The Effect of the Unexpected Change in U.S. Housing Market on Society

Computational Analysis of Communication

Bennur Kaya

2023-01-23

Introduction

The Covid-19 pandemic has affected the lives of millions of people around the globe. It showed how fragile human health is and that there are lacks in the health system. On the other hand, it has enabled us to quickly adapt technology to our daily life and gave the chance to maintain our business life from our home. It prepared the infrastructure for countless technological innovations and created many different occupations. Also, it showed the countries are more dependent to each other for raw materials, food and fuel because of globalization. These difficulties strain the economy, causing inflation to rise.

In “The impact of the COVID-19 pandemic on the demand for density: Evidence from the U.S. housing market” study carried out in April 2021 by Sitian Liu from Queen’s University, Canada and Yichen Su from Federal Reserve Bank of Dallas, they worked on the impact of the Covid-19 pandemic on the location demand for housing. They find that the pandemic has led to a shift in housing demand away from neighborhoods with high population density. They observed that neighborhoods with a greater share of telework-compatible jobs, more restaurants per capita, and higher pre-Covid-19 home prices witnessed a relative decline in home and rental prices and a relative increase in home inventory. They conclude as the smaller divergence in home price growth relative to that in home inventory and rent growth between central cities and suburbs suggest the market anticipates that future demand for central locations could bound back to some degree in the long run. However, there have been unexpected rent increases in metropolitan cities such as New York, London and Berlin since the time this study was published. As mentioned in The New York Times article “Why the Rent Is So High” by James Barron in August 2022, the typical New York City household spent about 20 percent of its income on rent in 1965, however, that number was close to 35 percent in 2022. This paper argues that the situation shows moving from neighborhoods with high population density is not just due to preferring to pay lower rent because of the decrease in the need to live close to workplaces, but also due to the financial difficulties caused by the unexpected price increase in rents after Covid-19. In such cases, people may move to the suburbs even if they want to stay in the city.

Rent is an incredibly big decision factor to move from city centers to the suburbs. In addition to (Sitian & Yichen, 2022), further analysis is conducted by scraping different kind of data to understand how the rent increase had an impact on people in the last one year and to better define people's thoughts. In the first research question, two YouTube videos which are "Why Rent In NYC Is Out Of Control Right Now" by CNBC Make it and "Why It's So Expensive To Live In The U.S." by CNBC focusing on the real estate related issues in New York and across the United States are examined. Sentiment analysis is made by using the subtitles. In the second research question, topic modeling is performed for various New York Times news focusing mainly on home and rental prices for the several months from February 2021 to August 2022. Topic modeling is done to observe what are topics or subjects that is mentioned related to New York rental difficulties, what tokens or words are most important in these topics, and to what extent it is about the identified topics, and this can be a mix (Corona, inflation etc.). In the last research question, Twitter web data scraping is done by focusing on certain keywords and it is aimed to observe whether people have a negative or positive feelings on this issue.

Literature Review

Overview of the Previous Works

After the COVID-19 pandemic, many office workers started working from home, online fast-paced consumption became the center of our lives, people created the time and space to cook at home instead of eating out, and real-life communication saw a sharp decline. Thus, the attractiveness of dense neighborhoods and big cities where business and consumption opportunities are spatially concentrated have decreased. Sitian Liu and Yichen Su conducted an analysis to reveal important aspects to understand the impact of pandemic in daily life and the possible indicators of the shift in housing demand from central cities and dense neighborhoods to the suburbs and neighborhoods with lower population density in "The impact of the COVID-19 outbreak on density demand: Evidence from the US housing market" article. They use various local housing indicators such as inventory, house price and rent to track spatial differences in change in housing demand and document a strong divergence in growth trajectories in these indicators in central and dense neighborhoods compared to suburban and remote neighborhoods. They conduct a a regression analysis with different local outcomes such as inventory, home prices (HPIs), and rent prices. Some of their findings are that the COVID-19 pandemic reduced housing demand in dense neighborhoods, and this reduced demand for density is driven by less need of living near jobs. Also, the shift from density is also due to the declining value of access to amenities. Hereby, they conclude that housing demand shifted from expensive locations to cheaper locations and shift in housing demand occurred mainly within cities toward suburbs. As a result of reduced demand for central cities and dense neighborhoods, they said neighborhoods with a greater share of telework-compatible jobs, more restaurants per capita, and higher pre-COVID-19 home prices witnessed a relative decline in home and rental prices, and a relative increase in inventory. However, contrary to what was stated in the article, there was a sharp increase in house and rent prices in the U.S. and in almost all major cities of the world after April 2021, when this article was first published, due to both inflation and corona.

As stated for the name of Douglas Elliman which is a technology-enabled comprehensive real estate services housing brokerage firm, the median rental price has increased by 24.7% in one year (Samuel, 2022). The net effective average rent in Manhattan exceeded 5000 Dollars, and the net effective median rent across New York exceeded 4000 Dollars for the first time. Also, they stated that the net effective median rent in Brooklyn reached a new high for the third consecutive month. Moreover, for Northwest Queens, both average and median rent and net effective average and net effective median rent set new records, and the market share of landlord concessions fell to its lowest level in seven years. Also, this report verifies that there was a clear decrease in the number of new leases. It was -55.3% for the studio apartments, -4.4% for the 1-Bedroom and -9.4% for the 3-bedroom rentals. Hence, this report may indicate that the reduced demand for density is not only driven partially by the diminished need for living close to telework-compatible jobs and the declining value of access to consumption amenities, but also because of unexpected increment in house and rental prices.

Rent prices have a strong relationship with economic factors in addition to the structural and environmental characteristics of housing stocks. Across the United States, rents are higher than ever and there is no single definitive answer why rent is so expensive. Reasons why rent is so high range from general inflation to fewer housing units being constructed in recent years to plain, old high demand for housing and much more. On the other hand, after the first COVID-19 cases, changes in both housing preferences and economic structure significantly affected the rental housing market due to pandemic conditions. From June 2021 to June 2022, the year-over-year inflation rate for all items was 9% (Cox, 2022) in the U.S. The inflation rate of rents, housing fuel and utilities, home appliances and household repairs, among other things directly and indirectly could contribute to why rent is so expensive. Another longer-term factor is the severe decline in new home construction for a decade after the housing market crash in the early 2000 and this reduction in new available housing has resulted in reduced supply of available homes to both buy and rent. In addition, this is a problem observed not only in the U.S. but around the world. Unexpected and sudden situations such as wars and epidemics also affect the economy and show their impact on a global scale. According to a recent analysis, the rise in rents for apartments in Germany has accelerated again after a phase of relatively moderate growth. According to data from the German Economic Institute (IW), rent offers in Düsseldorf (5.9 percent), Leipzig (7.8 percent) and Berlin (8.3 percent) increased in particular. In the capital, the increase was almost twice as high as the average increase in asking rents in the third quarter of the past three years (The Local, 2022). As another example, the imbalance between demand and supply continues to spread fear on the private rental sector for London as well. According to the latest data from Rightmove's Rental Price Tracker, it is found the average London rent was now £2,257 per month and it is considered as very high for any UK city (Jessel, 2022). Also, the past three months have seen an annual rise in asking prices of 15.8 percent. All in all, the unavoidable increase in rents as a result of the pandemic, inflation, the lack in the number of adequate housing and the rapid increase in the population is a global problem today.

Nevertheless, one of the most important factors that negatively affect human psychology is financial difficulties. When people give a large percentage of their monthly salary to housing, they need to compromise different priorities in their lives. For this, they may prefer to find a more affordable house or a region with more affordable houses. As mentioned in the above quantitative studies, many of them prove that the real estate problem in cities. How this unexpected price increase affects the society can be observed in many different ways. As an example, it is possible to learn about people's awareness level and opinions about this situation through street interviews. In terms of house and rental price, some people might be affected by this problem much more than others and have difficulties. But perhaps this does not cause such a big problem for many other people living in big cities because they are in a financially better condition. Some may think that this result is irreversible, and some may have more constructive thoughts about it. In addition to street interviews, different news sources that observe and inform the public can also be a guide for the opinion of the society. They can also include interviews with experts and present this situation to the public within a certain framework. Moreover, another method that can be done to learn the thoughts of people directly is to examine the comments on social media platforms where individual ideas are shared. Many people share their opinions on current issues and raise awareness via Twitter. As a result of the analysis of these tweets with certain methods, the opinion, tendency and feelings of the society on a subject can be understood to some extent.

Research Questions

Many recent studies focused on quantitative research on the rental price change in housing market and inflation by drawing attention to numerical results and comparing the old with the new. In addition to quantitative research, in this paper, it is aimed to observe the reaction, feelings and positive/negative thoughts of the society. Although it is often emphasized how abnormal this situation is compared to the past, it is important to see what its counterpart and reflections are in the society. In this study, there exist three different research questions which can be solve with three different data sources. It is aimed to observe various points of view through these different sources.

For the first research question, two YouTube videos which are *Why Rent In NYC Is Out Of Control Right Now* by CNBC Make and *Why It's So Expensive To Live In The U.S.* by CNBC are examined. Both videos have interviews with real locals of that distinct. Sentiment Analysis technique, which is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the attitude towards a particular topic etc. positive, negative, or neutral, is implemented. In this part, NRC sentiment dictionary is used to calculate the presence of eight different emotions and their corresponding valence in a text file. These emotions are "anger", "anticipation", "disgust", "fear", "joy", "sadness", "surprise", "trust", "negative", "positive". These emotions reflect the thoughts and feelings of the people about the house and rent prices in the videos. These videos are posted on Aug 2022 and May 2022 respectively. All in all, the research question can be defined as "*What emotions are expressed by the people in the videos due to the increase in housing market prices in the USA?".*

For the second research question, topic modeling technique is used. Topic model is a type of statistical model that can be used for discovering the abstract topics that occur in a collection of documents. In this part, the data set contain information about 10 different news from the New York Times which is about New York and real estate. On average there exit two to three different articles about this subject every month. News have been selected in a way not two articles from the same month is chosen. The time span is between February 2021 and July 2022. As a result of this analysis, it is aimed to know what are topics or subjects that is mentioned related to NYC rent, what tokens or words are most important in these topics, and to what extent it is about the identified topics and these can be a mix of various factors (pandemic, inflation etc.). There are number of algorithms to extract topics from a collection of texts, but the Latent Dirichlet Allocation is one of the most popular algorithms because it gives efficient results in highly interpretable topics. Interpretability of topics is an important feature of a topic model, since we do not only want to find statistically relevant groupings of words, but we also want to be able to label the identified topics with a topic name that others can relate to. There are certain steps that needs to be followed before developing a topic model. These are tokenization, transforming all characters to lowercase, removing punctuation and special characters, removing stop-words, term unification as lemmatizing or stemming. This analysis can give us information about the main topics of the news about the housing market and whether there is a relationship between the words under these topics. This can also lead to a case-specific analysis by focusing on different topics in the future. All in all, the second research question is "*What are the topics of the news? Can you see any semantic relation within the topic groups and how would you interpret the results?"*

Third research question seeks an answer for if the unexpected change in the US housing market a positive or negative impact on society by have using Twitter data. It aims to see the most frequently used words in individual thoughts and what it indicates. At the same time, it provides a chance to see whether the feelings are positive, negative or neutral. For this question, the focus is New York and California since New York City is the largest city in the US and is known internationally as a hub of culture, entertainment and education, and the economy of the State of California is the largest in the United States, with a $3.63 trillion gross state product (GSP) as of 2022. It offers the opportunity to observe the general opinion. So, the last research question is "*Are the unexpected changes in the US housing market having currently a positive or negative impact on society? What are the most frequently used words in individual thoughts on Twitter and what does it indicate?*"

Description of the Methods and Discussion of the Results

```
#Libraries

library(quanteda) #for managing and analyzing textual data
library(tibble) #provides a 'tbl_df' class (the 'tibble') with stricter
checking and better formatting than the traditional data frame
library(magrittr) #decrease development time and improve readability and
maintainability of code
library(dplyr) # fast, consistent tool for working with data frame like
objects, both in memory and out of memory.
library(stringr) # provides a cohesive set of functions designed to make
working with strings as easy as possible
library(rvest) #helps you scrape (or harvest) data from web pages
library(tidyverse) #set of packages that work in harmony because they share
common data representations and 'API' design
library(LDAvis) #tools to create an interactive web-based visualization of a
topic model that has been fit to a corpus of text data using Latent Dirichlet
Allocation
```

First research question: What emotions are expressed by the people in the videos due to the increase in housing market prices in the USA?

