas to have homogeneity with clusters of grey squirrels.

b) Are there any areas of Great Britain that saw a notable change in the number of grey squirrels when we compare the data for 2020 and 2022?

We start by calculating the overall population change from 2020 to 2022:

```
GreySquirrel_Yearly = GreySquirrels %>%  group_by(Year) %>% summarise(population = n())
kable(GreySquirrel_Yearly, caption = "Yearly Net Change")
```

Table 2: Yearly Net Change

Year	population
2020	5055
2022	5703

We find grey squirrels have increased by 12.8% across the two years.

We continue by finding administration areas with the largest empirical change in population:

```r
# Group by county and year, and calculate total sightings
Squirrels_by_Area = GreySquirrels %>%
  group_by(County, Year) %>%
  summarise(total_sightings = n()) %>%
  ungroup()  %>%
  pivot_wider(names_from = Year, values_from = total_sightings) %>%
  rename("population_2020" = "2020", "population_2022" = "2022") %>%
  mutate( Difference_2020_to_2022 = (population_2022 - population_2020),
          Percentage_Change = round(100*Difference_2020_to_2022/population_2020,2)) %>%
  arrange(desc(Difference_2020_to_2022))

Largest_Increase = head(Squirrels_by_Area, 5)
Largest_Decrease = tail(filter(Squirrels_by_Area,
                               !is.na(Squirrels_by_Area$Difference_2020_to_2022)),5)
```

```r
kable(Largest_Increase, caption = "The top 5 Locations")
```

Table 3: The top 5 Locations

County	population_2020	population_2022	Difference_2020_to_2022	Percentage_Change
Cumbria	84	235	151	179.76
North Yorkshire	343	449	106	30.90
Northumberland	98	183	85	86.73
Lancashire	58	135	77	132.76
Scottish Borders	24	79	55	229.17

```r
kable(slice(Largest_Decrease, n():1), caption = "The bottom 5 locations")
```

Table 4: The bottom 5 locations

County	population_2020	population_2022	Difference_2020_to_2022	Percentage_Change
East Sussex	55	32	-23	-41.82
Oxfordshire	151	129	-22	-14.57
Kent	82	63	-19	-23.17
Buckinghamshire	99	82	-17	-17.17
Norfolk	143	127	-16	-11.19

We find Cumbria has the largest positive empirical change in Grey Squirrels in UK, with an increase of 180%, and East Sussex having the largest negative change with a decrease of 42%.

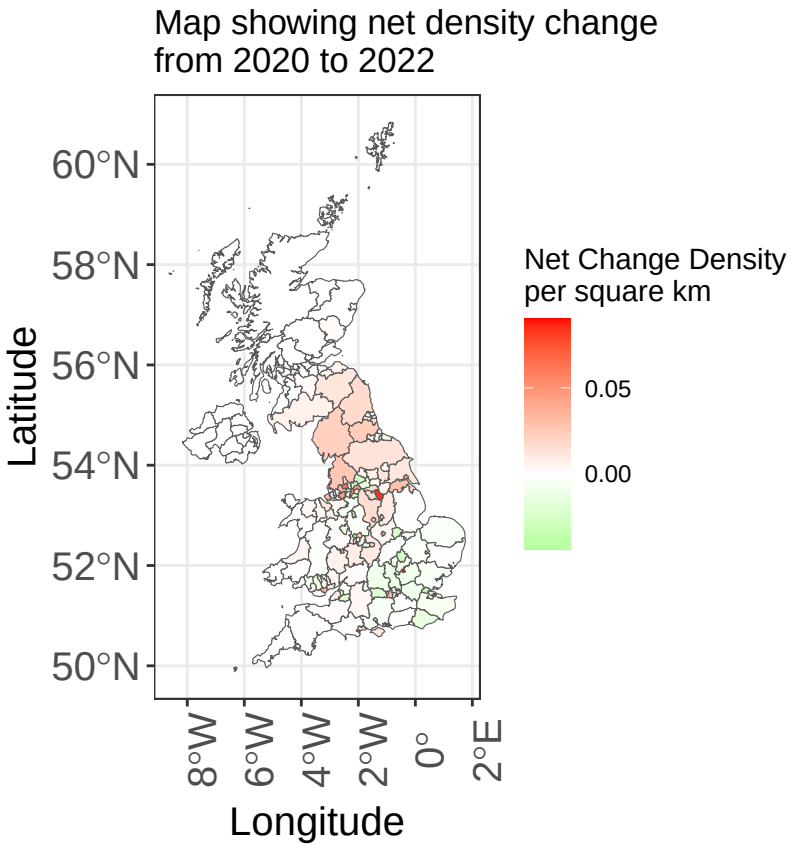
While analysing total net changes is useful to gauge which areas have the largest empirical change, it may be more effective to analyse the population density for different administration areas across the UK. We do this to account for the fact that not all administrations are equal size, and therefore must be treated differently.

```r
UK_County_squirrels = UK_County %>% full_join(Squirrels_by_Area, by=c("NAME_2" = "County"))
UK_County_squirrels$Area_m2 = st_area(UK_County_squirrels) # Calculate area of each region
UK_County_squirrels$Area_km2 = UK_County_squirrels$Area_m2*0.000001 # Convert to km^2
UK_County_squirrels[is.na(UK_County_squirrels)] = 0
UK_County_squirrels = UK_County_squirrels %>%
                      mutate(Density_2020 = population_2020/Area_km2,
```

```
                          Density_2022 = population_2022/Area_km2,
                          Density_diff = Difference_2020_to_2022/Area_km2)
```

We join our shapefile to our dataset by NAME_2 = County. This provides a good match between the two datasets, however some regions have no data and therefore are excluded. We analyse the population density change between 2020 and 2022, to highlight areas that have seen large changes compared to their size.

```
ggplot( data=UK_County_squirrels, aes(fill=as.numeric(Density_diff)) ) +
  geom_sf() + theme_bw() +
  scale_fill_gradient2(low = "green", high="red",
                       name = "Net Change Density \nper square km") +
theme( axis.title=element_text(size=15), axis.text=element_text(size=15),
       axis.text.x = element_text(angle =90)) +
labs( x="Longitude", y="Latitude",
      title = "Map showing net density change \nfrom 2020 to 2022")
```



This plot shows net increases in the population density of squirrels as red, and net decreases as green. We can see that the north of the UK and Scottish Borders have had a noticeable increase in population, with some smaller administrations reporting larger increases. Furthermore, we find that the general population centroid for Grey Squirrels is moving further north. We proceed to analyse the top 5 and bottom 5 counties in terms of net change in population density.

```
Top_5 = UK_County_squirrels %>%
  select(NAME_2,Density_diff, population_2020,population_2022) %>%
  drop_units() %>%
```

```
  slice_max(Density_diff,n=5) %>% st_drop_geometry() %>%
  rename("County" = "NAME_2","Density Difference per km2" = "Density_diff")
kable(Top_5, caption = "The top 5 counties with largest population density change")
```

