Coursework 2 - Solution and Marking Guidance

Christian Rohrbeck

April 2025

```
library(dplyr)
library(ggplot2)
library(tidyr)
library(tidytext)
library(patchwork)
library(lubridate)
library(sf)
library(gstat)
library(sp)
library(patchwork)
library(spatstat)
```

Question 1

Generative AI tools have gained a reputation for their ability to produce texts in the style of famous authors. We want to put this claim to the test using the work by Charles Dickens.

The file "Dickens_Fakes.csv" contains five pieces of text written by ChatGPT in the style of Charles Dickens, and "Dickens_Originals.csv" provides ten randomly selected chapters from the books "A Tale of Two Cities" and "Great Expectations". Explore the similarities and differences between the original and artificially produced pieces of text. Your analysis should consider both the words used and the emotional intent of the texts. What do you conclude regarding ChatGPT's ability to produce work in the style of Charles Dickens? How confident should we be about your conclusion?

Rationale

This question assesses the student's ability to perform a range of methods in text data analysis. In particular, the students need to consider both term frequency and sentiment of the text. There is also the potential to employ topic modelling to see whether the texts can be separated using unsupervised learning techniques. However, such an approach requires a lot of care, because one would need to remove characters and locations from the texts to avoid the analysis being biased towards these words.

The students have come across the work by Charles Dickens as an example in the lecture notes, but we never considered text generated by ChatGPT - so the students are partly familiar with the style of text, but not with the problem as such. They have also seen some examples on how to compare pieces of text, but we often only considered two books. As such, students have to clearly explain how they approach the comparison. Based on their submission to Coursework 1, I expect that many students will again not provide enough detail, or even discuss the limitations of their analysis.

Example of a first class solution

We start by loading the text data from the two data files

```
Fakes_raw <- read.csv("Dickens_Fakes.csv", encoding = "UTF-8")
Originals_raw <- read.csv("Dickens_Originals.csv", encoding = "UTF-8")
```

The first comparison we perform will focus on term frequency, that is, the frequency with which words appear within a text. We consider the five AI generated texts as a whole as they are all produced by the same generative AI algorithm, and we remove any stop words from the analysis:

```
Fakes_words <- Fakes_raw %>%
  unnest_tokens( word, Text ) %>%
  mutate( word = gsub( "_", "", word ) ) %>%
  count( word, sort = TRUE ) %>%
  mutate( tf = n / sum(n) ) %>%
  anti_join( stop_words, by="word" )
```

Remark: I would accept if stop words were kept in this context, but it needs to be explained

We now perform the same steps for the ten chapters by Charles Dickens:

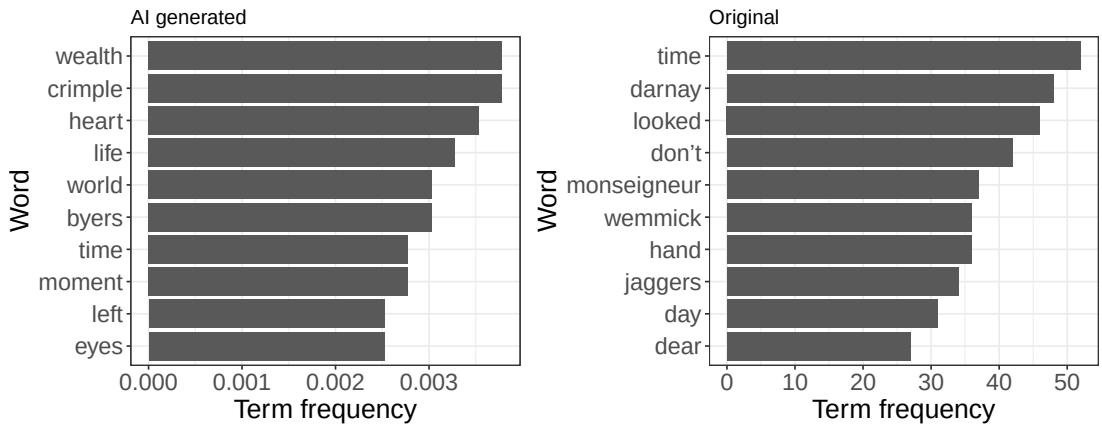
```
Originals_words <- Originals_raw %>%
  unnest_tokens( word, Text ) %>%
  mutate( word = gsub( "_", "", word ) ) %>%
  count( word, sort = TRUE ) %>%
  mutate( tf = n / sum(n) ) %>%
  anti_join( stop_words, by="word" )
```

Let's have a look at the ten most common words in each of the two data sets:

```
plot_Fakes <- Fakes_words %>%
  slice_head( n=10 ) %>%
  mutate( word = reorder(word,tf) ) %>%
  ggplot( aes( x=tf, y=word ) ) + geom_col() +
  labs( x="Term frequency", y="Word", title="AI generated" ) + theme_bw() +
  theme( axis.title=element_text(size=17), axis.text=element_text(size=15) )

plot_Originals <- Originals_words %>%
  slice_head( n=10 ) %>%
  mutate( word = reorder(word,tf) ) %>%
  ggplot( aes( x=n, y=word ) ) + geom_col() +
  labs( x="Term frequency", y="Word", title="Original" ) + theme_bw() +
  theme( axis.title=element_text(size=17), axis.text=element_text(size=15) )

plot_Fakes + plot_Originals
```