For the first research question, the subtitles of the YouTube videos which are Why Rent in NYC Is Out Of Control Right Now by CNBC Make and Why It's So Expensive To Live In The U.S. by CNBC are scraped by using Python. Data preparation, data cleaning and data analysis are done by using R programming.

```
# Why It’s So Expensive To Live In The U.S. by CNBC
library(readr) # provide a fast and friendly way to read rectangular data

# Importing the text file
video1 <- read_csv("youtubevideoscripting/video1.txt",
    col_names = FALSE)
```

```
# head(video1) # to see the first 6 rows
# summary(video1) # to have a detailed summary of the data
# dim(video1) # checking the dimension of the data set

# Preparing the data set for the first video
colnames(video1)[1] <- "Subtitle"

Name <- rep(c("Why It’s So Expensive To Live In The U.S. by CNBC"), each =
dim(video1)[1])
Name_video1 <- data.frame(Name)

first_video <- cbind(video1, Name_video1)
head(first_video, 2)
```

```
## Subtitle
## 1 Some big numbers coming out
## 2 of the housing market today,...and they're not looking

## Name
## 1 Why It’s So Expensive To Live In The U.S. by CNBC
## 2 Why It’s So Expensive To Live In The U.S. by CNBC
```

After importing the data pulled by Python into R, looking at the data set, it is seen that it has only one column. These subtitles are divided into different lines in the data set as they are divided into different lines on subtitles in YouTube. The subtitles of this video, which is approximately 46 minutes long, are represented on 864 different rows in the data set. In the data preparation process, firstly, the column name X1 is changed to “Subtitle”. For the comparative analysis in the following sections, the “Name” column is created.

```
# Why Rent In NYC Is Out Of Control Right Now - by CNBC Make it

# Importing the text file
video2 <- read_delim("youtubevideoscripting/video2.txt",
    delim = "\t", escape_double = FALSE,
    col_names = FALSE, trim_ws = TRUE)

# head(video2)
# summary(video2)
# dim(video2)

# Preparing the data set for the first video
colnames(video2)[1] <- "Subtitle"

Name <- rep(c("Why Rent In NYC Is Out Of Control Right Now - by CNBC Make
it"), each = dim(video2)[1])
Name_video2 <- data.frame(Name)

second_video <- cbind(video2, Name_video2)
head(second_video)
```

```
## Subtitle
## 1 We saw the number and it was just insane....It's literally doubling what
we're paying right now....We never expected that much....I mean, that was
just a real shock to us....This apartment is not worth $3,500....There is
nothing worth $3,500 in New York City....Rents in New York City are at an
all-time high....In June 2022, citywide median asking rent reached
## 2 $3,500....That's about a 35% increase from 2021....The median asking
rent is $4,100 in Manhattan, $3,200

## Name
## 1 Why Rent In NYC Is Out Of Control Right Now - by CNBC Make it
## 2 Why Rent In NYC Is Out Of Control Right Now - by CNBC Make it
```

Similarly, the second video's subtitles were created and downloaded to R. The data set containing the subtitles of this video, which is approximately 18 minutes long, contains 142 rows. In a same way, the column name X1 is changed to "Subtitle" and "Name" column is created.

After binding the data sets video1 and video2 , "Id" column is created. Now, the data set has 1006 rows with 3 columns as id, Subtitle and Name. The data set is ready for further analysis.

```
# Binding the data sets
all <- rbind(first_video, second_video)

id <- seq(1:dim(all)[1])

all <- cbind(id, all)

dim(all)
```

```
## [1] 1006    3
```

```
# rename user id and add row_id
 all <- all %>%
   mutate(user_id = id) %>%
   select(-id) %>%
   rowid_to_column("id")

subtitle_corpus <- corpus(all,
                         docid_field = "id",
                         text_field = "Subtitle")
```

Subtitle corpus is created. *Quanteda* package is imported which is a framework for quantitative text analysis to provide functionality for corpus management, creating and manipulating tokens and n-grams, exploring keywords in context and more. By removing punctuations, numbers, symbols and urls, tokenization is done.

```
# now we can tokenize again

subtitle_tokens <- quanteda::tokens(subtitle_corpus,
                                 remove_punct = TRUE,   # removes
punctuations
                                 remove_numbers = TRUE, # removes numbers
                                 remove_symbols = TRUE, # removes symbols
(also: emojis)
                                 remove_url = TRUE)     # removes urls

head(subtitle_tokens, 2)
```

```
## Tokens consisting of 6 documents and 2 docvars.
## 1 :
## [1] "Some"    "big"     "numbers" "coming"  "out"
##
## 2 :
## [1] "of"      "the"     "housing" "market"  "today"   "and"     "they're"
## [8] "not"     "looking"
```

Later, capital letters in tokens are converted to lowercase and stop words are removed.

```
# lower key tokens
subtitle_tokens_lk <- subtitle_tokens %>% tokens_tolower()

# remove stop words
subtitle_tokens_nosw <- subtitle_tokens_lk %>%
  tokens_remove(stopwords("english"))

head(subtitle_tokens_nosw, 2)
```

```
## Tokens consisting of 6 documents and 2 docvars.
## 1 :
## [1] "big"     "numbers" "coming"
##
## 2 :
## [1] "housing" "market"  "today"   "looking"
```

Then, stemming is applied to tweet tokens.

```
# STEMMING
subtitle_tokens_nosw %>% tokens_wordstem() %>%
    head(2)
```

```
## Tokens consisting of 6 documents and 2 docvars.
## 1 :
## [1] "big"    "number" "come"
##
## 2 :
## [1] "hous"   "market" "today"  "look"
```

By using *udpipe_download_model* function, an UDPipe model provided is downloaded by the UDPipe community for a specific language of choice. Ready-made models for 52 languages trained are provided via this function. Since the language in the videos are English, language parameter is set as "english".

udpipe function is used for tokenising, lemmatising, tagging and dependency parsing of raw text in TIF format. After lemmatising, the names of the videos are introduced as screen name in order to make a comparison in the further part.

```
library(udpipe)

ud_model_en <- udpipe_download_model(language = "english")

udpipe_lemmas <- udpipe(subtitle_corpus, object = ud_model_en) # Tokenising,
Lemmatising, Tagging and Dependency Parsing of raw text in TIF format

# hack in screen_name
udpipe_lemmas <-
  udpipe_lemmas %>%
  mutate(
    screen_name = if_else(doc_id <= 864 , "Why It’s So Expensive To Live In
The U.S.", "Why Rent In NYC Is Out Of Control Right Now")
  )

docs <- udpipe_lemmas %>%
  group_by(doc_id) %>%
  summarize(text = paste(lemma, collapse = " "), screen_name = screen_name)
%>%
  distinct()

lemma_tokens_nosw <- docs %>%
  corpus() %>%
  quanteda::tokens(remove_punct = TRUE,
                   remove_numbers = TRUE,
                   remove_symbols = TRUE,
                   remove_separators = TRUE,
                   remove_url = TRUE) %>%
  tokens_tolower() %>%
  tokens_remove(stopwords("english"))

library(quanteda.textstats)
```

```
lemma_tokens_nosw %>%
  textstat_collocations() %>%
  arrange(desc(count))
```

```
##               collocation count count_nested length    lambda         z
## 58               new york    35            0      2 9.3124234  6.482797
## 1               york city    18            0      2 5.0175782 12.959155
## 4              home price    15            0      2 3.6459243 10.712612
## 3          housing bubble    12            0      2 4.9245804 10.771542
## 2               right now    11            0      2 6.8202541 10.929531
## 5           single family    11            0      2 7.4583461 10.648906
## 8          housing market     9            0      2 3.4107653  8.520151
## 6             real estate     7            0      2 7.6371973  9.211463
```

As a result of the chunk, lemma tokens are created. When we look at result of these tokens consisting of two words, as can be expected, the most seen tokens are "new york", "york city" and "home price" respectively.

Then, kwic function returns a list of a keyword supplied by the user in its immediate context, identifying the source text and the word index number within the source text. It is used and keyword-in-context has 78 matches.

```
kwic(lemma_tokens_nosw, pattern = phrase("rent"))
```

```
## Keyword-in-context with 78 matches.
##   [103, 3]          time homebuyer | rent | home back
##    [15, 4]        wildfire go away | rent | growth
```

Later, *tokens_ngrams* is used to create a set of n-grams (tokens in sequence) from already tokenized text objects. *n* parameter is an integer vector specifying the number of elements to be concatenated in each n-gram and it is chosen as 2, 3, 4 and 5 in a sequence.

```
#N-GRAMS
tokens_ngrams <- lemma_tokens_nosw %>%
  tokens_ngrams(n = 2:5)
```

Then, a sparse document-feature matrix is constructed.

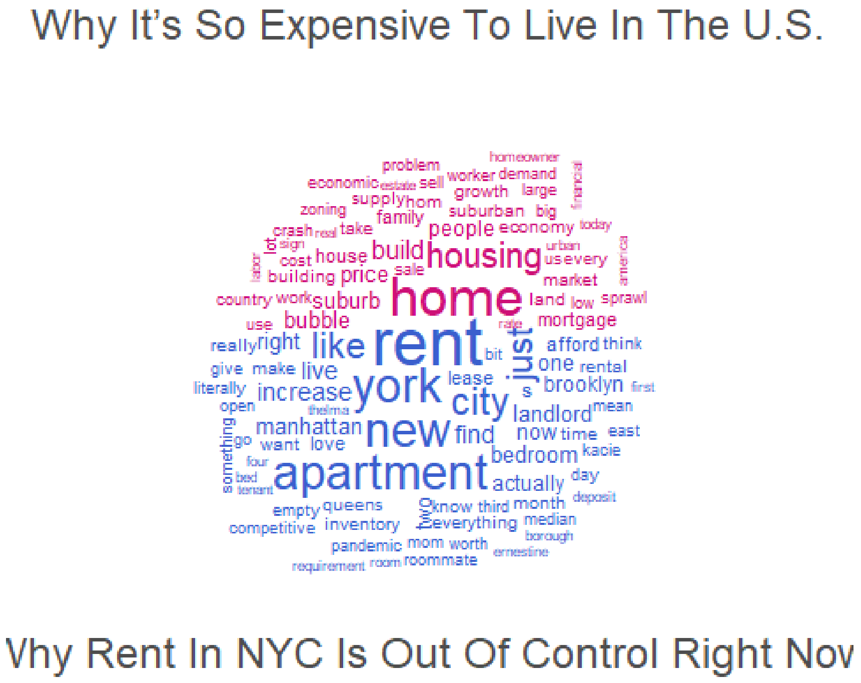
```
dfm <- dfm(lemma_tokens_nosw)

topfeatures(dfm, groups = screen_name)
```

```
## $`Why It’s So Expensive To Live In The U.S.`
##    home housing  people   price     can     see  market      go     get
city
##      83      56      49      42      40      37      36      34      33
32
##
```

```
## $`Why Rent In NYC Is Out Of Control Right Now`
##       rent        new       city       york apartment       like       just
one
##         48         36         31         29         26         26         23
17
##         go        see
##         16         16
```

Looking at the most frequently occurring features in a dfm, “home”, “housing” and “people” are the most frequently used words respectively in the first video. Also, “rent”, “new” and “city” are the most frequently used words respectively in the second video. It totally fits with the context of the videos since both are about renting problem of people in the U.S. or New York City.

```
# COMPARE

# grouped DFM
dfm_grouped <- dfm_group(dfm, groups = screen_name)

# wordcloud
library(quanteda.textplots)
textplot_wordcloud(dfm_grouped,
                   min_size = 0.65,
                   max_size = 4,
                   max_words = 110,
                   comparison = TRUE,
                   color = c("deeppink3", "royalblue3"))
```



Next, dfm object is plotted as a word cloud. A word cloud is a collection of words depicted in different sizes. The bigger and bolder the word appears, the more often it’s mentioned within a given text and the more important it is. As can be seen in the above best features results, the most used words are the biggest and bold ones. On the other hand, in the Why It’s So Expensive To Live In The U.S. video, “bubble” word also takes place. A housing bubble, also sometimes referred to as a “real estate bubble,” occurs when the price of housing rises at a rapid pace, driven by an increase in demand, limited supply and emotional buying and it is highly mentioned in the video. The words such as demand, mortgage, land, lot, economy and problem summarize the topic of the video as well. They emphasize that the price per single land lot is up 11% compared to last year, demand got so high for the houses, and they need to build more housing to meet natural growth in the economy. In the second video which is Why Rent In NYC Is Out Of Control Right Now, in addition to the most used words in the word cloud,” roommate”, “competitive”, “pandemic”, “deposit” and “requirement” are the words in word cloud. In the video, it is mentioned that at the beginning of the pandemic, as people left the city, the rents that fell at an unbelievable rate increased at an unexpectedly high rate. Aside from living alone in a rental house without the need for a roommate, high rents and deposits, the documentation requirements to prove your credibility, and the competitive environment have made it very difficult to find the most suitable home.