Table 5: The top 5 counties with largest population density change

County	Density Difference per km2	population_2020	population_2022
Luton	0.0903484	1	5
Rotherham	0.0844524	7	31
Bolton	0.0647893	2	11
Manchester	0.0515961	5	11
Oldham	0.0489026	3	10

```
Bottom_5 = UK_County_squirrels %>%
  select(NAME_2,Density_diff, population_2020,population_2022) %>%
  drop_units() %>%
  slice_min(Density_diff,n=5) %>% st_drop_geometry() %>%
  rename("County" = "NAME_2","Density Difference per km2" = "Density_diff")
kable(Bottom_5, caption = "The bottom 5 counties with largest population density change")
```

Table 6: The bottom 5 counties with largest population density change

County	Density Difference per km2	population_2020	population_2022
Wolverhampton	-0.0441409	6	3
Stoke-on-Trent	-0.0433479	6	2
Salford	-0.0309688	6	3
Slough	-0.0298088	4	3
Thurrock	-0.0297131	6	1

While we can see there are some areas with less Grey Squirrels, the overall trend shows more Grey Squirrels in 2022 than in 2020. We see a notable increase in counties Luton, Rotherham and Bolton. Rotherham being the most notable with the largest empirical change, and second highest population density change. Most areas with decreases we can see are decreasing by small amounts, whereas areas with increases tend to be of more significant value.

Question 3

The Utopian company Amaurot Cookies changed their cookie recipes at the beginning of 2021 due to customer reviews they received. The company would like to understand whether the new recipes have improved customer satisfaction. They thus extracted reviews that contained the word "cookie" for the years 2020-2022, and indicated whether reviews refer to their products or their competitors'. The company would like you to perform a data analysis that uses a sentiment lexicon to address the following two aspects:

We start by reading in the data and the AFINN sentiment Lexicon.

```
CookieReviews = read.csv("CookieReviews.csv")
AFINN = read.csv("AFINN Sentiment Lexicon.csv")
```

a) Are their new cookie recipes more positively received than their previous ones by customers?

For this dataset, we use a more dynamic approach towards sentiment analysis, and opt to use the AFINN dataset. The reasoning is that after trying the Bing sentiment lexicon, it did not capture the difference in sentiment between words such as 'fair' and 'bad' and flags commonly used words such as 'chunky' as negative, when in our case this could be positive. While, admittedly, we have less words to gauge sentiment on, hopefully their accuracy is improved in the following analysis.

```
CookieReviews = CookieReviews %>% mutate(Review_number = row_number())
CookieReviews_Amaurot = CookieReviews %>% filter(Company == "Amaurot Cookies") %>%
  unnest_tokens(Review_word,Text) %>%
  inner_join(AFINN, by = join_by( Review_word==word )) %>%
  mutate(Recipe = ifelse(Year < 2021, "Old Recipe", "New Recipe")) %>%
  group_by() %>% mutate(Total_Words = n()) %>% ungroup()

CookieReviews_Amaurot_grouped = CookieReviews_Amaurot %>% group_by(Review_number,Recipe) %>%
  summarise(Review_Sentiment = mean(value))
```

We filter our dataset, join the AFINN sentiment lexicon and sort into old and new recipes. We then group by the review number to get an overall sentiment per review, and then calculate our violin plots based on overall sentiment per review. Changing the data in this way means that we do not have to consider the difference in frequency between reviews or word count per review.

```
CookieReviews_Amaurot_grouped %>%
  ggplot(aes(x=reorder(Recipe, Review_Sentiment, mean, na.rm=TRUE),y=Review_Sentiment)) +
  geom_violin(trim = FALSE) +
  labs( x="Recipe", y="AFINN Sentiment Score",
title="Violin plots showing Sentiment of old vs new recipe for Amaurot" ) + theme_minimal()
```

From the violin plots, we see both old and new recipes have a similar density at a sentiment of approximately 1.7, with the New recipe skewed slightly towards a higher sentiment score. We find while the new recipe has greater minimum and maximum values, the old recipe has a higher density of points with lower sentiment. If we calculate the quartiles and mean of the data, this should also help us in drawing conclusions.

We want to identify values for Q1, median, Q3 and mean over both datasets in order to find which recipe 'takes the cake'.

```
# Define df for old and new recipe to make calculations more succinct.
CR_A_Old = filter(CookieReviews_Amaurot_grouped, Recipe == "Old Recipe")
CR_A_New = filter(CookieReviews_Amaurot_grouped, Recipe == "New Recipe")

Q_old = quantile(CR_A_Old$Review_Sentiment, prob=c(.25,.5,.75), type=1)
Q_new = quantile(CR_A_New$Review_Sentiment, prob=c(.25,.5,.75), type=1)
OldRecipe_mean = mean(CR_A_Old$Review_Sentiment)
NewRecipe_mean = mean(CR_A_New$Review_Sentiment)
```

We analyse that while the median sentiment on the New and Old recipes are both 2, the new recipe has a higher mean, 1.48 whereas the old recipe has a lower mean of 1.21. This, as well as the lower Q1 and Q3 values for the older recipe implies that the newer recipe has better reviews.

```
Reviews_Per_Year_OldRecipe = length(CR_A_Old$Review_number)
Reviews_Per_Year_NewRecipe = length(CR_A_New$Review_number)/2
```

Furthermore, we analyse that not only the new recipe has better reviews, but more reviews per year have been posted under the second review than the first, further demonstrating that the new recipe is better than the old. We analyse total reviews under the assumption that generally, reviews will only be conducted by people who are greatly in favour of the product, or greatly against.

b) How different are their reviews for 2021-2022 compared to their competitors' in terms of the frequency of words usually attributed with a positive/negative sentiment?