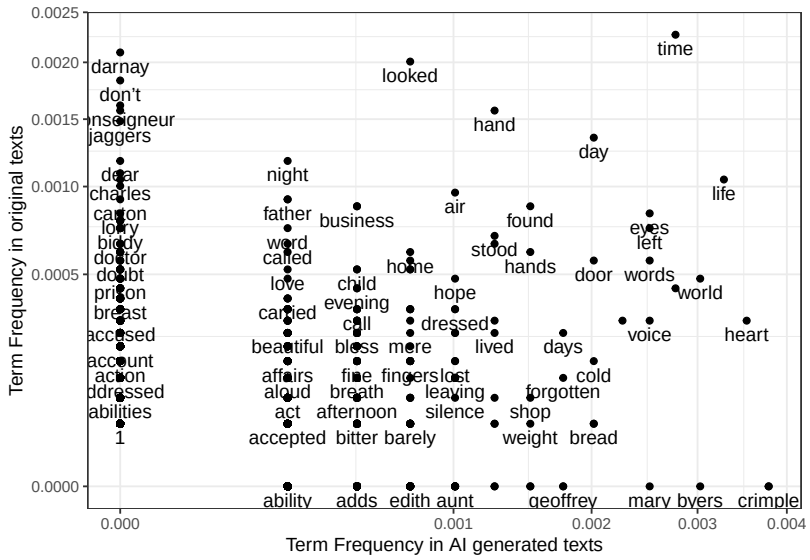


The word "time" is the only one which appears in both plots. Let's investigate this further and plot the frequencies for all words against each, we combine the calculated frequencies into a single data frame and any missing values are replaced by a zero - NAs in this case correspond to the word only occurring in one of the two data sets:

```
Words <- full_join( Fakes_words, Originals_words, by="word" ) %>%
  rename( Fakes=tf.x, Originals=tf.y ) %>%
  mutate( Fakes = case_when( is.na(Fakes) == TRUE ~ 0, .default = Fakes),
          Originals = case_when( is.na(Originals) == TRUE ~ 0, .default = Originals) )
```

Finally, we produce a scatter plot of the different term frequencies:

```
ggplot( Words, aes( x=Fakes, y=Originals ) ) +
  geom_point() +
  geom_text( aes(label=word), check_overlap = TRUE, vjust=1.5 ) +
  coord_trans( x="sqrt", y="sqrt" ) + theme_bw() +
  labs( x="Term Frequency in AI generated texts",
        y="Term Frequency in original texts" )
```



The plot does suggest that the two text data sets are quite different in terms of the words within them.

Let's move on to analyzing the emotional intent. We start by loading the AFINN sentiment lexicon:

```
AFINN <- read.csv("AFINN Sentiment Lexicon.csv")
```

Remark: It does not matter whether the AFINN or Bing sentiment lexicon is used here. Using both would be considered excessive.

We now consider the individual pieces of text within each data set individually and derive a sentiment score for each. Here, we define the sentiment score as the average sentiment within the text. This approach requires that the sentiment can be accurately assessed by assigning a sentiment value to each word and then averaging the sentiment across the words in the text.

First we derive the sentiment for the five AI generated stories:

```
Fakes_sentiment <- Fakes_raw %>%
  unnest_tokens( word, Text ) %>%
  mutate( word = gsub( "_", "", word ) ) %>%
  inner_join( AFINN, by="word" ) %>%
  group_by( Title ) %>%
  summarise( sentiment = mean(value) )
```

We now repeat this procedure for the original ten chapters:

```
Originals_sentiment <- Originals_raw %>%
  unnest_tokens( word, Text ) %>%
  mutate( word = gsub( "_", "", word ) ) %>%
  inner_join( AFINN, by="word" ) %>%
  group_by( Title, Chapter ) %>%
  summarise( sentiment = mean(value) )
```

Let's consider the calculated sentiment scores:

```
Fakes_sentiment
```

```
## # A tibble: 5 x 2
##   Title                   sentiment
##   <chr>                       <dbl>
## 1 The Downtrodden Poet       0.2
## 2 The Fallen Gentleman      -0.3
## 3 The Miser's Toll          -0.174
## 4 The Orphan's Heart         0.0111
## 5 The Shopkeeper's Lament   -0.893
```

```
Originals_sentiment
```

```
## # A tibble: 10 x 3
## # Groups:   Title [2]
##    Title                Chapter sentiment
##    <chr>                  <int>     <dbl>
##  1 A Tale of Two Cities      10    0.304
##  2 A Tale of Two Cities      13    0.271
##  3 A Tale of Two Cities      23    0.310
##  4 A Tale of Two Cities      26    0.533
##  5 A Tale of Two Cities      36   -0.270
##  6 Great Expectations        14    0.0556
##  7 Great Expectations        20   -0.371
##  8 Great Expectations        31   -0.0157
##  9 Great Expectations        32    0.392
## 10 Great Expectations        58    0.734
```