By checking lexical diversity, lexical richness of the text is examined. Type-Token Ratio (TTR) divides the number of unique tokens through all tokens within a corpus and it is useful for analyzing speakers’ linguistic skill and the complexity of ideas expressed in the videos. A high TTR indicates a high degree of lexical variation. It seems “Why Rent In NYC Is Out Of Control Right Now” has slightly higher Type-Token Ratio.

```
# Lexical Diversity
lexdiv <- textstat_lexdiv(dfm_grouped)
lexdiv
```

```
##                                      document       TTR
## 1   Why It’s So Expensive To Live In The U.S. 0.3082073
## 2 Why Rent In NYC Is Out Of Control Right Now 0.3814990
```

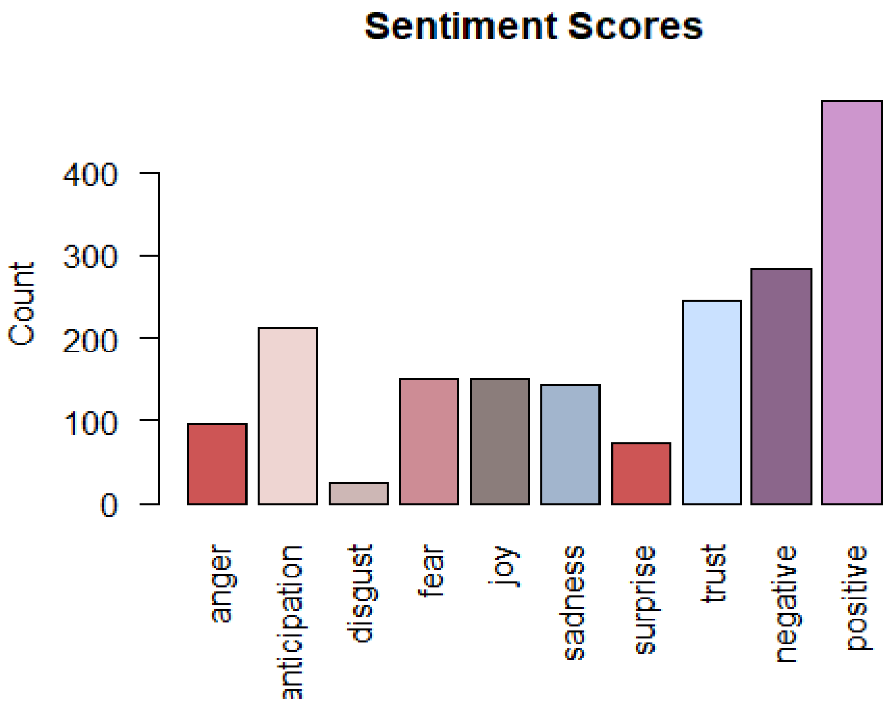
To be able to obtain the sentiment scores, get_nrc_sentiment function is used. It calls NRC sentiment dictionary to calculate the presence of eight different emotions such as anger, surprise or trust and their corresponding valence in a text file.

```
library(syuzhet) # extracts sentiment and sentiment-derived plot arcs from
text using a variety of sentiment dictionaries

#Obtain sentiment scores
s <- get_nrc_sentiment(subtitle_corpus) #calls NRC sentiment dictionary to
```

```
calculate the presence of eight different emotions and their corresponding
valence in a text file.
```

Finally, sentiment scores are represented in a bar plot. As it can be seen from the bar plot of Sentiment Scores of two videos below, the sentences used by the speakers in the videos were mostly positive. However, negative thoughts were the second emotion to come after positive thoughts. On the other hand, according to the bar chart, trust and anticipation are the other most observed sentiments. This may indicate that even if people have negative thoughts about the situation they are in, they have faith that the situation will get better again. Maybe that's why positive emotion is found intensely even though situation has some downsides. However, it seems that sentiment analysis may have received some parts positively, as the speakers talked about the short-term real estate market relief at the beginning of the pandemic. Finally, the graph also shows emotions such as anger, fear and sadness are also present.

```
barplot(colSums(s),
        las = 2,
        col =
c("indianred3","mistyrose2","mistyrose3","lightpink3","mistyrose4","lightstee
lblue3","indianred3","lightsteelblue1", "plum4","plum3"),
        ylab = 'Count',
        main = 'Sentiment Scores')
```



Second research question: What are the topics of the news? Can you see any semantic relation within the topic groups and how would you interpret the results?

For the second research question, topic modeling technique is used. News about New York and real estate were selected from the New York Times newspaper, not more than one news per month. The merged data set consist of 10 different news.

```
nytimes1 <- read_html("https://www.nytimes.com/2021/02/04/realestate/how-the-
pandemic-blew-up-rents.html")

nytimes_items1 <- nytimes1 %>% html_nodes(".css-at9mc1")

nytimes_new1 <- nytimes_items1 %>% html_text()

nytimes_news1 <- as.data.frame(nytimes_new1)

# Preparing the data set
colnames(nytimes_news1)[1] <- "Text"

Name <- rep(c("How the Pandemic Blew Up Rents"), each =
dim(nytimes_news1)[1])
Name_news1 <- data.frame(Name)

NYTnews1 <- cbind(nytimes_news1, Name_news1)



nytimes2 <-
read_html("https://www.nytimes.com/2021/03/20/realestate/landlords-covid-
eviction-rent.html")

nytimes_items2 <- nytimes2 %>% html_nodes(".css-at9mc1")

nytimes_new2 <- nytimes_items2 %>% html_text()

nytimes_news2 <- as.data.frame(nytimes_new2)

# Preparing the data set
colnames(nytimes_news2)[1] <- "Text"

Name <- rep(c("My Tenant Won’t Pay, but Won’t Go. What Can I Do?"), each =
dim(nytimes_news2)[1])
Name_news2 <- data.frame(Name)

NYTnews2 <- cbind(nytimes_news2, Name_news2)


nytimes3 <- read_html("https://www.nytimes.com/2021/04/01/realestate/the-
brooklyn-manhattan-rent-gap-is-shrinking-
```

```r
fast.html#:~:text=After%202015%2C%20rents%20in%20both,Miller%20began%20tracki
ng%20it%202008.")

nytimes_items3 <- nytimes3 %>% html_nodes(".css-at9mc1")

nytimes_new3 <- nytimes_items3 %>% html_text()

nytimes_news3 <- as.data.frame(nytimes_new3)

# Preparing the data set
colnames(nytimes_news3)[1] <- "Text"

Name <- rep(c("The Brooklyn-Manhattan Rent Gap Is Shrinking Fast"), each =
dim(nytimes_news3)[1])
Name_news3 <- data.frame(Name)

NYTnews3 <- cbind(nytimes_news3, Name_news3)


nytimes4 <- read_html("https://www.nytimes.com/2021/05/05/nyregion/nyc-rent-
stabilization-vote.html")

nytimes_items4 <- nytimes4 %>% html_nodes(".css-at9mc1")

nytimes_new4 <- nytimes_items4 %>% html_text()

nytimes_news4 <- as.data.frame(nytimes_new4)

# Preparing the data set
colnames(nytimes_news4)[1] <- "Text"

Name <- rep(c("New York City May Freeze Rent Again for More Than 2 Million
Tenants"), each = dim(nytimes_news4)[1])
Name_news4 <- data.frame(Name)

NYTnews4 <- cbind(nytimes_news4, Name_news4)


nytimes5 <- read_html("https://www.nytimes.com/2021/10/07/nyregion/a-
pandemic-story-brooklyn-tenants-who-stopped-paying-rent.html")

nytimes_items5 <- nytimes5 %>% html_nodes(".css-at9mc1")

nytimes_new5 <- nytimes_items5 %>% html_text()

nytimes_news5 <- as.data.frame(nytimes_new5)

# Preparing the data set
colnames(nytimes_news5)[1] <- "Text"
```

```r
Name <- rep(c("A Pandemic Story: Brooklyn Tenants Who Stopped Paying Rent"),
each = dim(nytimes_news5)[1])
Name_news5 <- data.frame(Name)

NYTnews5 <- cbind(nytimes_news5, Name_news5)


nytimes6 <- read_html("https://www.nytimes.com/2022/09/18/realestate/nyc-
development-rentals-condos-covid.html?searchResultPosition=5")

nytimes_items6 <- nytimes6 %>% html_nodes(".css-at9mc1")

nytimes_new6 <- nytimes_items6 %>% html_text()

nytimes_news6 <- as.data.frame(nytimes_new6)

# Preparing the data set
colnames(nytimes_news6)[1] <- "Text"

Name <- rep(c("In New York City, the Demand for New Developments Is Bouncing
Back"), each = dim(nytimes_news6)[1])
Name_news6 <- data.frame(Name)

NYTnews6 <- cbind(nytimes_news6, Name_news6)


nytimes7 <- read_html("https://www.nytimes.com/2022/06/22/nyregion/rent-
regulation-new-
york.html#:~:text=For%20an%20apartment%20to%20be,units%20are%20decontrolled%2
0once%20vacant.")

nytimes_items7 <- nytimes7 %>% html_nodes(".css-at9mc1")

nytimes_new7 <- nytimes_items7 %>% html_text()

nytimes_news7 <- as.data.frame(nytimes_new7)

# Preparing the data set
colnames(nytimes_news7)[1] <- "Text"

Name <- rep(c("Understanding Rent Regulation in N.Y.C."), each =
dim(nytimes_news7)[1])
Name_news7 <- data.frame(Name)

NYTnews7 <- cbind(nytimes_news7, Name_news7)
```

```
nytimes8 <- read_html("https://www.nytimes.com/2022/07/02/realestate/proof-
of-income-renting.html")

nytimes_items8 <- nytimes8 %>% html_nodes(".css-at9mc1")

nytimes_new8 <- nytimes_items8 %>% html_text()

nytimes_news8 <- as.data.frame(nytimes_new8)

# Preparing the data set
colnames(nytimes_news8)[1] <- "Text"

Name <- rep(c("How Do I Prove That I Can Pay the Rent if I Don't Have a
Monthly Salary?"), each = dim(nytimes_news8)[1])
Name_news8 <- data.frame(Name)

NYTnews8 <- cbind(nytimes_news8, Name_news8)


nytimes9 <- read_html("https://www.nytimes.com/2022/12/27/business/what-
would-it-take-to-turn-more-offices-into-housing.html")

nytimes_items9 <- nytimes9 %>% html_nodes(".css-at9mc1")

nytimes_new9 <- nytimes_items9 %>% html_text()

nytimes_news9 <- as.data.frame(nytimes_new9)

# Preparing the data set
colnames(nytimes_news9)[1] <- "Text"

Name <- rep(c("What Would It Take to Turn More Offices Into Housing?"), each
= dim(nytimes_news9)[1])
Name_news9 <- data.frame(Name)

NYTnews9 <- cbind(nytimes_news9, Name_news9)


nytimes10 <- read_html("https://www.nytimes.com/2022/11/24/nyregion/home-
ownership-new-york-city.html")

nytimes_items10 <- nytimes10 %>% html_nodes(".css-at9mc1")

nytimes_new10 <- nytimes_items10 %>% html_text()

nytimes_news10 <- as.data.frame(nytimes_new10)

# Preparing the data set
```

```
colnames(nytimes_news10)[1] <- "Text"

Name <- rep(c("Is Homeownership Slipping Even Further Out of Reach for New
Yorkers?"), each = dim(nytimes_news10)[1])
Name_news10 <- data.frame(Name)

NYTnews10 <- cbind(nytimes_news10, Name_news10)


# Binding the data sets

all_news <-
rbind(NYTnews1,NYTnews2,NYTnews3,NYTnews4,NYTnews5,NYTnews6,NYTnews7,NYTnews8
,NYTnews9,NYTnews10)

Id <- seq(1:dim(all_news)[1])

all_news <- cbind(Id, all_news)

head(all_news, 2)
```

```
##   Id
## 1  1
## 2  2
                                                                   Text
## 1 The impact of the pandemic on New York City real estate has perhaps been
most stark in the Manhattan rental market, which had vacancies skyrocket in
2020 as tenants fled to greener places, to less expensive places from which
they could work remotely, or to their parents’ homes. But record numbers of
new lease signings in the last quarter of 2020 show that renters are once
again filling apartments, lured by incentives like slashed rents, free months
of rent, and the elimination of fees.
## 2
Manhattan can be an outlier when it comes to national real estate trends, but
it was far from the only place where the cost of a rental apartment shifted
after the pandemic hit. Apartment List’s most recent National Rent Report
reveals the cities across the country where rents changed the most from
January 2020 to January 2021 — and not all of them went down. It’s the basis
of this week’s chart.

##                             Name
## 1 How the Pandemic Blew Up Rents
## 2 How the Pandemic Blew Up Rents
```