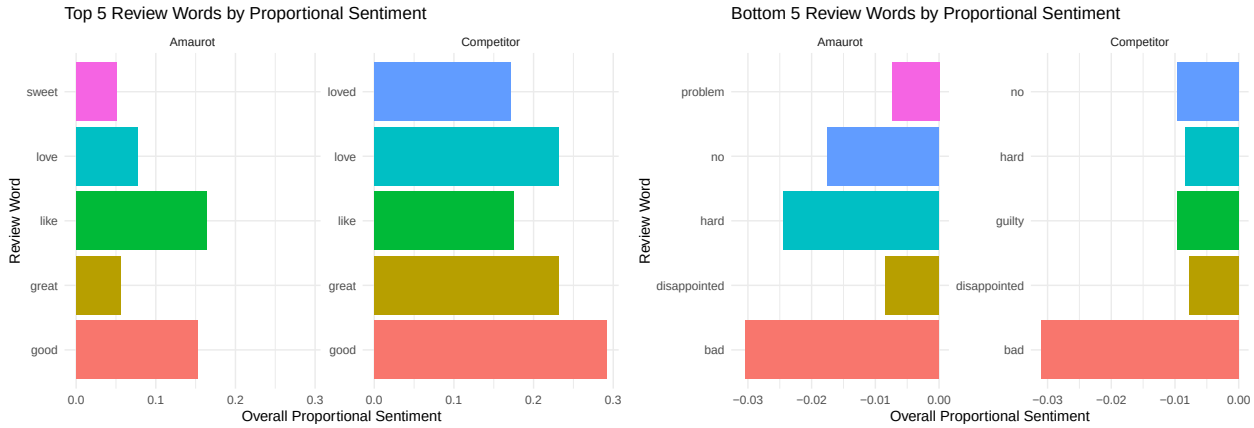
To find the words which were most frequent in each set of reviews, and had the largest sentiment impact, we consider multiplying their frequency and their sentiment score in order to find words which had the largest sentiment impact in the dataset. We can then plot the top 5 and bottom 5 from both Amaurot Cookies and their competitors to help identify differences in the words used.

```
CookieReviews_Competitor = CookieReviews %>% filter(Company == "Competitors") %>%
  unnest_tokens(Review_word,Text) %>%
  inner_join(AFINN, by = join_by( Review_word==word )) %>%
  group_by() %>% mutate(Total_Words = n()) %>% ungroup()

CookieReviews_Competitor_grouped = CookieReviews_Competitor %>%
  group_by(Review_word,value,Total_Words) %>% summarise(Frequency = n()) %>%
  mutate(total_sentiment_impact = value*Frequency,
         proportional_sentiment = total_sentiment_impact/Total_Words)

CookieReviews_Amaurot = CookieReviews_Amaurot %>%
  group_by(Review_word,value,Recipe,Total_Words) %>% summarise(Frequency = n()) %>%
  mutate(total_sentiment_impact = value*Frequency,
         proportional_sentiment = total_sentiment_impact/Total_Words)
```

We filter Amaurot Cookies to only get those with new recipes, and then add a flag called 'review' so that we can combine the datasets for a facet wrap. We then extract the top 5 words from each review set and plot them. We follow a similar method for the bottom 5 words too.

```
Amaurot_Freq = CookieReviews_Amaurot %>% filter(Recipe == "New Recipe") %>%
  select(Review_word, proportional_sentiment) %>% cbind(review = "Amaurot")
Competitor_Freq = CookieReviews_Competitor_grouped %>%
  select(Review_word, proportional_sentiment) %>% cbind(review = "Competitor")
All_Freq = bind_rows(Amaurot_Freq[,3:5],Competitor_Freq[2:4])

All_Freq_top5 = All_Freq %>%
  group_by(review) %>%
  slice_max(order_by = proportional_sentiment, n = 5)

# Plotting with ggplot2
ggplot(All_Freq_top5, aes(x = Review_word, y = proportional_sentiment, fill = Review_word)) +
  geom_bar(stat = "identity") +
  facet_wrap(~review, scales = "free_y") +
  coord_flip() +
  labs(title = "Top 5 Review Words by Proportional Sentiment",
       x = "Review Word",
       y = "Overall Proportional Sentiment") +
  theme_minimal() + theme(legend.position = "")

All_Freq_bottom5 = All_Freq %>%
  group_by(review) %>%
  slice_min(order_by = proportional_sentiment, n = 5, with_ties = FALSE)

# Plotting with ggplot2
ggplot(All_Freq_bottom5, aes(x = Review_word, y = proportional_sentiment, fill = Review_word)) +
  geom_bar(stat = "identity") +
```

```
  facet_wrap(~review, scales = "free_y") +
  coord_flip() +
  labs(title = "Bottom 5 Review Words by Proportional Sentiment",
       x = "Review Word",
       y = "Overall Proportional Sentiment") +
  theme_minimal() + theme(legend.position = "")
```



Interestingly, we find very similar word patterns across both samples, with both samples having a high frequency of words such as 'love', 'like', 'great', 'good' in positive sentiment, and the only difference in the top 5 words being Amaurot having 'sweet' vs the Competitors 'loved'. We find a similar outcome when looking at the bottom 5 words, where we have similarities of words 'problem', 'no', 'hard', 'disappointed' and the only difference being Amaurot's 'problem' vs the Competitors 'guilty'. We recognise the word 'guilty' may not necessarily come with negative intent, and could instead be used in context to 'guilty pleasure' etc, however this is one of the draw backs on conducting sentiment analysis on individual words instead of n-grams.

We calculate an overall sentiment score comparing Amaurot's new recipe against their competitors using the following:

```
mean(CR_A_New$Review_Sentiment) # Amaurot Cookie New recipe reviews
```

```
## [1] 1.480715
```

```
mean(CookieReviews_Competitor$value) # Competitor Reviews
```

```
## [1] 1.995793
```

We see overall the competitor's reviews receive a higher overall mean when analysing sentiment, with Amaurot Cookies' New Recipe recieving a mean sentiment score of 1.48 in comparison to their Competitor's 2.00.

Question 4

The local authorities in the Utopian capital city Amaurot have seen an alarming increase in the number of people being diagnosed with kidney damage. Diagnostic tests revealed that many patients had high lead concentration levels in their blood. The authorities fear that contamination in the tap water may be the source for these increased levels. They thus measured the level of lead concentration in the tap water of each

patient who reported with abdominal pain (another symptom associated with too high lead concentration levels) and ran diagnostics to determine the cause for the patient's pain.