We see that there is quite a variety in the scores for both kinds of text, but there is an indication that the AI generated pieces of text are more negative than the originals. Let's have a look:

```
Fakes_sentiment %>% summarise( Mean=mean(sentiment), Median=median(sentiment) )
```

```
## # A tibble: 1 x 2
##     Mean Median
##    <dbl>  <dbl>
## 1 -0.231 -0.174
```

```
Originals_sentiment %>% summarise( Mean=mean(sentiment), Median=median(sentiment) )
```

```
## # A tibble: 2 x 3
##   Title                 Mean Median
##   <chr>                <dbl>  <dbl>
## 1 A Tale of Two Cities 0.229 0.304
## 2 Great Expectations   0.159 0.0556
```

We see that there is quite a difference in the sentiment scores.

To summarize, we find that there are marked differences between the AI generated and original texts, both in terms of the words used and the emotional intent. As such, we would disagree with the claim that the AI generated text really captures the style by Charles Dickens.

However, there are quite a few caveats to our conclusion. Firstly, we only studied a small subset of the work by Charles Dickens and the five pieces of generated text are quite short, which means that there is a quite a bit of uncertainty in the estimated term frequencies. Secondly, we may just by chance sampled chapters which are more positive than the actual work by Charles Dickens - we have to remember we only study two books here. Finally, a judgement based on words used and sentiment of short pieces of text is unlikely to fully explore the writing style - Charles Dickens is renowned for his social commentary, including highlighting social issues such as poverty, child labour and class disparity.

Marking Guidance

An analysis as above would fetch 12 out of 15 marks (i.e. 80%). A higher mark can be achieved by also applying topic modelling and clearly arguing for why it may be useful, without the analysis becoming too excessive. A more in-depth analysis of word frequency may also be awarded with a higher mark.

The following list outlines some scenarios and the mark they would achieve:

- **11 marks** - Analysis is as above but there are some minor weaknesses in terms of the discussion of the assumptions or the level of detail regarding the reliabiliy of the conclusions
- **10 marks** - Assumptions are not clearly stated, but the rest of the analysis is as above
- **9 marks** - Overall approach is okay, but the report barely considers assumptions and limitations underpinning the analysis.
- **8 marks** - Analysis considers sentiment and term frequency, but the choice of plots/methods is not always suitable, and assumptions/limitations are not discussed in enough detail
- **6-7 marks** - Some attempt at answering the questions and the report demonstrates some understanding of how data visualization may be used to answer the question. The overall analysis lacks however depth and assumption and conclusions are not discussed
- **0-5 marks** - An answer which demonstrates very little understanding of the text data analysis introduced in the course.

Minor issues in presentation or an excessive number of plots should be penalized by deducting up to 2 marks, unless it results in the student achieving a failing grade. In extreme cases, such as printing the data files or providing way too many plots, more than 2 marks may be deducted.

Question 2

There are two tasks the government of Utopia asks you to complete using the provided data:

i. *Explore the spatial distribution (including spatial dependence) of the monthly average deficit in groundwater levels for the 102 monitoring sites separately for March and August. Discuss all the assumptions you make for the analysis.*

ii. *Produce maps which provide estimates for the monthly average deficit in groundwater levels across the region for March and August 2024. Discuss the reliability of your estimates.*

Rationale

This question assesses several of the key concepts and techniques for point-referenced data we introduced in the course, and also presents a couple of extra challenges which are intended to separate the good from the best students.

At the beginning, students need to restructure the data using, for instance, the techniques on data wrangling we covered in the first weeks of the course. Next, they need to demonstrate that they can visualize and correctly interpret their plots. I expect that almost all students will manage this task.

The first task further requires the production of semi-variograms for each of the two months and to discuss whether the assumptions hold or not. Students have seen quite a few examples on this, but some additional complications were added. Specifically, the data for August does not satisfy the assumption of stationarity, but this is not clearly visible from the plot. Students have seen one example in the problem sheets where a region was divided into two parts and semi-variograms for the subregions were compared. If they follow the same approach, they will find that the semi-variograms are quite different for the west and east of the region. I would expect that about half of students will identify this issue and that it will form a barrier splitting 2.1 (and stronger) students from the weaker students.

For the final part, students should use inverse distance weighting as introduced in the lectures to produce a map of estimates. I expect that most students will be able to produce the maps, although some students may select a poor value for the power parameter p. The discussion on reliability is likely the distinguishing factor between the best students - it was never really explained in the course that inverse distance weighting may give poor results when we are dealing with non-stationarity in the data. So weaker students may argue that reliability is good, while only the very best students will give an answer which uses the conclusions from the previous tasks.