The data set consists of 195 rows and 3 columns. These columns are Id, Name and Text. Some of the examples of the news headlines are “How the Pandemic Blew Up Rents”, “My Tenant Won’t Pay, but Won’t Go. What Can I Do?”, “The Brooklyn-Manhattan Rent Gap Is Shrinking Fast”, “The Brooklyn-Manhattan Rent Gap Is Shrinking Fast” and “New York City

May Freeze Rent Again for More Than 2 Million Tenants" and "A Pandemic Story: Brooklyn Tenants Who Stopped Paying Rent".

```
# Some more examples
all_news %>% select(Text) %>% sample_n(2,seed=1234) %>% pull()
```

```
## [1] "If your sister decides to keep the house, she will have to deal with
the tenant and the back rent — itself a monthslong process that will come
with steep attorney fees. And if she hasn't already started an eviction case,
she will be waiting in a long line."
## [2] "When Covid gripped the city in early 2020, Manhattan already had a
huge overhang of new condos — over 8,000 unsold units, which would have taken
more than eight years to sell at the time, Mr. Miller said. A healthy market
can sell out between two and two and a half years."
```

Then, data is examined to check if there are any duplicates.

```
# Duplicate check

all_news %>%
    group_by(Text) %>%
    summarize(n_reviews=n()) %>%
    mutate(pct=n_reviews/sum(n_reviews)) %>%
    arrange(-n_reviews) %>%
    top_n(10,n_reviews)
```

There are no frequent texts. No duplicate exists.

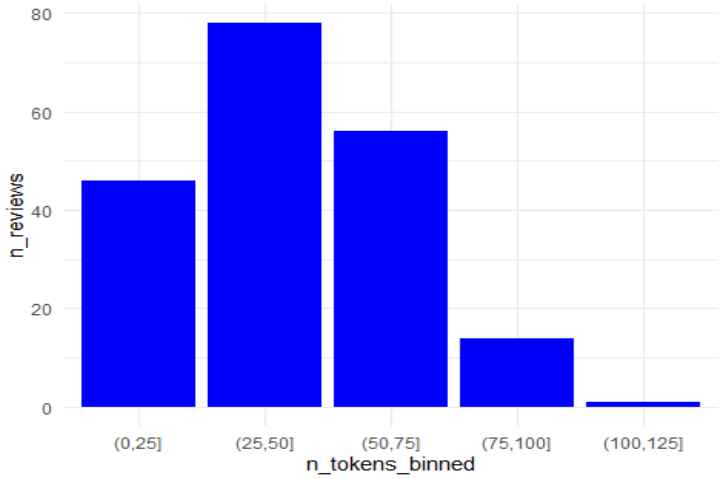
For the cleaning and preparation process of raw data, punctuations are removed, and it is converted to lower case. Later, and indicator validReview is created to eliminate some of the rows for example if the text is less than 5 characters, it won't be included to further analysis.

```
data <- all_news %>%
  #remove some punctuation with space
  #  and remove other punctuation and set to lower case.
  mutate(TextClean=gsub('[[:punct:]]+', '',
         gsub('\\\\n|\\.|\\,|\\;',' ',tolower(substr(Text,3,nchar(Text)-
1))))) %>%
  # create indicator validReview that is 0 to delete
  mutate(validReview=case_when(#texts less than 5 characters in length
                               nchar(TextClean) <5 ~ 0,
                               TRUE ~ 1))
```

Then, text is separated into tokens. The news with the number of tokens above a certain limit help much more in identifying different topics within news.

```
# Preparing the data on the token level

# divide text into separate words
```

```
tokens <- data %>%
    select(Id, TextClean) %>%
    unnest_tokens(word, TextClean)

tokens %>%
  group_by(Id) %>% summarise(n_tokens = n()) %>%
  mutate(n_tokens_binned = cut(n_tokens, breaks = c(0,seq(25,250,25),Inf)))
%>%
  group_by(n_tokens_binned) %>% summarise(n_reviews = n()) %>%
  ggplot(aes(x=n_tokens_binned,y=n_reviews)) +
    geom_bar(stat='identity',fill='blue') + theme_minimal()
```

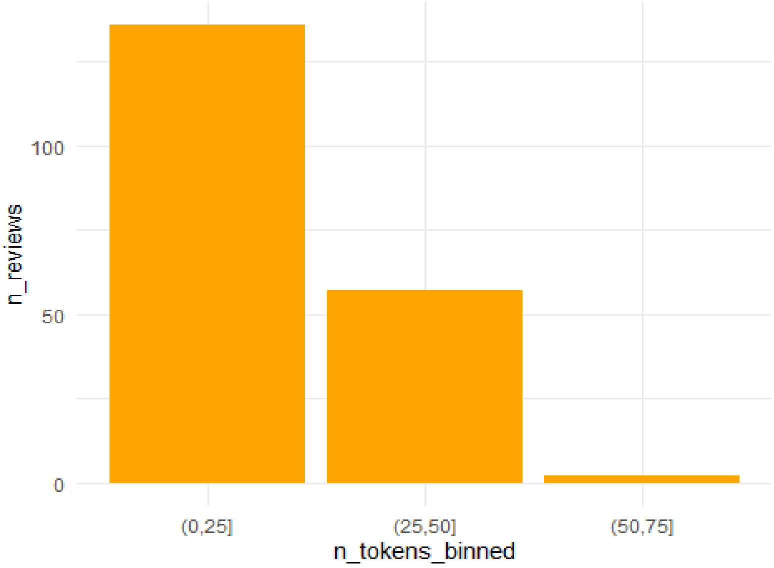


It seems that the news with more than 50 token might help more in identifying different topics within news.

Before acting on the number of tokens, stop words are removed since they have no added value in many cases. The number of stop words are 1298 from package stopwords. "'ll", "about", "already" or "'ve" can be given as an example for the most frequently used stop words.

```
# get stopwords from package stopwords
stopwords_sw_iso <-stopwords::stopwords(language = 'en',source='stopwords-
iso')

cat(paste0('Number of stop words from package stopwords (source=stopwords-
iso): ',length(stopwords_sw_iso),'\n\n'))

cat(paste0('First 50 stop words: ',paste(stopwords_sw_iso[1:50], collapse=',
'),', ...'))

stop_words <- stop_words %>% mutate(stopword=1)

#We're ready to drop the stop words.
```

```r
# First, let's check how a random text looked before removing stop words.

example = tokens %>% ungroup() %>% distinct(Id) %>%
  sample_n(size=1,seed=1234)
data %>% filter(Id==pull(example))  %>% select(Text) %>%
  pull() %>% paste0('\n\n') %>% cat()

# remove stopwords
tokens_ex_sw <- tokens %>%
    left_join(y=stop_words, by= "word", match = "all") %>%
filter(is.na(stopword))

# ... and recheck after removing stopwords
tokens_ex_sw %>% filter(Id==example) %>%
    summarize(Text_cleaned=paste(word,collapse=' ')) %>% pull() %>% cat()

# check new lengths after removing stop words
tokens_ex_sw %>%
  group_by(Id) %>% summarise(n_tokens = n()) %>%
  mutate(n_tokens_binned = cut(n_tokens, breaks = c(0,seq(25,250,25),Inf)))
%>%
  group_by(n_tokens_binned) %>% summarise(n_reviews = n()) %>%
  ggplot(aes(x=n_tokens_binned,y=n_reviews)) +
    geom_bar(stat='identity',fill='orange') + theme_minimal()
```



After removing stop words, the number of word tokens in news changed.

Later on, instead of using tokens as single words, combinations of subsequent words - named bigrams for two adjacent words and trigrams for three — are created since they are also very useful tokens in NLP tasks.

In the previous step, stop words are identified and removed. Since it is defined that these terms to be irrelevant, removing bigrams for which at least one of the tokens is aimed to be removed. Identification needs to be performed for relevant bigrams on the texts prior to removing stop words, otherwise it can cause many bigrams that actually were not present in the text but are a result of removing one or more intermediary terms.

```
# create bigrams with the unnest_tokens function, specifying the ngram lenght
(2)
bigrams <- tokens %>%
    group_by(Id)  %>%
    summarize(TextClean=paste(word,collapse=' ')) %>%
    unnest_tokens(bigram, token = "ngrams",n = 2, TextClean)

print(paste0('Total number of bigrams: ',dim(bigrams)[1]))

## [1] "Total number of bigrams: 8286"

#remove bigrams containing stopwords
bigrams_separated <- bigrams %>%
    separate(bigram, c('word1', 'word2'), sep=" ")

bigrams_filtered <- bigrams_separated %>%
    filter(!word1 %in% stop_words$word & !word2 %in% stop_words$word)

bigrams_united <- bigrams_filtered %>%
    unite(bigram, word1, word2, sep = '_')

print(paste0('Total number of bigrams without stopwords:
',dim(bigrams_united)[1]))

# show most frequent bigrams
top10_bigrams = bigrams_united %>% group_by(bigram) %>% summarize(n=n()) %>%
  top_n(10,wt=n) %>% select(bigram) %>% pull()
print(paste0('Most frequent bigrams: ',paste(top10_bigrams,collapse=", ")))
```

As a result, the total number of biagrams is 8286. On the other hand, the number of biagrams without stopwords equals to 1640. Most of the biagrams consist of stop words. The most frequent biagrams are 3_percent, 5_percent, housing_stock, median_rent, monthly_rent, office_conversions, oneyear_leases etc.