The local authorities have now approached you to help them tackle the health emergency. They provided you with the patient data, and a shapefile and grid for Amaurot. To hide Amaurot's location, constants have been added to the the latitude and longitude coordinates, but the shapes they define are correct. The local authorities ask you to perform the following analysis:

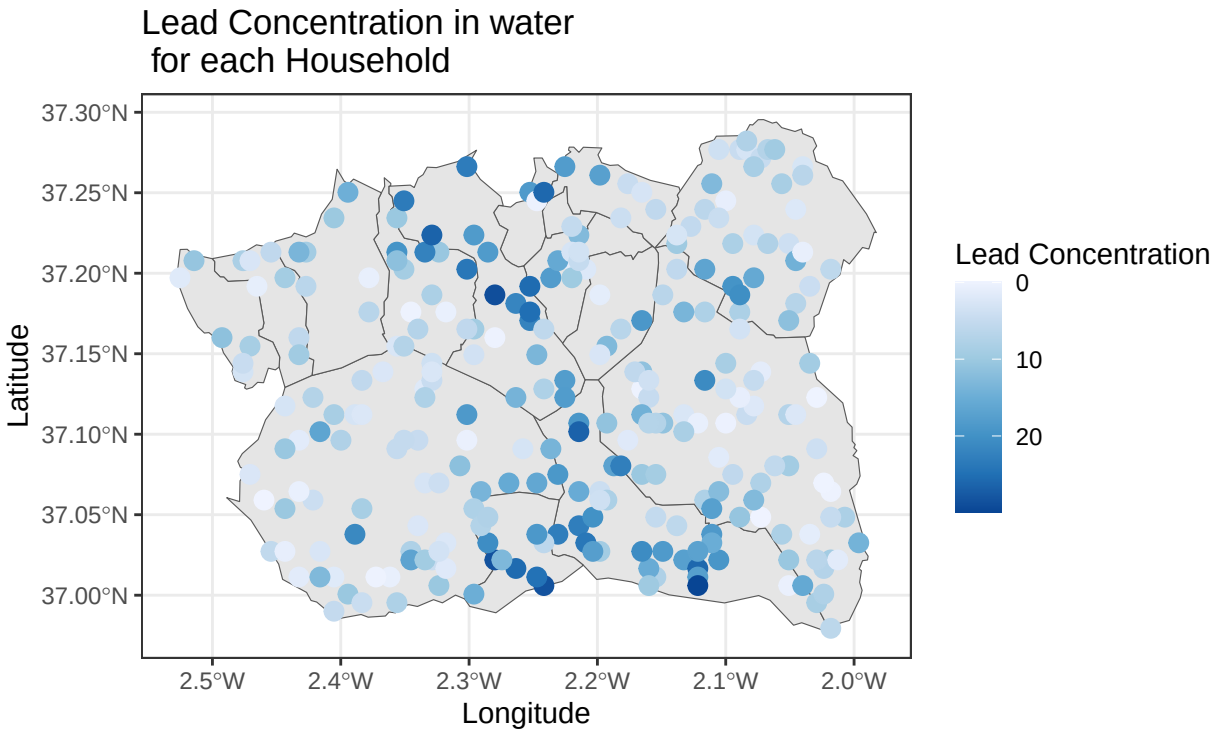
a) Explore the spatial distribution (including the spatial dependence) of the measured lead levels in the city's drinking water.

We read in the data

```
Amaurot_shp = st_read("AmaurotShapefile/AmaurotShapefile.shp", quiet=TRUE)
Amaurot_shp = Amaurot_shp %>% st_simplify( dTolerance = 50)
Amaurot_grid = read.csv("AmaurotGrid.csv")
Amaurot_patients = read.csv("Amaurot Patients.csv")
```

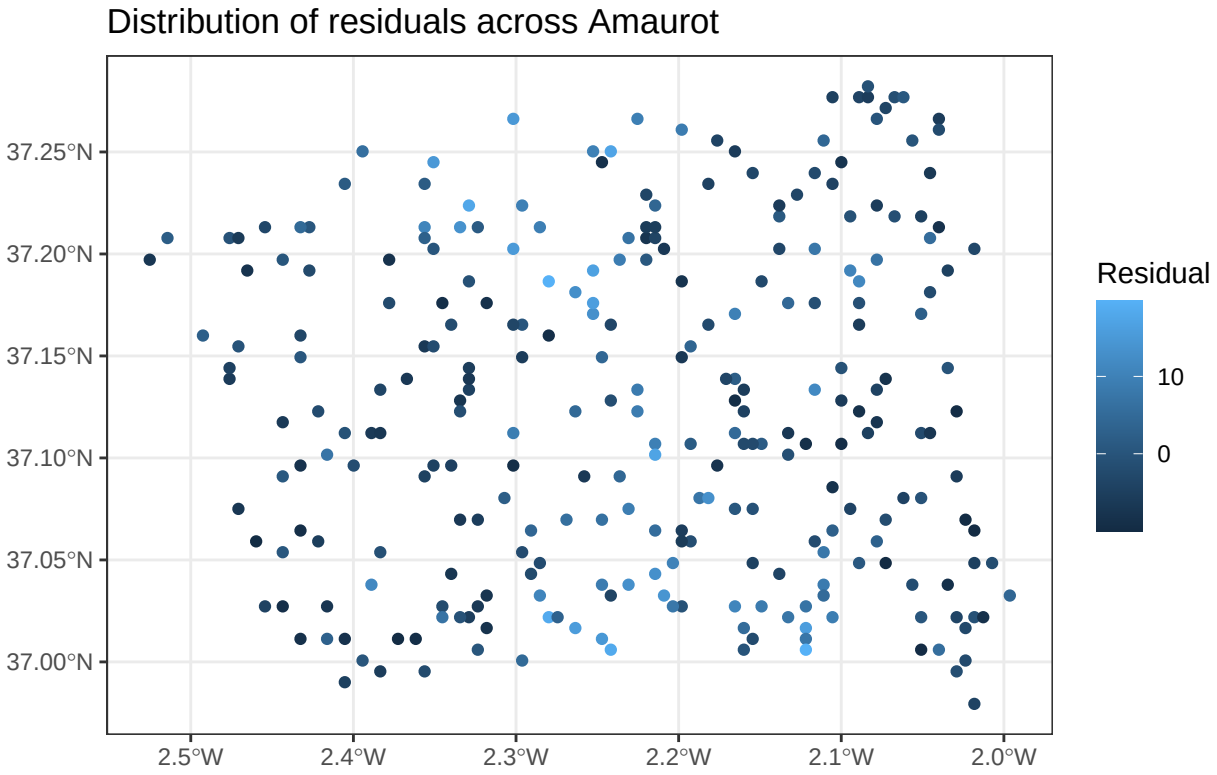
We first plot our data and analyse the overall distribution of points, we do this to check for homogeneity and quality of data throughout our sample region.

```
ggplot( Amaurot_shp ) + theme_bw() + geom_sf() +
geom_point( data=Amaurot_patients, aes(x=Lon, y=Lat, color=Lead), size=3 ) +
scale_color_distiller( palette="Blues", trans="reverse" ) +
labs( x="Longitude", y="Latitude", color="Lead Concentration",
      title = "Lead Concentration in water \n for each Household")
```