Example of a first class solution

We first load the data sets and the shapefile

```
Water     <- read.csv( "Tesremos_Water.csv" )
Tesremos  <- read_sf( "Tesremos_Shapefile/Tesremos.shp" )
Locations <- read.csv("Tesremos_Locations.csv" )
```

Task i: Visualization of average monthly values The first step is to do some data cleaning and wrangling. Let's start by converting the date to its correct type:

```
Water <- Water %>% mutate( Date = as_date(Date, format="%Y-%m-%d" ) )
```

We are asked to consider March and August separately and thus we split the data

```
WaterMarch  <- filter( Water, month(Date) == 3 )
WaterAugust <- filter( Water, month(Date) == 8 )
```

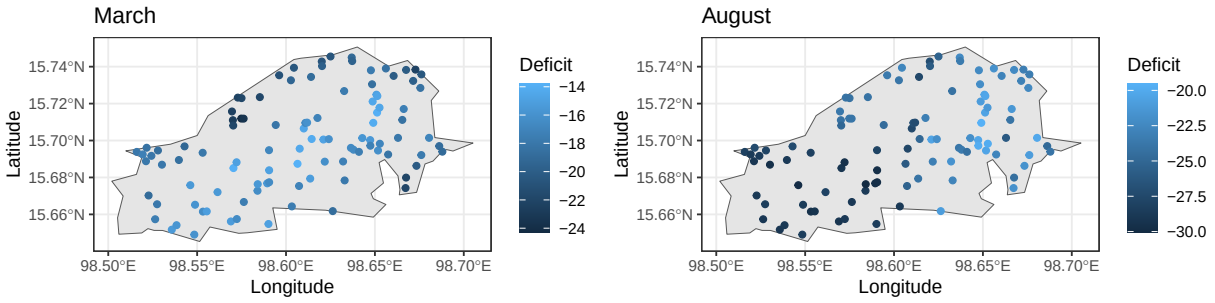
The next step is to calculate the average deficit for each site and month, which we achieve using the following piece of code:

```
WaterMarch <- WaterMarch %>%
  pivot_longer( cols = Site_1:Site_102, names_to = "Site" ) %>%
  group_by( Site ) %>%
  summarise( Average = mean( value ) ) %>%
  full_join( Locations, by=c("Site"="ID") )

WaterAugust <- WaterAugust %>%
  pivot_longer( cols = Site_1:Site_102, names_to = "Site" ) %>%
  group_by( Site ) %>%
  summarise( Average = mean( value ) ) %>%
  full_join( Locations, by=c("Site"="ID") )
```

We are now ready to visualize the average deficits for March and August:

```
March <- ggplot( Tesremos ) + geom_sf() + theme_bw() +
  geom_point( data=WaterMarch, aes(x=Longitude, y=Latitude, color=Average) ) +
 labs( title="March", x="Longitude", y="Latitude", color="Deficit" )
August <- ggplot( Tesremos ) + geom_sf() + theme_bw() +
  geom_point( data=WaterAugust, aes(x=Longitude, y=Latitude, color=Average) ) +
  labs( title="August", x="Longitude", y="Latitude", color="Deficit" )
March + August
```



We find that all sites observed a deficit for both months, with the deficit in March a bit lower than in March, in particular, 14-24% in March as compared to 20-30% in August. When considering March, the highest deficit was observed for the northern part of the Tesremos, while the area with the highest deficit is south-western Tesremos. The deficit seems to change only slightly between spatially close sites and we will explore this in more detail in the next task.

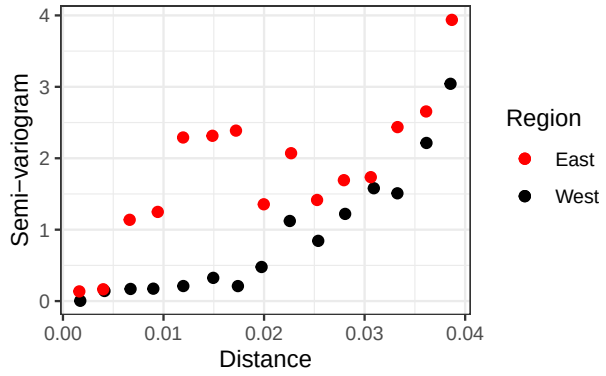
Task ii: Analysis of Spatial dependence We want to explore spatial dependence using the semi-variogram. To estimate the semi-variogram we require the assumptions of (i) **constant mean** and (ii) dependence between sites is fully specified by their spatial distance. The constant mean assumption seems reasonable since we are modelling anomalies, in particular, deviations from the average level in percent.