Then, positive and negative words are introduced. The positive and negative words files mainly belong to the article "Mining and Summarizing Customer Reviews" article by Minqing Hu and Bing Liu from University of Illinois Chicago. In the lists, there many misspelled words which are not mistakes. They are included as these misspelled words appear frequently in social media content. There are 2006 words in the positive words list and "accomplishment", "empathy", "helping" and "tough" are example of the positive words. Also, there exist 4783 negative words in the negative words list and "accusation", “costly”, “hating” and “tension” are example of the negative word in the list. By summing all positive words (+1) and all negative words (-1) and standardizing by the total number of positive/negative words in the text, sentiment score is calculated.

```
positive_words_en <- read_csv("positive-words.txt",
col_names=c('word'),col_types='c') %>% mutate(pos=1,neg=0)
negative_words_en <- read_csv("negative-words.txt",
col_names=c('word'),col_types='c') %>% mutate(pos=0,neg=1)

#combine positive and negative tokens and print statistics
sentiment_en <- rbind(positive_words_en, negative_words_en)
sentiment_en %>%
summarize(sentiment_words=n_distinct(word),positive_words=sum(pos),
                           negative_words=sum(neg)) %>% print()
```

```
## # A tibble: 1 × 3
##   sentiment_words positive_words negative_words
##             <int>          <dbl>          <dbl>
## 1            6786           2006           4783
```

```
# original text
Text <- data %>% select(Id,Text)
# add cleaned text
TextClean <- tokens_ex_sw %>% group_by(Id) %>%
  summarize(TextClean=paste(word,collapse=' '))
# add bigrams without stopwords
Bigrams <- bigrams_united %>% group_by(Id) %>%
  summarize(bigrams=paste(bigram,collapse=' '))

# combine original text with cleaned text
NYTimes <- Text %>% inner_join(TextClean,by='Id') %>%
  left_join(Bigrams,by='Id')
```

Now that we’ve analyzed, filtered and cleaned the news texts, we can use them as the starting point for topic modelling.

There are important steps for topic modeling. As an example, the new steps are tokenization of the prepared data including bigrams, sample news for training the topic model, filtering relevant tokens and creating document term matrix. Biagrams and unigrams are combined into TextClean and text is devided into separate words. “impact”, “pandemic”, “york”, “city”, “real” and “estate” are the examples of tokens.

```
# combine unigrams and bigrams into TextClean and divide text into separate
words
new_tokens <- NYTimes %>%
    mutate(TextClean = paste0(TextClean,bigrams)) %>%
    select(Id, TextClean) %>%
    unnest_tokens(token, TextClean) %>%
    # filter out news texts with less than 25 tokens
    group_by(Id) %>% mutate(n_tokens = n()) %>% filter(n_tokens>=25) %>%
    ungroup() %>% select(-n_tokens)

head(new_tokens)
```

```
## # A tibble: 6 × 2
##      Id token
##   <int> <chr>
## 1     1 impact
## 2     1 pandemic
## 3     1 york
## 4     1 city
## 5     1 real
## 6     1 estate
```

70% of the data selected randomly into training set and the remaining 30% sample into test data set. There are still unique tokens available for the topic model. There are 79 unique news and 2074 unique tokens selected to train topic model.

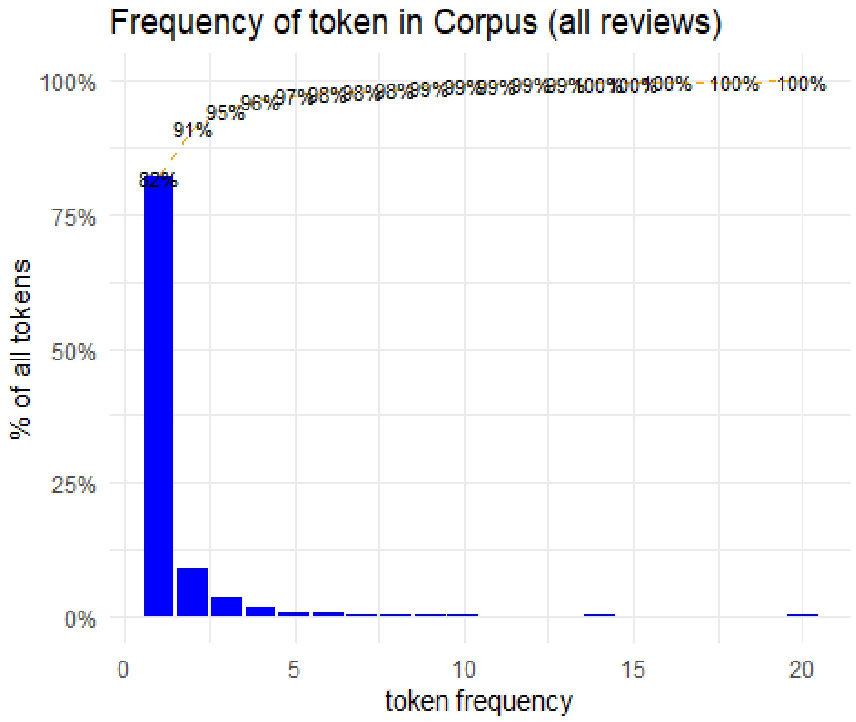
```
# Tokenize our prepared text (including bigrams)

# Use splitted data: 70% of the data selected randomly into training set and
the remaining 30% sample into test data set
library(readxl)

NYTimes_Training <- read_excel("NYTimes_Training.xlsx") #randomly assigned
Training & Test groups (0.7-0.3)
NYTimes_Training$Training <- NYTimes_Training$Training %>% replace_na(0)

tokens_train <- new_tokens %>%
    inner_join(NYTimes_Training,by = "Id") %>%
    mutate(train_smpl = case_when(
             Training == 1 ~ 1,
             TRUE~0))

#create train data using train_smpl as filter
tokens_train <- tokens_train %>% filter(train_smpl == 1)

sprintf('%s unique news and %s unique tokens selected to train topic model',
    n_distinct(tokens_train$Id),n_distinct(tokens_train$token))
```

```
## [1] "79 unique news and 2074 unique tokens selected to train topic model"
```

Now that bigrams are added to the tokens and texts are re-tokenized, there are still many unique tokens available for the topic model. It's best practice to get rid of the longtail of infrequent terms.

```
# Filter tokens
```

```
tokens_train %>%
  group_by(token) %>% summarize(token_freq=n()) %>%
  mutate(token_freq_binned =
case_when(token_freq>20~20,TRUE~as.numeric(token_freq))) %>%
  group_by(token_freq_binned) %>% summarise(n_tokens = n()) %>%
  mutate(pct_tokens = n_tokens/sum(n_tokens),
         cumpct_tokens = cumsum(n_tokens)/sum(n_tokens)) %>%
  ggplot(aes(x=token_freq_binned)) +
          scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
          geom_bar(aes(y=pct_tokens),stat='identity',fill='blue') +

geom_line(aes(y=cumpct_tokens),stat='identity',color='orange',linetype='dashe
d') +

geom_text(aes(y=cumpct_tokens,label=scales::percent(cumpct_tokens,accuracy=1)
),
                    size=3) + theme_minimal() +
          ggtitle("Frequency of token in Corpus (all text)") + xlab("token
frequency") +
          ylab("% of all tokens")
```



By the help of the plot showing the huge amount of the unique tokens occurs rarely, frequently used tokens are chosen. The plot above clearly shows that a huge amount of the unique tokens occurs rarely in all the news. Around 82% of all unique tokens occur 1 time and 91% of all unique tokens occur at most 2 times in the corpus. These low frequency tokens impact the topic model analysis. To focus on frequently used tokens, we select tokens that occur 2 times or more in the prepared train data.

```
tokens_train %>%
  group_by(token) %>% summarize(token_freq=n()) %>%
  mutate(min_2_freq = case_when(token_freq<2~'token frequency: <2',
                                TRUE~'token frequency: >=2')) %>%
  group_by(min_2_freq) %>% summarise(n_unique_tokens =
n(),n_tokens=sum(token_freq)) %>%
  mutate(pct_unique_tokens = scales::percent(n_unique_tokens /
sum(n_unique_tokens)),
         pct_all_tokens=scales::percent(n_tokens / sum(n_tokens)))
```

```
## # A tibble: 2 × 5
##   min_2_freq           n_unique_tokens n_tokens pct_unique_tokens
pct_all_tokens
##   <chr>                          <int>    <int> <chr>             <chr>
## 1 token frequency: <2             1702     1702 82%               54.1%
## 2 token frequency: >=2             372     1443 18%               45.9%
```

There are 1702 unique tokens which occur at most 1 time in the corpus. Also, 372 out of 1443 tokens are seen at least 2 times and unique.

After filtering the tokens to use for building the topic model, the input for LDA is created. This requires a document-term-matrix or DTM which is a matrix within the rows all our documents and in the columns all the terms. The dimension of document-term-matrix is Documents (78) x Tokens (112) and the average token frequency is 10.74.

```
# Creating DTM

# remove infrequent tokens
tokens_train_smpl <- tokens_train %>%
  group_by(token) %>% mutate(token_freq=n()) %>%  filter(token_freq>=4)

# create document term matrix
dtm <- tokens_train_smpl %>%
  cast_dtm(document = Id,term = token,value = token_freq)

#check dimenstions of dtm
cat(paste0('DTM dimensions: Documents (',dim(dtm)[1],') x Tokens
(',dim(dtm)[2],')',
           ' (average token frequency: ',round(sum(dtm)/sum(dtm!=0),2),')'))
```

```
## DTM dimensions: Documents (78) x Tokens (112) (average token frequency:
10.74)
```

Then, LDA has a number of parameters that impact the outcome for the topic modeling. The most important ones are *k* indicating the number of topics, *method* referring to the topic models package enables different optimization methods, *control* standing for list of control variables to guide estimation, *nstart* indicating the number of runs to perform with the same settings but different seeds, and lastly, *best* referring to only the run with the best fitting result is kept. Picking the best k is difficult and it is best to try different number of k's.

```
lda_fit <- LDA(dtm, k = 3)
```

Finally, groups under several topics are created.

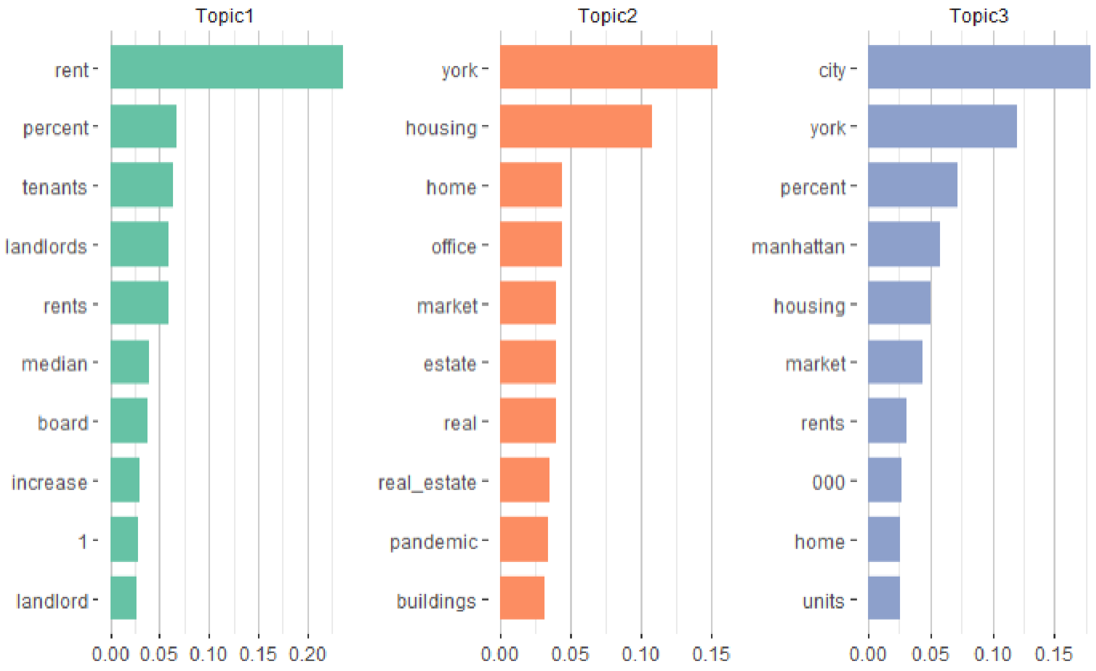
```
#Evaluate Topic Model

# phi (topic - token distribution matrix) -  topics in rows, tokens in
columns:
phi <- posterior(lda_fit)$terms %>% as.matrix
cat(paste0('Dimensions of phi (topic-token-matrix):
',paste(dim(phi),collapse=' x '),'\n'))

cat(paste0('phi examples (8 tokens): ','\n'))

phi[,1:8] %>% as_tibble() %>% mutate_if(is.numeric, round, 5) %>% print()

# theta (document - topic distribution matrix) -  documents in rows, topic
probs in columns:
theta <- posterior(lda_fit)$topics %>% as.matrix
cat(paste0('\n\n','Dimensions of theta (document-topic-matrix): ',
           paste(dim(theta),collapse=' x '),'\n'))

cat(paste0('theta examples (8 documents): ','\n'))

theta[1:8,] %>% as_tibble() %>% mutate_if(is.numeric, round, 5) %>%
  setNames(paste0('Topic', names(.))) %>% print()
```

As *k* equals to 3, there are 3 different topic groups.