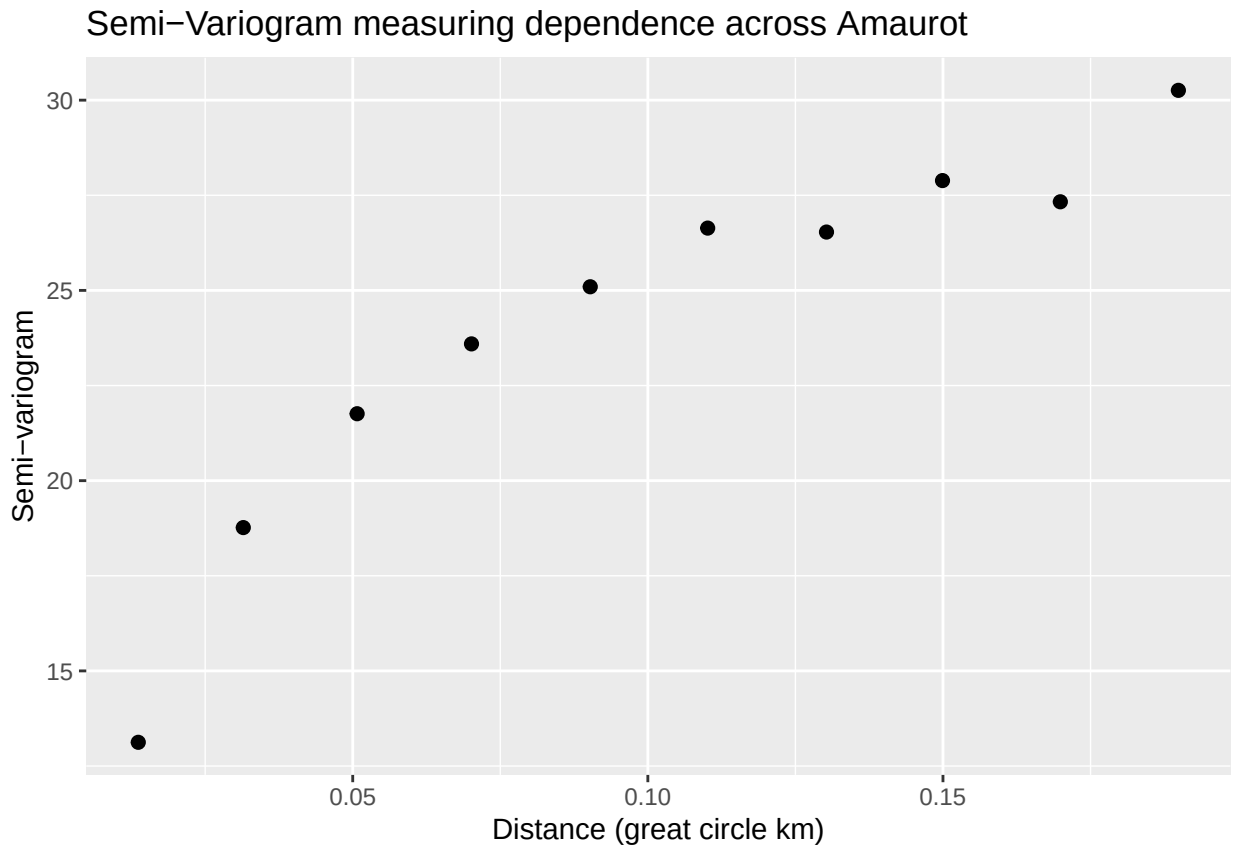
We find the data is homogeneous across Amaurot, with sufficient data points taken in each district. This allows us to have confidence in our analysis. We proceed by analysing the semi-variogram describing the spatial dependence of the Lead concentration by household.

```
Amaurot_patients = drop_na(Amaurot_patients,Lead)

estim = lm( Lead ~ Lon + Lat, data=Amaurot_patients ) # fit a linear model
Amaurot_patients = mutate( Amaurot_patients, Z = estim$residuals ) #calculate residuals

Z_sf = st_as_sf( Amaurot_patients, coords=c("Lon", "Lat") ) %>% st_set_crs( 4326 )
ggplot( data=Z_sf ) + theme_bw() + geom_sf( aes(color=Z) ) +
  labs(color ="Residual",title = "Distribution of residuals across Amaurot")
```



In order to determine whether a semi-variogram is a good analysis technique, we consider the Stationarity and Isotropy of our plot. We are mainly looking to see whether high and low values happen everywhere across the sample space (Isotropic) and the dependence between points can be described by spatial distance (homogeneous). These assumptions seem reasonable, so we continue with our analysis.

```
coordinates(Amaurot_patients) = ~Lon+Lat
gamma_hat <- variogram( Z~1, data = Amaurot_patients, width=0.02, cutoff=0.2 )
ggplot( gamma_hat, aes( x=dist, y=gamma/2 ) ) + geom_point( size=2 ) +
labs( x="Distance (great circle km)", y="Semi-variogram",
      title ="Semi-Variogram measuring dependence across Amaurot" )
```