Some more care is required when considering whether spatial dependence is fully defined by spatial distance. This assumption may not be sensible for August, because the values in the south-west of Tesremos seem to hardly change, while more differences are feasible in the rest of the area. So we may need to consider subregions for analysing the spatial dependence. To investigate this further, we split the region into two equally sized subregions and compare the estimated semi-variograms:

```
DependenceAugustWest <- filter( WaterAugust, Longitude < 98.6 )
coordinates(DependenceAugustWest) <- ~Longitude + Latitude
gamma_hat_West <- variogram( Average~1, DependenceAugustWest, cutoff=0.04 )

DependenceAugustEast <- filter( WaterAugust, Longitude >= 98.6 )
coordinates(DependenceAugustEast) <- ~Longitude + Latitude
gamma_hat_East <- variogram( Average~1, DependenceAugustEast, cutoff=0.04 )

ggplot( gamma_hat_West, aes( x=dist, y=gamma/2 ) ) +
  geom_point( size=2, aes( color="West" ), show.legend = TRUE ) +
  geom_point( data=gamma_hat_East, aes( x=dist, y=gamma/2, color="East" ),
              size=2, show.legend = TRUE ) +
  scale_color_manual( name = "Region", values = c("West"="black", "East"="red") ) +
  theme_bw() + labs( x="Distance", y="Semi-variogram" )
```



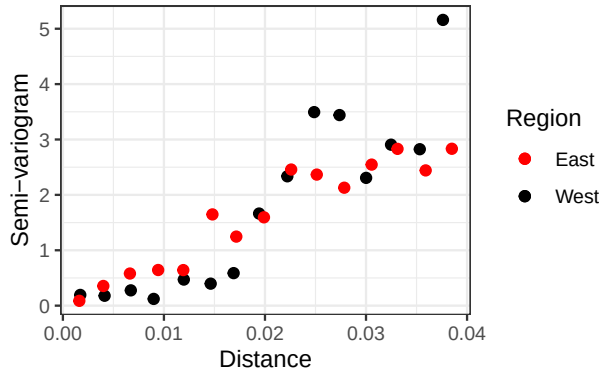
While both semi-variograms indicate that dependence decreases with increasing spatial distance, there is quite a difference between the estimate for low distances. In particular, in Eastern Tesremos, dependence seems to weaken faster will spatial distance in Weastern Tesremos. Furthermore, observations are close to independent when the distance exceeds about 0.15, while we find dependence between sites in Western Tesremos even when they are much farther apart. As such, we conclude that we dependence generally weakens with spatial distance, but the degree of dependence at the same distance varies across Tesremos.

Let's consider the data for March. Here we see that the pattern in the data fits better with the assumption that the dependence is fully specified by the spatial distance. Specifically, the level of differences between spatially close points seems similar across the whole region. We again estimate the semi-variogram for the two subregions we considered before:

```
DependenceMarchWest <- filter( WaterMarch, Longitude < 98.6 )
coordinates(DependenceMarchWest) <- ~Longitude + Latitude
gamma_hat_West <- variogram( Average~1, DependenceMarchWest )

DependenceMarchEast <- filter( WaterMarch, Longitude >= 98.6 )
coordinates(DependenceMarchEast) <- ~Longitude + Latitude
gamma_hat_East <- variogram( Average~1, DependenceMarchEast)

ggplot( gamma_hat_West, aes( x=dist, y=gamma/2 ) ) +
  geom_point( size=2, aes( color="West" ), show.legend = TRUE ) +
  geom_point( data=gamma_hat_East, aes( x=dist, y=gamma/2, color="East" ),
              size=2, show.legend = TRUE ) +
  scale_color_manual( name = "Region", values = c("West"="black", "East"="red") ) +
  theme_bw() + labs( x="Distance", y="Semi-variogram" )
```



The estimated semi-variograms for the two subregions show good agreement. We again see that the value of the semi-variogram increases with increasing distance, which suggests that dependence decreases the further two sites are apart. We also find that sites are close to independent when they are 0.25 distance or more apart.

Remark: I would give credit to a student who splits the data into north and south, and then concludes that again non-stationarity does not hold. However, I will require that there is a proper reason for this approach and students are not just guessing something, or trying to make it work somehow.

Task iii: Predictive maps We want to use inverse distance weighting (IDW) to estimate the deficit across all sites in the region. First we define a function to perform IDW, and which we used in the lectures:

```
IDW <- function( X, S, s_star, p){
  d <- sqrt( (S[,1]-s_star[1])^2 + (S[,2]-s_star[2])^2 )
  w <- d^(-p)
  if( min(d) > 0 )
    return( sum( X * w ) / sum( w ) )
  else
    return( X[d==0] )
}
```

We now span a grid over the region:

```
points_lon <- seq( 98.5, 98.7, length.out=50 )
points_lat <- seq( 15.65, 15.75, length.out=30 )
pixels <- as.matrix( expand.grid( points_lon, points_lat ) )
```

The IDW() function is then applied iteratively to compute the predicted value for each grid cell - we found that a value of $p = 2.3$ for the power parameter works reasonably well:

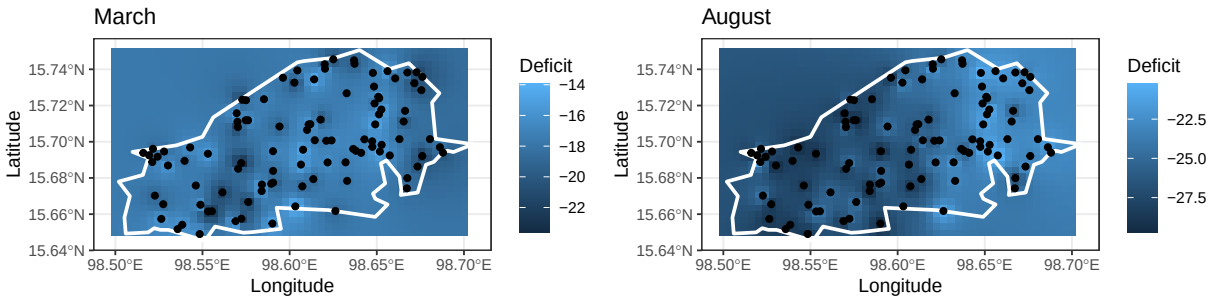
```
predictMarch<- predictAugust <- c()
coord <- cbind( Locations$Longitude, Locations$Latitude )
for( j in 1:length(pixels[,1]) ){
  predictMarch[j] <- IDW( X=WaterMarch$Average, S=coord, s_star=pixels[j,], p=2.3 )
  predictAugust[j] <- IDW( X=WaterAugust$Average, S=coord, s_star=pixels[j,], p=2.3 )
}
```

We can now plot the predicted values for March and August:

```
Predict <- data.frame( "Lon"=pixels[,1], "Lat"=pixels[,2],
                       "March"=predictMarch, "August"=predictAugust )

March <- ggplot( data=Predict ) + theme_bw() +
  geom_raster( aes( x=Lon, y=Lat, fill=March ) ) +
```

```
  geom_sf( data=Tesremos, alpha=0.0, color="white", linewidth=1 ) +
  geom_point( data=Locations, aes(x=Longitude, y=Latitude) ) +
  labs( title="March", x="Longitude", y="Latitude", fill="Deficit" )
August <- ggplot( data=Predict ) + theme_bw() +
  geom_raster( aes( x=Lon, y=Lat, fill=August ) ) +
  geom_sf( data=Tesremos, alpha=0.0, color="white", linewidth=1 ) +
  geom_point( data=Locations, aes(x=Longitude, y=Latitude) ) +
  labs( title="August", x="Longitude", y="Latitude", fill="Deficit" )
March + August
```



There are multiple aspects we need to consider in terms of the reliability of the estimates. On the one hand, we are probably doing a good job since we have a dense network of monitoring sites and values are only changing slowly over space. However, on the other hand, or approach does not account for the fact that spatial dependence changes over space - so $p = 2.3$ may not be a good choice across the whole region.

Remark: I expect that students comment at least briefly on the selection of the power parameter p.

Marking Guidance

A solution as presented above would be awarded 18 out of 20 marks as it really demonstrates a thorough and advanced understanding of the methods introduced in the course. A higher mark can be achieved for some nice discussion of the assumptions and reliability which goes beyond the example above.

The following list outlines some scenarios and the mark they would achieve:

17 marks: An analysis, but with some minor weaknesses in the clarity of the argument.

14-16 marks: All the plots are produced as above, but the reliability was not discussed in enough detail, or there are weaknesses in the discussion of the assumptions for the semi-variogram

12-13 marks: Plots are correctly produced, but assumptions and reliability are poorly discussed.

10-11 marks: Approaches are generally correctly applied (plots for tasks i,ii and iii are produced), but the more challenging aspects were missed in the analysis. For instance, the assumptions of the semi-variogram may be mentioned but not properly discussed and the power parameter was set to an unsuitable value.

8-9 marks: Student demonstrates the ability to reproduce some of the plots from the course (tasks i and iii) with the new data, but shows little understanding of the fundamental ideas and is unable to discuss assumptions and reliability.

0-7 marks: An analysis which demonstrates very little understanding of the concepts on point-referenced data covered in the course.

Minor issues in presentation or an excessive number of plots will be penalized by deducting up to 2 marks, unless it results in the student achieving a failing grade. In extreme cases, such as printing the data files or providing way too many plots, more than 2 marks may be deducted.

Question 3

The government of Utopia has now approached you for help with the analysis of the text messages sent to 971 up till now. Specifically, for the messages with the hashtag #IoSTSHelp, they ask you to:

i. *Visualize the number of messages per island and identify, as precisely as possible, the areas from which the most distress messages were sent.*

ii. *Identify the kind of help required. The four key areas are: lack of food, lack of fuel, lack of shelter and need for medical assistance.*

The government of Utopia is planning to use your analysis approach for any future tropical storms. Specifically, they want to use your approach to identify the areas that need help and the kind of help required, without a government employee having to manually evaluate the text messages. Discuss the strengths and limitations of your analysis approach regarding this aspect.