To explore the topic model, let's start by looking at the most important tokens per topic. To do so, we need to specify when a token is important for a topic. We could argue that the token is important for the topic when it has a high probability to occur within a topic p(token|topic). Let's see how this looks for the 3-topics topic model.

```
library(reshape2) # reboot of the reshape package

# get token probability per token per topic
topics <- tidy(lda_fit)

# only select top-10 terms per topic based on token probability within a
```

```
topic
plotinput <- topics %>%
  mutate(topic = as.factor(paste0('Topic',topic))) %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

# plot highest probability terms per topic
names <- levels(unique(plotinput$topic))
colors <- RColorBrewer::brewer.pal(n=length(names),name="Set2")

plist <- list()

for (i in 1:length(names)) {
  d <- subset(plotinput,topic == names[i])[1:10,]
  d$term <- factor(d$term, levels=d[order(d$beta),]$term)

  p1 <- ggplot(d, aes(x = term, y = beta, width=0.75)) +
  labs(y = NULL, x = NULL, fill = NULL) +
  geom_bar(stat = "identity",fill=colors[i]) +
  facet_wrap(~topic) +
  coord_flip() +
  guides(fill=FALSE) +
  theme_bw() + theme(strip.background  = element_blank(),
                     panel.grid.major = element_line(colour = "grey80"),
                     panel.border = element_blank(),
                     axis.ticks = element_line(size = 0),
                     panel.grid.minor.y = element_blank(),
                     panel.grid.major.y = element_blank() ) +
  theme(legend.position="bottom")

  plist[[names[i]]] = p1
}

library(gridExtra)

do.call("grid.arrange", c(plist, ncol=3))
```



k is determined as 3 because more than 3 topic group do not provide extra information. It is aimed to give logical names to the topics. For this reason, it is the focus what the words under the topics represent or how these words can be generalized. They all contain pretty similar words and there is no big difference between the groups.

Looking at the first topic group, it can be summarized as, the tenants have difficulty in agreeing with the landlords about the rent prices of the houses. Landlords tend to increase rents by high percentages, and these prices have risen well above median house prices over time. So, this topic may summarize the context of the relationship between tenants and landlords. On the other hand, Topic2 topic group consists of the word "pandemic" with "housing", "office", "buildings" and "real estate". It is mentioned in the news that with the effect of the pandemic, the price of not only houses but also all kinds of properties are highly affected. So, this topic group can be link to pandemic related issues. Lastly, since Topic3 consists of the words such as "percent", "market", "000" and "unit". This topic group may be considered as a word group containing mathematical/statistical words. It can mainly emphasize the effect of pandemic on housing market with the help of numbers.

```
# phi (topic - token distribution matrix) - tokens in rows, topic scores in
columns:
phi <- posterior(lda_fit)$terms %>% as.matrix

# theta (document - topic distribution matrix) - documents in rows, topic
probs in columns:
```

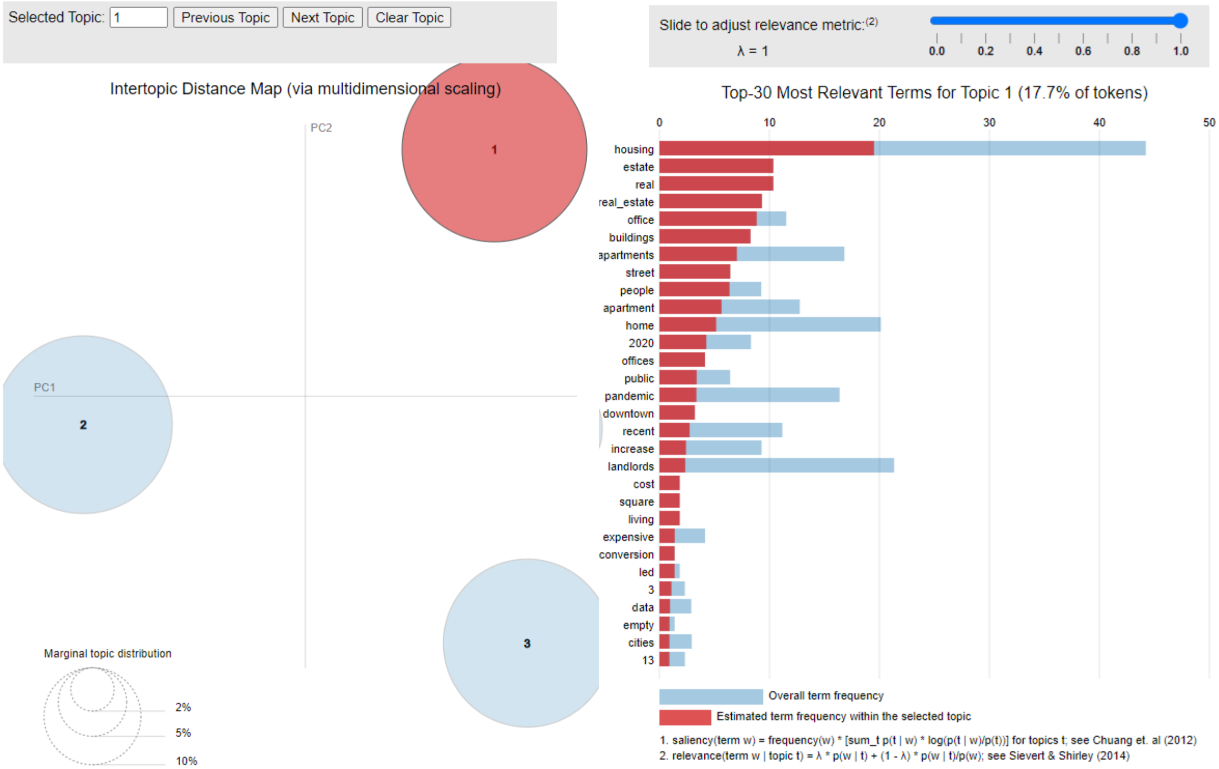
```r
theta <- posterior(lda_fit)$topics %>% as.matrix

# number of tokens per document
doc_length <- tokens_train_smpl %>% group_by(Id) %>%
  summarize(doc_length=n()) %>% select(doc_length) %>% pull()

# vocabulary: unique tokens
vocab <- colnames(phi)

# overall token frequency
term_frequency <- tokens_train_smpl %>% group_by(token) %>%
  summarise(n=n()) %>% arrange(match(token, vocab)) %>% select(n) %>% pull()


# create JSON containing all needed elements
json <- createJSON(phi, theta, doc_length, vocab, term_frequency)

# Iterations towards the winning Topic Model

# modify the tokens to consider in topic model
tokens_train_smpl_new <- tokens_train %>%
  # remove infrequent tokens (<4)
  group_by(token) %>% mutate(token_freq=n()) %>%  filter(token_freq>=4) %>%
ungroup() %>%
  # combine some tokens that are dominant in solutions and represent same
meaning
  mutate(token = case_when(token == 'month' ~ 'monthly', token == 'building'
~ 'apartments',
                             token == 'rents' ~ 'rent', token == 'landlord' ~
'landlords',
                             TRUE~token)) %>%
  # remove some 'too frequent' tokens
  filter(!token  %in% c('1','000','york_city','piscione','ms_piscione'))


# recreate the document term matrix after modifying the tokens to consider
dtm_new <- tokens_train_smpl_new %>%
    cast_dtm(document = Id,term = token,value = token_freq)

#check dimensions of dtm
cat(paste0('DTM dimensions: Documents (',dim(dtm_new)[1],') x Tokens
(',dim(dtm_new)[2],')',
           ' (average token frequency:
',round(sum(dtm_new)/sum(dtm_new!=0),2),')'))

## DTM dimensions: Documents (78) x Tokens (106) (average token frequency:
10.64)

# estimate lda with k topics, set control variables nstart=n to have n runs,
#   best=FALSE to keep all run results and set the seed for reproduction
```

```
lda_fit_def <- LDA(dtm_new, k = 7,control =
list(nstart=1,best=TRUE,seed=5678))
saveRDS(lda_fit_def,'lda_fit_def.RDS')
```

Finally, JSON is created to better observe the words contained in different topic groups in a more interactive way. JavaScript Object Notation (JSON) stores text-based data in a human-readable format.

```
#since LDAvis package cannot be used in notebook, we use pyLDAvis, this can
be used in notebook
# We need to export the R topic model output to use in python's pyLDAvis

# phi (topic - token distribution matrix) -  tokens in rows, topic scores in
columns:
phi <- posterior(lda_fit_def)$terms %>% as.matrix

# theta (document - topic distribution matrix) -  documents in rows, topic
probs in columns:
theta <- posterior(lda_fit_def)$topics %>% as.matrix

# number of tokens per document
doc_length <- tokens_train_smpl_new %>% group_by(Id) %>%
  summarize(doc_length=n()) %>% select(doc_length) %>% pull()

# vocabulary: unique tokens
vocab <- colnames(phi)

# overall token frequency
term_frequency <- tokens_train_smpl_new %>% group_by(token) %>%
  summarise(n=n()) %>% arrange(match(token, vocab)) %>% select(n) %>% pull()

# use tsne method to calculate distance between topics (default sometimes
fails
# for details:
https://www.rdocumentation.org/packages/LDAvis/versions/0.3.2/topics/createJS
ON)
library(tsne) # t-distributed stochastic neighbor embedding
svd_tsne <- function(x) tsne(svd(x)$u)

# create JSON containing all needed elements
json <- createJSON(phi, theta, doc_length, vocab,
term_frequency,mds.method=svd_tsne)

# render LDAvis - in RStudio, it opens a new window with the interactive
LDAvis tool
serVis(json) # press ESC or Ctrl-C to kill
```

As it can be observed below, topics can be compared visually. The overall term frequency and estimated term frequency within the selected topic can be examined. It shows top 30 most relevant terms for each topic, as well as other statistics.



Selected Topic: 1
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 1 (17.7% of tokens)
0 10 20 30 40 50
housing
estate
real
real_estate
office
buildings
apartments
street
people
apartment
home
2020
offices
public
pandemic
downtown
recent
increase
landlords
cost
square
living
expensive
conversion
led
3
data
empty
cities
13
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Third Research Question: Are the unexpected changes in the US housing market having currently a positive or negative impact on society? What are the most frequently used words in individual thoughts on Twitter and what does it indicate?