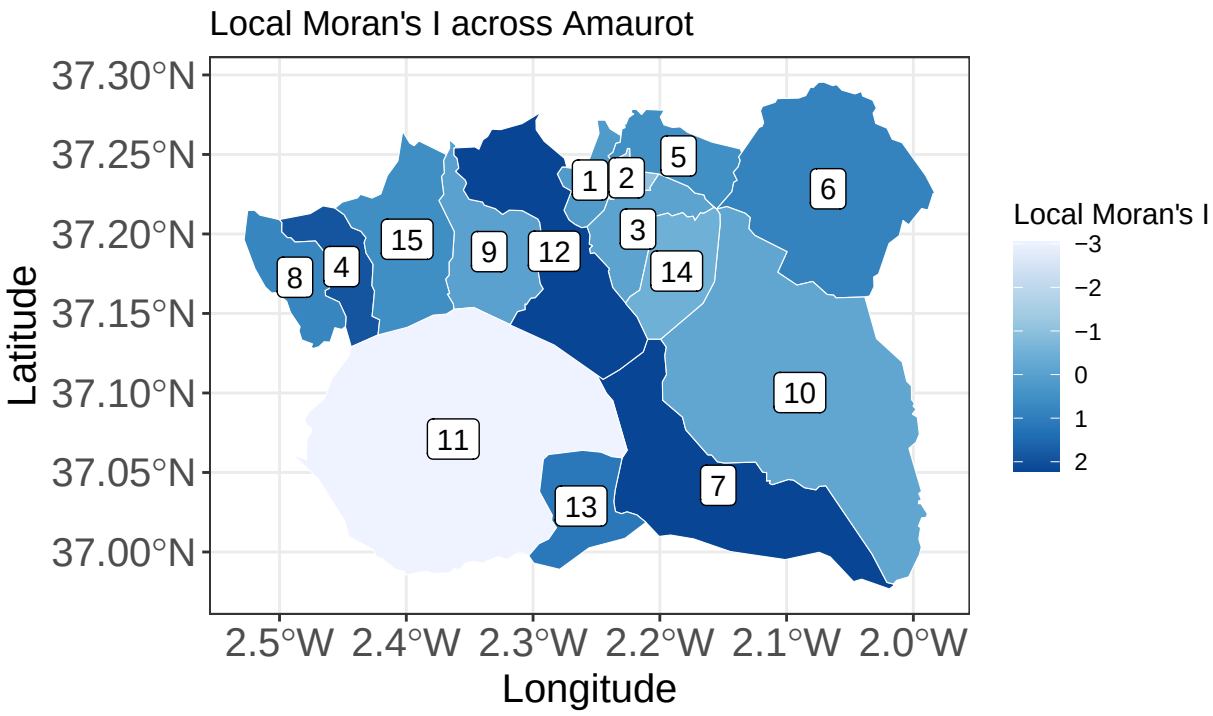


We see that households close together are more spatially dependent, meaning if one has a high concentration of lead, then others in the area are likely to have it too. We see the gradient of the semi-variogram dropping as we go further distances, implying independence of Lead levels for longer distances. The spatial dependence in short distances could be due to the pipes being shared between households at a short distance, however at larger distances different pipes/water networks are used and therefore show less dependence.

We now investigate the total occurrences by district and calculate local Moran's I

```
Amaurot_patients_sf = st_as_sf(Amaurot_patients, coords=c("Lon", "Lat") ) %>%
  st_set_crs( 4326 )
Amaurot_patients = st_join(Amaurot_patients_sf, Amaurot_shp)

Amaurot_shp = Amaurot_shp %>%
  mutate(Clean_District_Name = str_trim(substr(NAME, nchar(NAME) - 1, nchar(NAME))))

Amaurot_patients_district = Amaurot_patients %>%
  group_by(NAME) %>% summarise(Overall_Lead = mean(Lead), Total_Affected = n())

Amaurot_patients_district = as.data.frame(Amaurot_patients_district)
Amaurot_shp = left_join(Amaurot_shp,Amaurot_patients_district, by = "NAME")

# Calculate Local Moran's I
library(spdep)
neighbours_Amaurot = poly2nb(Amaurot_shp)
neighbours_Amaurot = nb2listw( neighbours_Amaurot, style="B" )
MoranLocal = localmoran( x=Amaurot_shp$Overall_Lead, listw=neighbours_Amaurot )
MoranLocal = st_drop_geometry(MoranLocal)
```

```
Amaurot_shp = cbind(Amaurot_shp,MoranLocal)

ggplot( data=Amaurot_shp, aes(fill=Ii, label = Clean_District_Name )) +
  geom_sf(color = "white") + theme_bw() +
scale_fill_distiller( palette="Blues", trans="reverse" ) +
    geom_sf_label(fill = "white", fun.geometry = sf::st_centroid) +
theme( axis.title=element_text(size=15), axis.text=element_text(size=15) ) +
labs( x="Longitude", y="Latitude", fill = "Local Moran's I",
      title = "Local Moran's I across Amaurot")
```



Labels have been added to the graph to add clarity on which district number refers to which region, this will become particularly useful in the future when we want to compare districts using map visualisations. We are able to determine that districts 7, 12 and 13 all have high dependence on one-another, perhaps suggesting they are part of the same network supplying water to the regions.

b) Identify a threshold for lead concentration in drinking water beyond which the risk of a person developing kidney damage increases.

We start by classifying our data into three groups, those with Kidney Damage, those with no disease, and those with another related disease.

```
# Classify into 3 groups, those with kidney damage,
# those with other disease, and those with no disease.
Amaurot_patients = Amaurot_patients %>%
  mutate(Kidney_damage_analysis = case_when(
```

```
    str_detect(Disease, "Kidney Disease") ~ "Kidney Damage",
    str_detect(Disease, "No treatment required") ~ "No treatment required",
    TRUE ~ "Other Disease"))
```

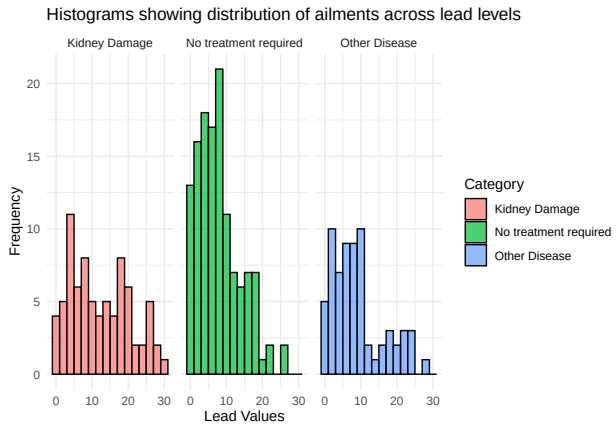
Next, we want a way to analyse our data to find a threshold value for which kidney damage becomes more common over other diseases, we would expect to find this value by identifying a visible spike in kidney damage cases when compared to other disease. We note that we wouldn't necessarily expect to see a spike over the 'no treatment required' category, since this is far more common over our sample.

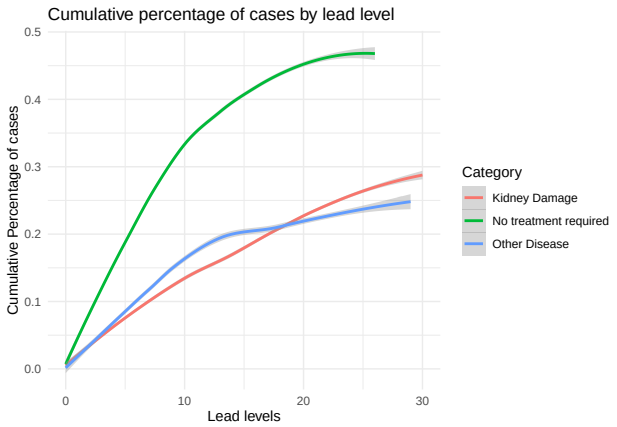
We round lead values to every 0.5, calculate a cumulative sum for each category, then divide our cumulative sum by the total frequency in order to analyse the proportion of cases at each lead level increment.

```
# Take lead values and round to 0.5, group by ailment and lead levels,
# calculate individual and total frequencies.
Amaurot_patients_leadAnalysis = Amaurot_patients %>%
  mutate(Lead_levels_rounded = round(Lead*2)/2) %>%
  group_by(Kidney_damage_analysis,Lead_levels_rounded) %>%
  summarise(Frequency = n()) %>%
  group_by() %>% mutate(TotalFreq = sum(Frequency)) %>% ungroup()
# Divide frequency by total frequency within each ailment
# in order to analyse % cumulative graph
Amaurot_patients_leadAnalysis_cumsum = Amaurot_patients_leadAnalysis %>%
  group_by(Kidney_damage_analysis) %>% mutate(CumulativePerc = cumsum(Frequency)/TotalFreq)
```