Rationale

This question requires the student to demonstrate their ability to work with text data, and asks them to visualize lattice and point pattern data. The following aspects are assessed by the different parts of the question

1. Part i is about visualizing lattice and point pattern data. This task can be performed by slightly modifying examples considered in the lecture notes and problem notes. Weaker students may already struggle with this part as we never considered an example combining text and spatial data analysis.
2. Part ii is very open-ended and students will have to dig into the data to get an idea about what they may be looking for, such as keywords - students will need to think slightly beyond what was covered in the course. It is expected that this task will separate students quite a bit, in particular, since they need to clearly describe their approach.
3. The final part of the question requires students to critically evaluate their approach and identify strength and potential weaknesses.

All the messages considered in this question have been generated using ChatGPT.

Example of a first class solution

i. Visualizing the numbers and identifying areas with many messages We start by loading the data:

```
IslesSofara <- read_sf("Isles_of_Sofara_Shapefile/Isles_of_Sofara.shp")
Messages <- read.csv( "Isles_of_Sofara_Messages.csv" )
```

Looking at the data, there are two steps we need to do before the actual. The first is to filter the observations with the hashtag as instructed. We can do this using the function grep():

```
Messages <- Messages[ grep("#IoSTSHelp", Messages$Message), ]
```

We further find that there are messages with missing longitude and latitude, presumably due to people not being able to access their location. We derive the proportion of messages with this feature:

```
mean( is.na( Messages$Longitude) )
```

```
## [1] 0.04395604
```

We find that the exact location is missing for about 4.4% of messages.

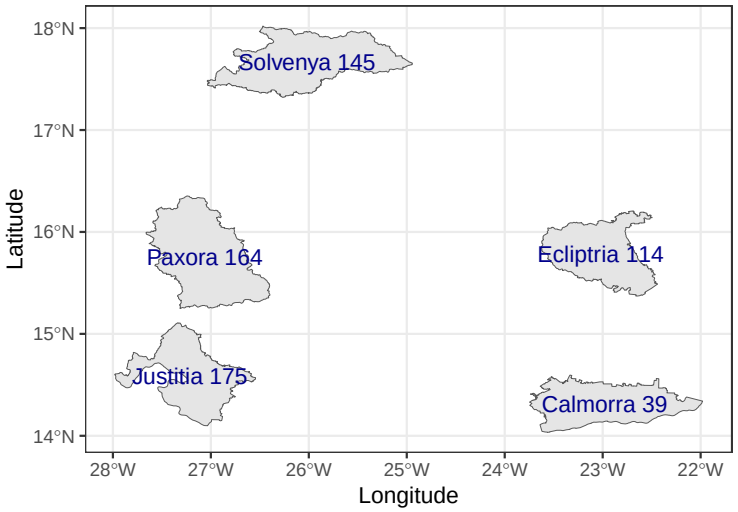
After this initial steps, we are now ready to explore the number of messages per island, which we derive using:

```
Number_Messages <- Messages %>%
  group_by( Island ) %>%
  summarise( Number = n() )
```

One possible way to visualize the data is to plot the islands and add the number of messages to the plot:

```
Islands_centroids <- st_centroid( IslesSofara ) %>%
  full_join(Number_Messages, by=c("Name"="Island") ) %>%
  unite( "Value", c("Name","Number"), sep = " " )

ggplot( IslesSofara ) + geom_sf( ) + theme_bw() +
  geom_sf_text( data=Islands_centroids, aes( label = Value ), color="darkblue" ) +
  labs( x="Longitude", y="Latitude" )
```



Remark: This is a fancier approach than I would expect most students to take. I don't expect students to produce a map - if they provide a bar plot which clearly shows the values for the five islands, this will also be awarded with full marks. If students provide the plot above with only the numbers, than that's also correct. If they produce a plot as in the lecture notes using colour as visual cue, I may deduct marks as the numbers are not clearly visualized.

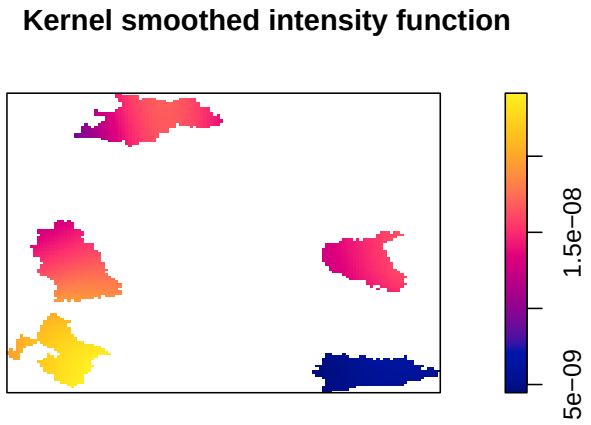
We are now moving on to identifying the areas with a high number of messages. We are dealing with point pattern data and use the kernel smoothed intensity function, and we now need to drop the messages with missing longitude and latitude coordinates:

```
Islands <- IslesSofara %>% st_transform( crs=3857 )

Messages_loc <- Messages %>%
  drop_na( ) %>%
  st_as_sf( coords=c("Longitude","Latitude"), crs=4326 ) %>%
  st_transform( crs=3857 ) %>%
  st_coordinates( )

Messages_ppp <- ppp( Messages_loc[,1], Messages_loc[,2], window = as.owin(Islands) )
```

```
lambdaC <- density.ppp( Messages_ppp, edge=TRUE )
plot( lambdaC, main="Kernel smoothed intensity function" )
```



We identify that the south-eastern part of th island of Justitia recorded the highest intensity. As such, this is an area from which many messages were sent.