In the third research question, the positive or negative effects of the unexpected change in the US housing market on society is aimed to observed by using Twitter data. As mentioned above, the focus is New York and California since they both have big economies. It gives the chance of observing the general opinion of the people.

```
# Twitter Web data Scraping

library(twitteR) # provides access to the Twitter API
library(openssl) # toolkit for Encryption, Signatures and Certificates Based
on OpenSSL
library(httpuv) # provides low-level socket and protocol support for handling
HTTP and WebSocket requests directly from within R
library(base64enc) # provides tools for handling base64 encoding
library(devtools) # collection of package development tools
```

```
# Twitter permission

# consumer_key = "XXX"
# consumer_secret = "YYY"
# access_token = "ZZZ"
# access_secret = "VVV"

# setup_twitter_oauth(consumer_key, consumer_secret, access_token,
access_secret)
```

For the New York data set, tweets were selected containing the words “New York”, “NYC”, “rent”, “house price”, “real estate” and “properties”. *search_tweets* only return the data of the last six to nine days. Also, these tweets do not include retweets.

```
library(rtweet) #an implementation of calls designed to collect and organize
Twitter data

NewYork_data1 <- searchTwitter("New York+rent", n=500)
NewYork_data2 <- searchTwitter("New York+house price", n=500)
NewYork_data3 <- searchTwitter("NYC+rent", n=500)
NewYork_data4 <- searchTwitter("New York+real estate", n=500)
NewYork_data5 <- searchTwitter("New York+properties", n=500)

NYC1 <- twListToDF(NewYork_data1)
NYC2 <- twListToDF(NewYork_data2)
NYC3 <- twListToDF(NewYork_data3)
NYC4 <- twListToDF(NewYork_data4)
NYC5 <- twListToDF(NewYork_data5)

# Binding the data sets

NewYorkCityRent<- rbind(NYC1,NYC2,NYC3,NYC4,NYC5)
head(NewYorkCityRent, 3)
```

```
## text
## 1 Rent is up across the board, but increases are hitting low-income
renters the hardest. \n\nLet this be the year we wi… https://t.co/tXM0ocyCqJ
## 2              as much as i said i wanted to move to new york i could
NEVER 😭 it’s just tooooo congested an rent is high asf for no reason lmaoo


   favorited favoriteCount replyToSN  created            truncated replyToSID
## 1         FALSE         2          2023-01-12 20:18:53      TRUE       <NA>
## 2         FALSE         0          2023-01-11 19:06:48     FALSE       <NA>


##                    id   replyToUID
## 1 1617255501913366528         <NA>
## 2 1617237362597142528         <NA>
```

```
statusSource
## 1   <a href="http://twitter.com/download/android" rel="nofollow">Twitter
for Android</a>
## 2   <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter
for iPhone</a>
```

```
##      screenName retweetCount isRetweet retweeted longitude latitude
## 1     QueensWFP            1     FALSE     FALSE      <NA>     <NA>
## 2   thatboymuva            0     FALSE     FALSE      <NA>     <NA>
```

```
# Summary statistics of NewYorkCityRent data set

dim(NewYorkCityRent)
```

```
## [1] 3211   16
```

```
distinct_NewYorkCityRent <- distinct(NewYorkCityRent)
dim(distinct_NewYorkCityRent)
```

```
## [1] 3095   16
```

The NewYorkRent data set contains 3211 tweets. After removing the duplicates, there are 3095 tweets left. Also, there exist 16 columns such as text, favoriteCount, id, screenName, retweetCount etc.

Likewise, another data set containing the words “California”, “CA”, “rent”, “house price”, “real estate” and “properties” is created without retweets.

```
# Twitter Web data Scraping

CA_data1 <- searchTwitter("California+rent", n=500)
CA_data2 <- searchTwitter("California+house price", n=500)
CA_data3 <- searchTwitter("CA+rent", n=500)
CA_data4 <- searchTwitter("California+real estate", n=500)
CA_data5 <- searchTwitter("California+properties", n=500)

CA1 <- twListToDF(CA_data1)
CA2 <- twListToDF(CA_data2)
CA3 <- twListToDF(CA_data3)
CA4 <- twListToDF(CA_data4)
CA5 <- twListToDF(CA_data5)

# Binding the data sets
CaliforniaRent<- rbind(CA1,CA2,CA3,CA4,CA5)
head(CaliforniaRent$text, 2)
```

```
##        text
## [1]   "RT @latimes: The proposal would also block evictions until February
2024 for tenants who have unauthorized pets or who added residents who…"
```

```
## [2]   "Consider These If You Want To Rent An
Apartment\nhttps://t.co/uKK7X71XjT\n#SanDiego #RealEstate #DreamHome
#ForSale… https://t.co/UrOZrnU2j2"


   favorited favoriteCount replyToSN created             truncated replyToSID
## 1         FALSE          0        2023-01-12 19:17:21  TRUE      <NA>
## 0         FALSE          0        2023-01-12 16:52:53  TRUE      <NA>
##                     id   replyToUID
## 1 1617254245217652718         <NA>
## 2 1617215379675276772         <NA>

statusSource
## 1   <a href="http://twitter.com/download/android" rel="nofollow">Twitter
for Android</a>
## 2   <a href="http://twitter.com/download/android " rel="nofollow">Twitter
for iPhone</a>

##      screenName retweetCount isRetweet retweeted longitude latitude
## 1  historydefin            1     FALSE     FALSE      <NA>     <NA>
## 2        latime            1      TRUE     FALSE      <NA>     <NA>
```

```
# Summary statistics of CaliforniaRent data set

dim(CaliforniaRent)
```

```
## [1] 3202   16
```

```
distinct_CaliforniaRent <- distinct(CaliforniaRent)
dim(distinct_CaliforniaRent)
```

```
## [1] 3109   16
```

CaliforniaRent data set contains 3202 tweets. After removing the duplicates, 93 repeated tweets are eliminated and there exist 3109 tweets left. The number of columns is the same with the New York data set.

NewYorkCityRent and CaliforniaRent data sets were merged by row binding for further analysis. Also, the tweets which include the words “angry”, “anger”, “happy”, “happiness” are examined. 63 of the tweets directly includes those words representing direct feelings and several example tweets can be seen below.

```
library(RVerbalExpressions) # create regular expressions easily


# Merging New York and California data sets
```

```
all <- rbind(NewYorkCityRent, CaliforniaRent)

#checking the tweets consist angry, anger, happy, happiness
regex <-
  rx_with_any_case() %>%
  rx_either_of("angry ", "anger ", "happy ", "happiness")

all %>%
  filter(str_detect(text, regex)) %>%
  select(text) %>%
  head(3)
```

```
##
text
## 1       My new co-worker George Santos is a distraction and a danger to
democracy https://t.co/TvVv8U9ER9 via @NBCNewsTHINK… https://t.co/MHhLjongBW
## 2               @craftyworkingmo @falc_peter @DemforDeSantis Typical
response I’m happy to see he still lives rent free in y’all’s heads. F New
York
## 3 Its TODAY!!\nJoin the NY/NJ RENA &amp; the New York Real Estate Auctions
for a Happy Hour Meetup.\nYour network will be e… https://t.co/FPZdzPDwGh
```

Next, a corpus object is created from the data set for these two states.

```
# The corpus object

# rename user id and add row_id
 all <- all %>%
   mutate(user_id = id) %>%
   select(-id) %>%
   rowid_to_column("id")

 # Error in rowid_to_column(., "id") :
 # could not find function "rowid_to_column"

tweets_corpus <- corpus(all,
                        docid_field = "id",
                        text_field = "text")
```

Also, summary of the tweet corpus can be seen below. Corpus contains of 6413 documents. As it can be seen, the first text (tweet) has 25 tokens.

```
summary(tweets_corpus, n = 1)
```

```
## Corpus consisting of 6413 documents, showing 1 document:
##
##  Text Types Tokens Sentences favorited favoriteCount replyToSN
##     1    23     25         2     FALSE             2      <NA>
##                created truncated replyToSID replyToUID
##  2023-01-22 20:18:53      TRUE       <NA>       <NA>
##  statusSource
##  <a href="http://twitter.com/download/android" rel="nofollow">Twitter for
Android</a>
##  screenName retweetCount isRetweet retweeted longitude latitude
##   QueensWFP            1     FALSE     FALSE      <NA>     <NA>
##             user_id
##  1617255501913366528
```

Then, tweets_corpus is tokenized. Punctuations, numbers, symbols and urls are removed.

```
# Tokenization
tweet_tokens <- quanteda::tokens(tweets_corpus,
                                 remove_punct = TRUE,   # removes
punctuations
                                 remove_numbers = TRUE, # removes numbers
                                 remove_symbols = TRUE, # removes symbols
(also: emojis)
                                 remove_url = TRUE)     # removes urls

head(tweet_tokens, 1)
```

```
## Tokens consisting of 1 document and 15 docvars.
## 1 :
##  [1] "Rent"       "is"         "up"         "across"     "the"
##  [6] "board"      "but"        "increases"  "are"        "hitting"
## [11] "low-income" "renters"
## [ ... and 9 more ]
```

Later, stop words are removed and capital letters in tokens are converted to lowercase.

```
# lower key tokens
tweet_tokens_lk <- tweet_tokens %>% tokens_tolower()

# remove stop words
tweet_tokens_nosw <- tweet_tokens_lk %>%
  tokens_remove(stopwords("english"))

head(tweet_tokens_nosw,1)
```

```
## Tokens consisting of 1 document and 15 docvars.
## 1 :
##  [1] "rent"       "across"     "board"      "increases"  "hitting"
##  [6] "low-income" "renters"    "hardest"    "let"        "year"
## [11] "wi"
```

Then, stemming is applied to tweet tokens. Stemming is the process of reducing inflected words to their word stem, base or root form. It makes the training data denser and reduces the size of the dictionary.

```r
# Stemming

tweet_tokens_nosw %>% tokens_wordstem() %>%
    head(1)
```

```
## Tokens consisting of 1 document and 15 docvars.
## 1 :
##  [1] "rent"      "across"    "board"     "increas"   "hit"       "low-
incom"
##  [7] "renter"    "hardest"   "let"       "year"      "wi"
```

```r
ud_model_en <- udpipe_download_model(language = "english")

udpipe_lemmas <- udpipe(tweets_corpus, object = ud_model_en) # Tokenising,
Lemmatising, Tagging and Dependency Parsing of raw text in TIF format

docs <- udpipe_lemmas %>%
  group_by(doc_id) %>%
  summarize(text = paste(lemma, collapse = " ")) %>%
  distinct()

lemma_tokens_nosw <- docs %>%
  corpus() %>%
  quanteda::tokens(remove_punct = TRUE,
                   remove_numbers = TRUE,
                   remove_symbols = TRUE,
                   remove_separators = TRUE,
                   remove_url = TRUE) %>%
  tokens_tolower() %>%
  tokens_remove(stopwords("english"))

library(quanteda.textstats)

lemma_tokens_nosw %>%
  textstat_collocations() %>%
  arrange(desc(count)) %>%
  head(10)
```

```
##         collocation count count_nested length    lambda        z
## 14         new york  1806            0      2  9.796067 34.14010
## 156     real estate   584            0      2 13.191390 20.03598
## 1         york city   307            0      2  4.048135 46.00446
## 33  landlord longer   170            0      2 10.157743 28.07077
## 8   rental property   166            0      2  5.856771 35.91010
## 24      allow evict   165            0      2 10.541939 30.09731
## 39     longer allow   165            0      2 10.893392 27.77044
## 23     evict tenant   164            0      2  9.023604 30.15305
## 7     single family   154            0      2  8.261766 38.13903
## 3       family home   151            0      2  6.117846 41.29211
```

Then, kwic function is used and it returns a list of a keyword supplied by the user in its immediate context, identifying the source text and the word index number within the source text.

```
kwic(lemma_tokens_nosw, pattern = phrase("rent"))
```

```
## Keyword-in-context with 1,222 matches.
##  [1000, 10]  nyc want get trap work |
##  [1001, 6]  @aaronacarr nyc nimby infestation destroy |
##
##  rent | across board increase hit low
##  rent | film version pulitzer tony award
```

Keyword-in-context has 1222 matches when the phrase is chosen as “rent”.