We now use the above data wrangling to plot cumulative percentages against lead levels to find a threshold.

```
ggplot(Amaurot_patients, aes(x = Lead, fill = Kidney_damage_analysis)) +
geom_histogram(binwidth = 2, position = "identity", alpha = 0.7, color = "black") +
facet_wrap(~Kidney_damage_analysis) +
labs(title = "Histograms showing distribution of ailments across lead levels",
     x = "Lead Values", y = "Frequency", fill = "Category") +
theme_minimal()

ggplot(Amaurot_patients_leadAnalysis_cumsum,
       aes(x = Lead_levels_rounded, y = CumulativePerc,
           color = Kidney_damage_analysis, group = Kidney_damage_analysis)) +
  geom_smooth() +
  labs(title = "Cumulative percentage of cases by lead level",
       x = "Lead levels",
       y = "Cumulative Percentage of cases", color = "Category") +
  theme_minimal()
```





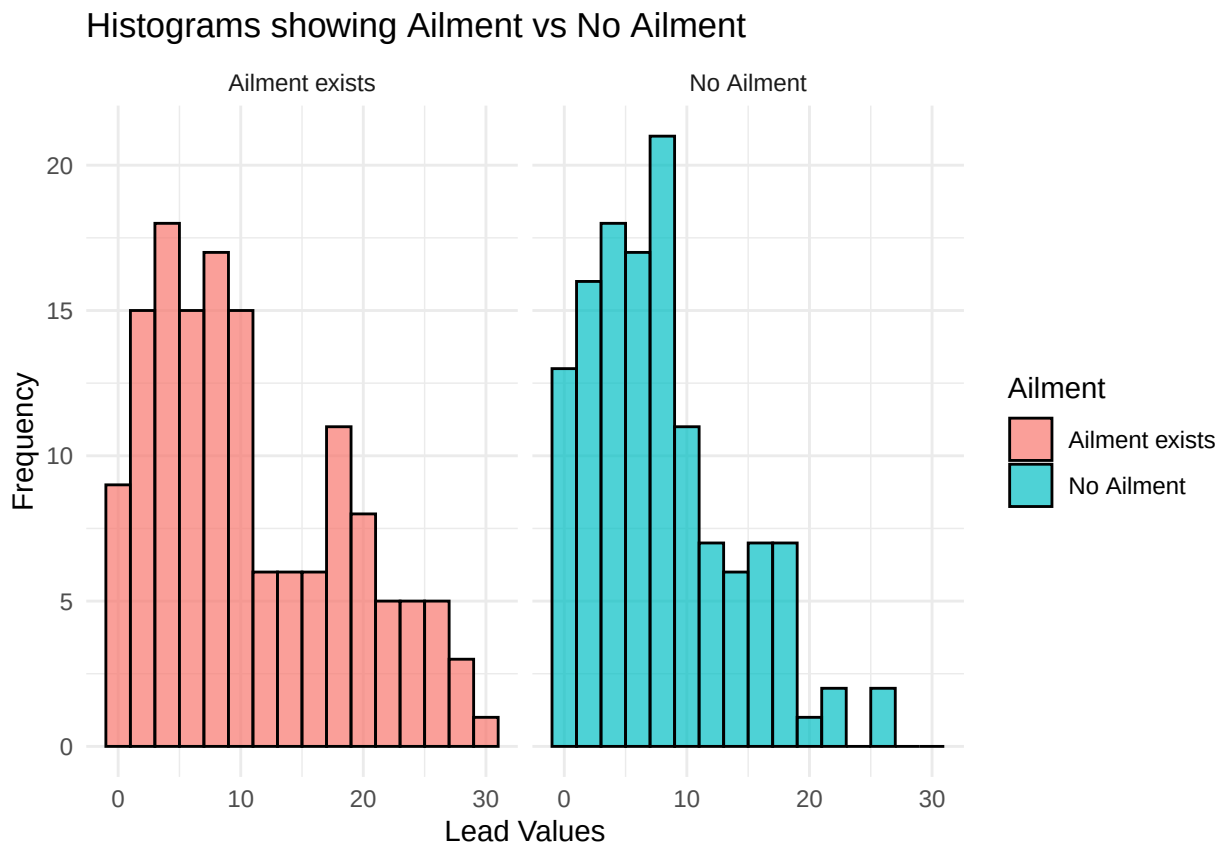
The above cumulative percentage graph demonstrates the distribution at different lead concentrations of presenting one of the known ailments. We want to find the point in which the likelihood of having kidney damage increases. We can see that lead levels 17.5, it is more likely for households above this threshold to have Kidney damage compared to other diseases, whereas below this threshold, other ailments present themselves more commonly. Throughout the dataset we can see that there are still a high proportion of households presenting no ailments when compared to those who do. This is most likely because many houses measured have had low lead levels and while presented with stomach pains, no long term diseases were found - as demonstrated by the steep gradient at low lead levels and shallow gradient at higher levels.

We do note that while kidney disease and lead levels will almost surely have a positive correlation, kidney disease and other diseases can all occur independently of lead levels, and therefore while we can find an approximate threshold, more analysis is required beyond the scope of the course.

c) The local authorities wish to deploy a team to one of the districts to reduce lead concentration to a safe level. Which district would benefit the most from such an action?

We want to reduce lead concentrations to a safe level within one region. We have established a 'safe' level for kidney disease alone, but not for all ailments. We first analyse what a 'safe' level should be, then use this to make an informed decision on which district should be targeted first.

```
Amaurot_patients = Amaurot_patients %>%
mutate(Ailment = case_when(Kidney_damage_analysis == "No treatment required" ~ "No Ailment",
                           TRUE ~ "Ailment exists"))

ggplot(Amaurot_patients, aes(x = Lead, fill = Ailment)) +
geom_histogram(binwidth = 2, position = "identity", alpha = 0.7, color = "black") +
facet_wrap(~Ailment) +
labs(title = "Histograms showing Ailment vs No Ailment", x = "Lead Values", y = "Frequency") +
theme_minimal()
```