Remark: The code above is very similar to that used in the lecture notes for analysing data on tornadoes, and thus it is expected that most students will be able to do this. There is also another method I would accept, which is to consider each island separately and to use the kernel smoothed intensity function or quadrat counting. However, this needs to be presented concisely. An answer which applies quadrat counting to the data as above is unlikely receive full marks, as the interpretation is very hard.

ii. Identifying the help required We start by giving each message an ID - this will help with later deriving the number of people affected:

```
Emergency <- Messages %>%
  mutate( ID = 1:nrow(Messages) ) %>%
  select( ID, Message )
```

To better label the messages, we aim to identify key words for categories (a)-(d). The first step is to split the data into individual words

```
Emergency <- unnest_tokens( Emergency, Word, Message )
```

We now check which words occur the most frequent and we will use them as keywords if they are suitable:

```
TopWords <- Emergency %>%
  anti_join( stop_words, by=c("Word"="word") ) %>%
  count( Word, sort=TRUE ) %>%
  slice_head( n=20 )
TopWords
```

```
##          Word   n
## 1   iostshelp 637
## 2        fuel 317
## 3     medical 258
## 4     shelter 209
```

```
## 5        left 202
## 6        food 169
## 7        care 140
## 8       storm 138
## 9    urgently 132
## 10    running 126
## 11 completely 118
## 12      we're 117
## 13   supplies 110
## 14        eat  79
## 15  exhausted  79
## 16     hungry  79
## 17  attention  74
## 18      badly  74
## 19    injured  74
## 20    damaged  72
```

One may argue that the following are good keywords for the different categories:

```
Food <- c( "food", "hungry", "eat" )
Medical <- c( "injured", "care", "medical" )
Shelter <- c("shelter", "damaged" )
Fuel <- c( "fuel" )
```

Our approach is now to check whether the keywords appear in the message. If they apepar in the message, we will conclude that this person requires that kind of help:

```
Emergency %>%
  filter( Word %in% Medical ) %>%
  summarise( Type = "Medical help", Number = length(unique(ID) ) )
```

```
##           Type Number
## 1 Medical help    329
```

```
Emergency %>%
  filter( Word %in% Food ) %>%
  summarise( Type = "Lack of food", Number = length(unique(ID) ) )
```

```
##           Type Number
## 1 Lack of food    248
```

```
Emergency %>%
  filter( Word %in% Shelter ) %>%
  summarise( Type = "Lack of shelter", Number = length(unique(ID) ) )
```

```
##              Type Number
## 1 Lack of shelter    209
```

```
Emergency %>%
  filter( Word %in% Fuel ) %>%
  summarise( Type = "Lack of fuel", Number = length(unique(ID) ) )
```

```
##           Type Number
## 1 Lack of fuel    317
```

Our approach identifies that 329 people required medical help, 248 messages mentioned a lack of food, a total of 209 messages reported a lack of shelter, and 317 people reported a lack of fuel.

Remark: I expect that many students will only use the function grep(). This is a possible approach, but it

won't receive high marks as it does not engage wit the data in enough detail.

iii. Discussion of the strengths and weaknesses The analysis in part i. can be applied to any future set of messages to quickly identify the areas affected by the tropical storm, as long as, sufficient people are able to send their location information - in the absence of that information we cannot say much.

Our approach for identifying the help required needs some further testing. There is also a risk that messages may be mislabelled, since we only focus on certain keywords - we could look further down the list to improve the approach further. We are also unable to distinguish messages which, for instance. mention fuel, but report sufficient fuel instead of a lack. We believe that the government may have some data on the help they provided and thus the accurracy of our estimates may be assessed.

Marking Guidance

A solution as presented above would be awarded 13 out of 15 marks. A higher mark can be achieved for some nice discussion on iii. or a more advanced approach for ii.

The following list outlines some scenarios and the mark they would achieve:

12 marks: An analysis, but with some minor weaknesses in the clarity of the argument.

11 marks: Results are as above, but the discussion lacks some depth OR proportion of missingess was not stated.

9-10 marks: General approach is correct, but the function grep() is simply used for ii. without much fine-tuning.

8 marks: Choice of plots for identifying areas with high numbers is not optimal and the approach in ii. is very basic.

6-7 marks: Some of the important information is retrieved, but there overall approach misses some key aspects

0-5 marks: An analysis which demonstrates very little understanding of point pattern data and text data analysis.

Minor issues in presentation or an excessive number of plots will be penalized by deducting up to 2 marks, unless it results in the student achieving a failing grade. In extreme cases, such as printing the data files or providing way too many plots, more than 2 marks may be deducted.