Later on, N-grams models which are useful in many text analytics applications where sequences of words are relevant are used to see the top features in the groups.

```
#N-GRAMS
tokens_ngrams <- lemma_tokens_nosw %>%
  tokens_ngrams(n = 2:5)

dfm <- dfm(lemma_tokens_nosw)

topfeatures(dfm, 5)
```

```
##          rt         new        york california        rent
##        2718        2409        1818        1569        1222
```

Top features are “rt”, “new”, “york”, “california” and “rent” as expected since they are used as the key words for searching the tweets.

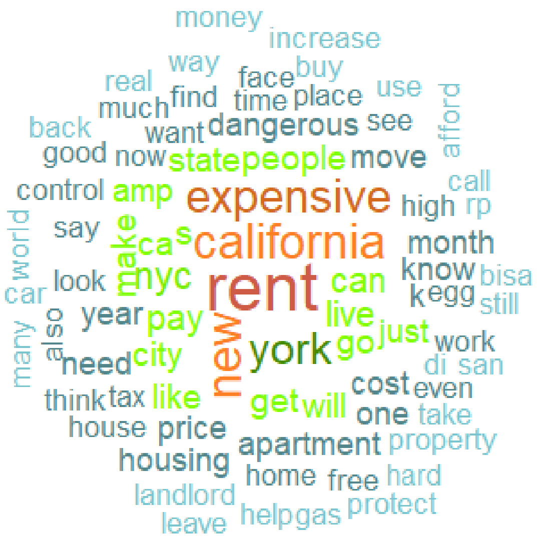
```
dfm_mentions <- dfm_select(dfm, "@*")

topfeatures(dfm_mentions, 5)
```

```
##           @occrp @antiracistsouth    @katieporteroc          @latime
##              144              117                97               69
##    @historydefin
##               64
```

Also, the most mentioned account names are occrp, anritacistsouth, katieporteroc, lateime and historydefin. It is interesting that the second and three of these nicknames are pages that defend against racism and violence. The account named as “antiracistsouth” is called “Anti-Racist South | #StopCopCity”, while the account “occrp” is called “Organized Crime and Corruption Reporting Project”.

The word cloud is created for observing the most frequently used words.

```
# Word Cloud for the states

# grouped DFM
tweet_dfm_grouped <- dfm_group(dfm)

# wordcloud
library(quanteda.textplots)
textplot_wordcloud(tweet_dfm_grouped,
                   min_size = 1.7,                    ,
                   max_size = 4,
                   max_words = 80,
                   color = colors()[45:60])
```



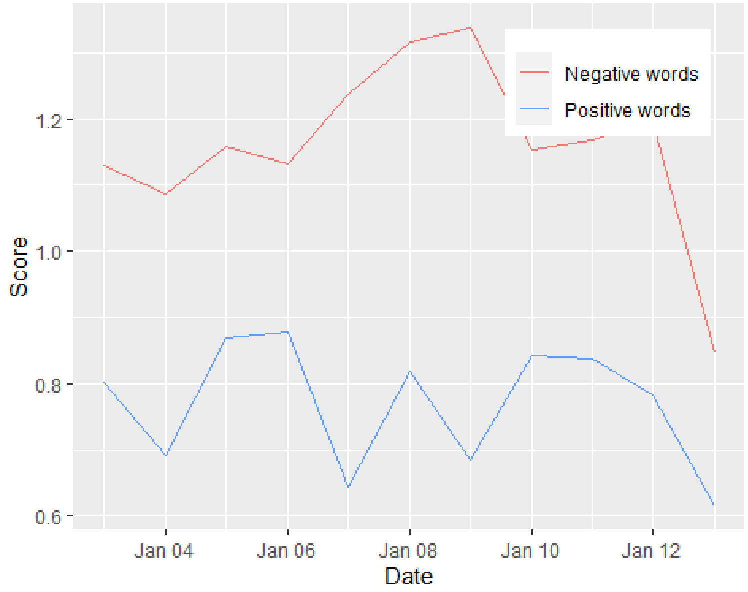
As seen in the word cloud, the words “rent”, “california”, “new” & “york”, “expensive” are the most used words, respectively. From another point of view, words such as “tax”, “dangerous”, “cost” and “increase” are also included in the word cloud summarizing the tweets.

As a last step, two dictionaries were introduced again to understand whether tweeter users have negative or positive feelings on related subject.

```
pos = scan("positive-words.txt", what = "character", comment.char = ";")
neg = scan("negative-words.txt", what = "character", comment.char = ";")

getSentimentScore = function(tweet_text, pos, neg) {

  sentence = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", tweet_text)

  sentence = gsub("@\\w+", "", sentence)

  sentence = gsub("[[:punct:]]", "", sentence)
  sentence = gsub("[[:cntrl:]]", "", sentence)

  sentence = gsub("[[:digit:]]", "", sentence)

  sentence = gsub("http\\w+", "", sentence)

  sentence = gsub("^\\s+|\\s+$", "", sentence)

  sentence = iconv(sentence, "UTF-8", "ASCII", sub = "")
  sentence = tolower(sentence)

  word.list = strsplit(sentence, " ")

  score = numeric(length(word.list)) # Loop through each tweet
  positive = numeric(length(word.list))
  negative = numeric(length(word.list))
  for (i in 1:length(word.list)) {

    pos.matches = match(word.list[[i]], pos)
    neg.matches = match(word.list[[i]], neg)

    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)

    score[i] = sum(pos.matches) - sum(neg.matches)
    positive[i] = sum(pos.matches)
    negative[i] = sum(neg.matches)
  }
  return(data.frame(positive_score = positive,
                    negative_score = negative,
                    sentiment_score = score))
}

df <- all
```

As a result of data cleaning and preparation, this question is completed by visualizing the use of positive and negative words in tweets.

```
library(ggplot2) # create elegant data visualisations using the grammar of
graphics

output = getSentimentScore(df$text, pos, neg)
df$sentiment = output[, "sentiment_score"]
df$pos_sentiment = output[, "positive_score"]
df$neg_sentiment = output[, "negative_score"]
df$day = as.Date(cut(df$created, breaks = "day"))
df %>%
  group_by(day) %>%
  summarise(meanPos = mean(pos_sentiment), meanNeg = mean(neg_sentiment),
meanSent = mean(sentiment)) -> sentiment_byday

ggplot(sentiment_byday,aes(x=day)) +
  geom_line(aes(y=meanPos, colour="Positive words")) +
  geom_line(aes(y=meanNeg,colour="Negative words")) +
  scale_colour_manual(values=c(`Positive words`="#619CFF", `Negative
words`="#F8766D")) +
  theme(legend.justification = c(1, 1), legend.position = c(.95, .95),
legend.title=element_blank()) +
  xlab(label="Date") +
  ylab(label="Score")
```



As seen in the line graph formed as a result of sentiment analysis of tweets, the score of negative words is significantly higher than the score of positive words. This may indicate that Twitter users tend to use negative words more on issues such as rent, properties, house price and real estate in New York and California. This may also indicate that they have negative feelings for this subject.

All in all, in this paper, different approaches are performed to answer three different research questions. In the first research question, it is aimed to determine the emotions in the videos with sentiment analysis. Although the speakers had feelings such as fear and sadness, it is also observed that they had anticipation and trust. Also, positive emotion is the strongest emotion. It may be concluded as people are aware of the difficulties brought by the pandemic and believe that if the right steps are taken in time, everything will get better. They approach this situation constructively, not destructively. In the second study, topic modeling of various news on New York Times about New York and the housing market over time is made. It is observed that there are many similarities between the three identified topic groups. However, besides this, it is observed that the first topic summarized the relation between the tenant and the landlord. The second topic group summarized the rent increase related to the pandemic. Lastly, the third topic could be a topic that includes mathematical/numerical information. Finally, tweets including "rent", "house price", "real estate" and "properties" for New York and California is analyzed. Its aim is to understand what people think today as a result of price fluctuations for more than 1 year and to try to understand whether they still have problems with the rental price. As a result, the words "expensive", "dangerous", "cost" and "increase" became the most frequently used words. At the same time, negative words used in tweets outnumber positive words.

Limitations & Outlook

Although this study is summative and informative since it gives the chance to gain insights about the feelings of the people against the rapidly increasing rent prices for different reasons in recent years, it is still open to development in many ways.

First of all, instead of using only two videos for the first research question, a more detailed sentiment analysis could have been done by preparing a larger dataset containing street interviews on real estate and rental prices, informative videos prepared by video-sharing and social media platforms (YouTube, TikTok etc.) and broadcast news. Secondly, the news selected for the second research question is highly limited. It contains only 10 random news from New York Times online newspaper. Instead, news published by the national press as well as the local press in New York can be added. The topic groups that might be emerged as a result of this more detailed analysis may create new ideas. Maybe in the future, a program can be created to solve the problems respectively by prioritizing the subject topics. At the same time, the parameter for the number of topics and other metrics can be re-evaluated for the expanded dataset. Thirdly, for the last research question, sentiment analysis can be done where long-term change and its effect on the public can clearly be observed by obtaining the necessary permissions from Twitter. In this study, tweets between 6-9 days are used. With the extended analysis in terms of both data set and time range, it can be better summarized that this issue is an important situation that puts many people into trouble and requires an immediate action to see the results in long term (regulations, construction projects etc.).

If I had the chance to take this work to a further level, I would like to start street interviews myself and create a data set where I can get other insights for the sentiment analysis. While these studies reveal how people get affected by this, they can increase the motivation of the press and higher institutions to take precautions. Creating various statistical data and sharing findings can add more importance of that situation. Besides, Twitter is not the only place where ideas are shared. In addition to it, the research can be further developed by including data from social media platforms like Quora, Facebook, LinkedIn, Reddit etc.

where ideas can be published. Although the numerical representation of the change in the real estate market reveals the frightening picture, it is important whether the public is aware of the change, how they are affected by it and how they cope with it, and whether they believe in the change. For this reason, I think that reaching different audiences through different social media channels is one of the important ideas that will improve this study.

References

[1] Sitian, L., & Yichen, S. (2021, July 22). *The impact of the COVID-19 pandemic on the demand for density: Evidence from the U.S. housing market*. ScienceDirect.com. Retrieved January 26, 2023, from https://www.sciencedirect.com/sdfe/reader/pii/S0165176521002871/pdf

[2] Barron, J. (2022, August 8). *Why the Rent Is So High*. Retrieved February 26, 2023, from https://www.nytimes.com/2022/08/08/nyregion/why-the-rent-is-so-high.html

[3] [CNBC Make It]. (2022, August 20). *Why Rent In NYC Is Out Of Control Right Now* [Video]. YouTube. https://www.youtube.com/watch?v=c7tgsUv3UlU

[4] [CNBC]. (2022, May 9). *Why It's So Expensive To Live In The U.S.* [Video]. YouTube. https://www.youtube.com/watch?v=xANwNonI9aw

[5] Samuel, M. (2022, June 31). *The Elliman Report: June 2022 Manhattan, Brooklyn & Northwest Queens*. Retrieved January 27, 2023, from https://millersamuel.com/files/2022/07/Rental_06_2022.pdf?updated

[6] Cox, J. (2022, July 13). Inflation rose 9.1% in June, even more than expected, as consumer pressures intensify. *CNBC*.

[7] The Local. (2022, December 12). *Where rental prices are increasing the fastest in Germany*. The Local Germany. Retrieved January 27, 2023, from https://www.thelocal.de/20221212/eastern-germany-sees-biggest-spike-in-rental-prices-nationwide

[8] Jessel, E. (2022, October 21). *London rents: Why is the cost of renting spiralling out of control?* Evening Standard. Retrieved January 27, 2023, from https://www.standard.co.uk/homesandproperty/renting/london-rents-cost-of-renting-out-of-control-b1020357.html

[9] Wikimedia Foundation. (2023, January 2). *Economy of California*. Wikipedia. Retrieved January 27, 2023, from https://en.wikipedia.org/wiki/Economy_of_California

[10] Bing, L., & Minqing, H. (2004). *Mining and summarizing customer reviews*. University of Illinois Chicago. Retrieved January 27, 2023, from https://www.cs.uic.edu/~liub/publications/kdd04-revSummary.pdf