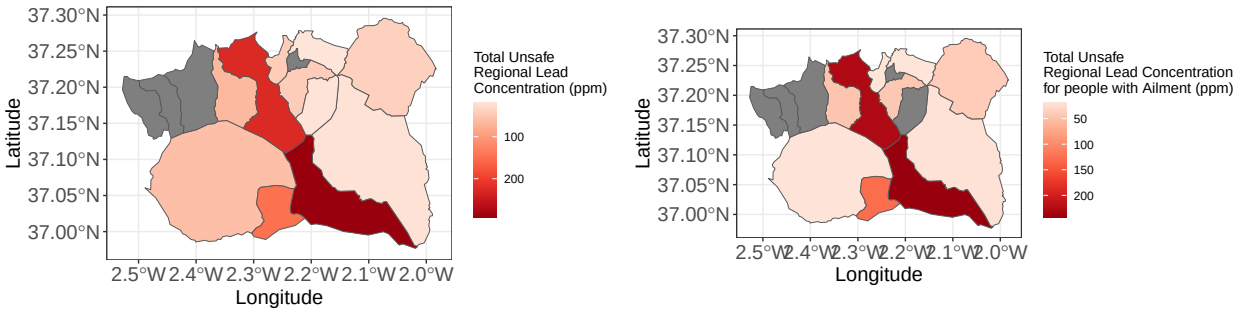
We find from our histograms that there is a spike in the number of cases with ailments after lead levels of approximately 18, whereas the graph with 'no treatment required' seems to drop off, indicating not many households have no ailments after lead levels hit 18. This justifies us in our analysis to consider any values over 18 to no longer be safe, however values below this pose less threat. We continue by filtering out data with lead levels less than 18, finding the total frequency and mean. We do the same but also filter on households which report ailment. Using this data we will be able to plot the total lead concentrations per area on a map.

```
Amaurot_patients_UnsafeLevels = Amaurot_patients %>%
  filter(Lead > 18) %>%
  group_by(NAME) %>%
  summarise(Frequency_All = n(), Mean_All = mean(Lead)) %>%
  as.data.frame()

Amaurot_patients_UnsafeLevels_WithAilment = Amaurot_patients %>%
  filter(Lead > 18) %>%
  filter(Ailment == "Ailment exists") %>%
  group_by(NAME) %>%
  summarise(Frequency_Ailment = n(), Mean_Ailment = mean(Lead)) %>%
  as.data.frame()

Amaurot_patients_UnsafeLevels = full_join(Amaurot_patients_UnsafeLevels,
  Amaurot_patients_UnsafeLevels_WithAilment,
  by = "NAME") %>%
  select(NAME,Frequency_All,Mean_All,
  Frequency_Ailment,Mean_Ailment) %>%
  mutate(Total_Lead_conc = Frequency_All*Mean_All,
```

```
  Ailment_Lead_conc = Frequency_Ailment*Mean_Ailment)

Amaurot_shp = Amaurot_shp %>% left_join(Amaurot_patients_UnsafeLevels,by = "NAME")
```

We continue by plotting our data to show areas with the highest unsafe levels of lead in water, and also those with unsafe levels and ailment. This will allow us to determine which area is best focused on by the local authorities.

```
ggplot( data=Amaurot_shp, aes(fill=Total_Lead_conc) ) +
geom_sf() + theme_bw() +
scale_fill_distiller( palette="Reds", trans="reverse" ) +
theme( axis.title=element_text(size=15), axis.text=element_text(size=15) ) +
labs( x="Longitude", y="Latitude",
      fill = "Total Unsafe \nRegional Lead \nConcentration (ppm)" )

ggplot( data=Amaurot_shp, aes(fill=Ailment_Lead_conc) ) +
geom_sf() + theme_bw() +
scale_fill_distiller( palette="Reds", trans="reverse" ) +
theme( axis.title=element_text(size=15), axis.text=element_text(size=15) ) +
labs( x="Longitude", y="Latitude",
      fill = "Total Unsafe \nRegional Lead Concentration \nfor people with Ailment (ppm)" )
```



When both looking at total lead concentration in a district, and lead concentration from only those with ailments, we find a similar picture. District 7 has the most dangerous lead concentration out of all districts. Both plots agree with the conclusion that District 7 is the worst, in both terms of households with ailment, and overall.

We consider total concentration because by only looking at mean levels within regions we can't tell exactly how many people are affected, it would be inappropriate to suggest the local government to treat an area with the highest average lead levels if there are only 3 houses affected in that region. It is therefore more appropriate to consider the total lead concentration in a region as a gauge on how many people are affected, as well as how dangerous the levels are.

d) Write a non-scientific summary of your analysis for parts (a)-(c) that can be understood by someone with A-level Mathematics knowledge. State possible recommendations that may be of interest to the authorities of Amaurot.

We were tasked by Amaurot Government to analyse the lead levels across their city, as they have seen an alarming number of patients expressing symptoms common with lead poisoning. We were given a dataset of

households, their lead concentration in tap water, and their locations. We found the data was well collected and appropriate for analysis.

By exploring the spatial distribution of measured lead levels, we find households within close proximity (less than 100m) of one another were likely to have similar lead levels, whereas households situated further than 100m away from the data point tended to have independent concentrations. The report therefore recommends the local government should identify regions where testing has not been conducted, and spot test households at 100m increments, to ensure the most effective data sampling. Furthermore, households found to be within 100m of a household with unsafe lead concentrations should be notified that they are at risk. We also find that Districts 7, 12 and 13 have a high level of dependence on their neighbouring districts, the report recommends the government further analyses their water network as it is a possibility that these regions run under the same water network, this theory further supported by the fact they are all neighbouring districts.

We found a threshold at which kidney damage becomes the most common disease type as 17.5ppm (parts per million), however it is suggested that the government aims to keep levels far below this, as other diseases can still occur below this threshold; particularly in vulnerable communities. As expected, the rate at which we have no disease present plateaus as lead concentration levels increase. Other diseases are more likely to occur at lower lead concentrations, however they are less common at higher lead levels where kidney disease becomes prevalent. The report recommends more data is collected across Amaurot, and a 'normal' level of kidney disease is defined, this way hypothesis testing can be conducted on whether levels presented are statistically significant.

In order to determine where the government should send a team to, we calculated a total lead concentration based on levels above the threshold we considered safe. This allowed us to prioritise districts with both high lead levels, and high frequency of occurrence. We concluded the government should send a team to District 7, which has the highest total unsafe lead concentration - a frequency of 13 houses found to have dangerous levels with an average of 22.4ppm. District 7 was also found to have the highest total unsafe lead concentration in households that present disease - a total of 11 houses found to have dangerous lead concentration and presented ailments, with an average concentration of 22ppm. Furthermore, the government should conduct further testing in the surrounding districts, as we have determined the lead concentrations in Districts 12 and 13 depend on neighbouring districts, which District 7 is.

The report recommends the local government sets up an efficient system where people who report any of the symptoms related to lead poisoning have a toxicology test to determine lead levels in their system, and also have their water tested within their household. By making the general public aware of the risk, the government can act on predictive analysis and preventative measures.