Final_assignment

2024-01-08

The running time of the code is long. Therefore, I have commented out the codes that have created the meta data sets in the code appendix. I have saved these data frames to csv files in my git repository through which it can be downloaded. The link to my repository is: https://github.com/aashikachoudhary22/Final_Assignment-MY472-.git (https://github.com/aashikachoudhary22/Final_Assignment-MY472-.git)

Introduction:

The research question aims to understand if any factors differentiate between British MPs who ask economic questions in comparison to MPs who ask health and welfare-related questions. In order to answer this question, I have looked at both oral and written questions asked by MPs and segmented the MPs on factors such as gender, party, minority status, region representation and region specific median wage. The key metric I have created in my analysis is the ratio of the total number of economic questions to health and welfare questions asked by MPs within each segment. This ratio enables us to understand if any groups of MPs tend to favor any particular category of question. I have assessed this metric over time between 2019 and 2020 to understand if the onset of covid-19 had caused any difference in the observation.

Data:

The functions I created extracted up to 100 oral questions and up to 100 written questions for each month between January 2019 and December 2020. I have assumed that the API has provided me with a random selection of questions from each month which ensures that the dataset used is representative.

In order to determine whether a question is related to health and welfare or the economy, I created a vocabulary of words for each of the two categories. If a word from a vocabulary occurred in a question then it was assigned to the relevant category. The vocabularies are a combination of some general words and others that had been in the news during 2019-2020. Eg: Brexit, inflation, bank rates, recession, oil prices, GPD, trade wars had accounted for the most economic conversations in these two years. For health and welfare, it was the welfare cuts of 2019, mental health, impact of digital platforms, knife crime, climate change, covid-19’s impact on hospitals, education, etc. I have not added the words “covid” or “pandemic” to any of the vocabularies because these words were used in the context of both economic and health and welfare-related conversations.

Next, I used the quanteda package to create a corpus from the questions obtained from the API. I then convert the corpus into a document matrix through which I look for the words mentioned from the vocabularies. I obtained additional MP data from the API including MP name, gender, party and constituency, and accessed some additional data sources - ethnic minority status of MPs via Wikipedia and median income for each constituency (2023) obtained via a file from the House of Commons Library.

I have 2 final datasets - one for economic questions and one for health and welfare questions across 2019 and 2020. Both of these also contain the relevant MP metadata. I then group by month and the necessary factors and calculate the ratio of number of economic questions asked to number of health and welfare questions asked, before creating the relevant plots.

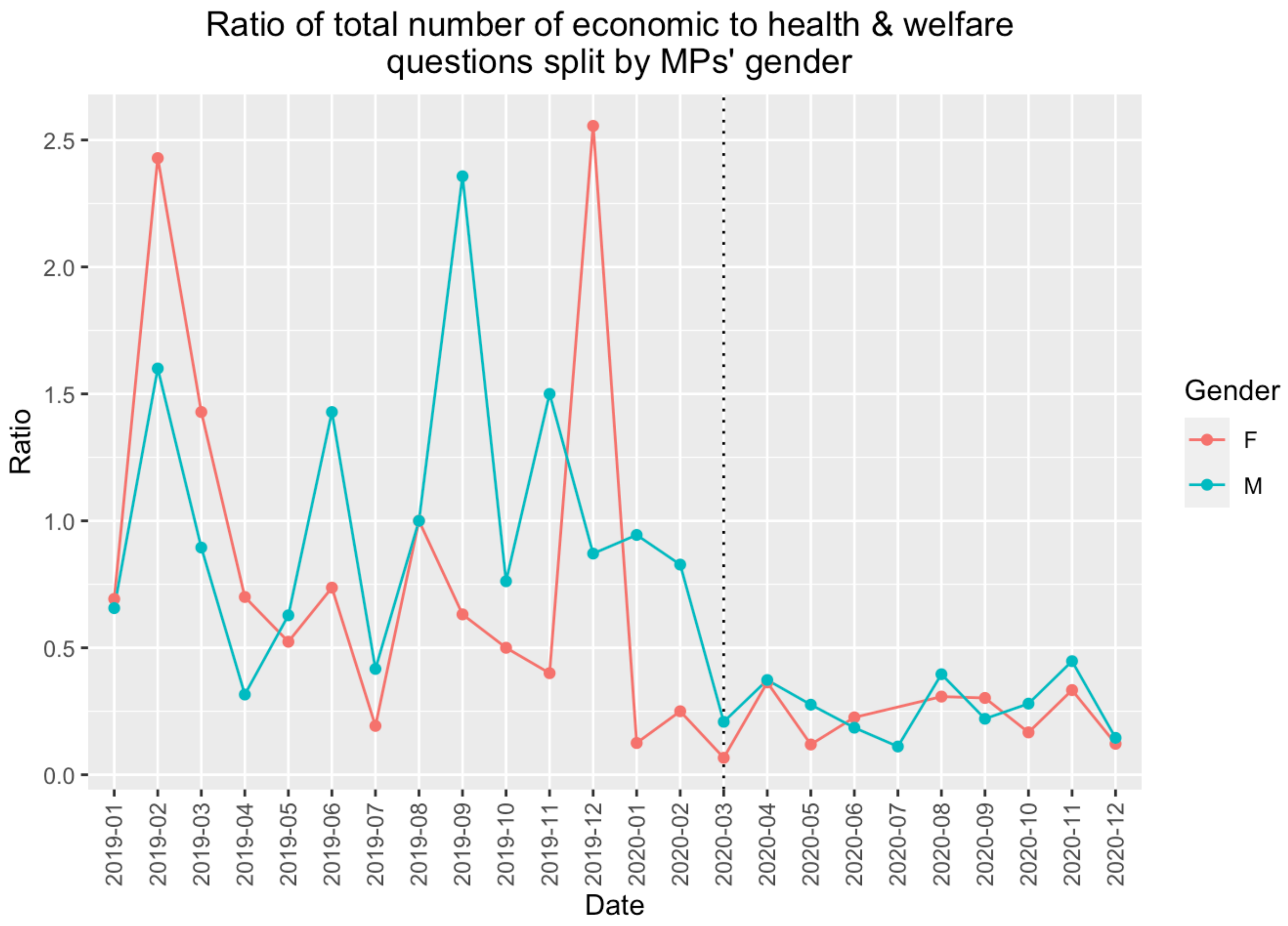
```
## [1] "Sample rows from economic data set"
```

```
##   credit poverty payment bank industry manufacturing fiscal growth money cost
## 1      0       0       0    0        0             0      0      0     0    0
##   market salary services rent infrastructure business job tax economy
## 1      0      1        0    0              0        0   0   0       0
##   unemployment loss budget workers income pension revenue pay trade cash rate
## 1            0    0      0       0      0       0       0   0     0    0    0
##   brexit inflation gdp deficit oil dollar surplus Member_ID    Date
## 1      0         0   0       0   0      0       0      4420 2019-01
##      Member_name Gender                   party                    constituency
## 1 Gavin Newlands      M Scottish National Party Paisley and Renfrewshire North
##   RegionName WageMedianConst WageMedianRegion Minority_indicator
## 1   Scotland             802              702                  0
```

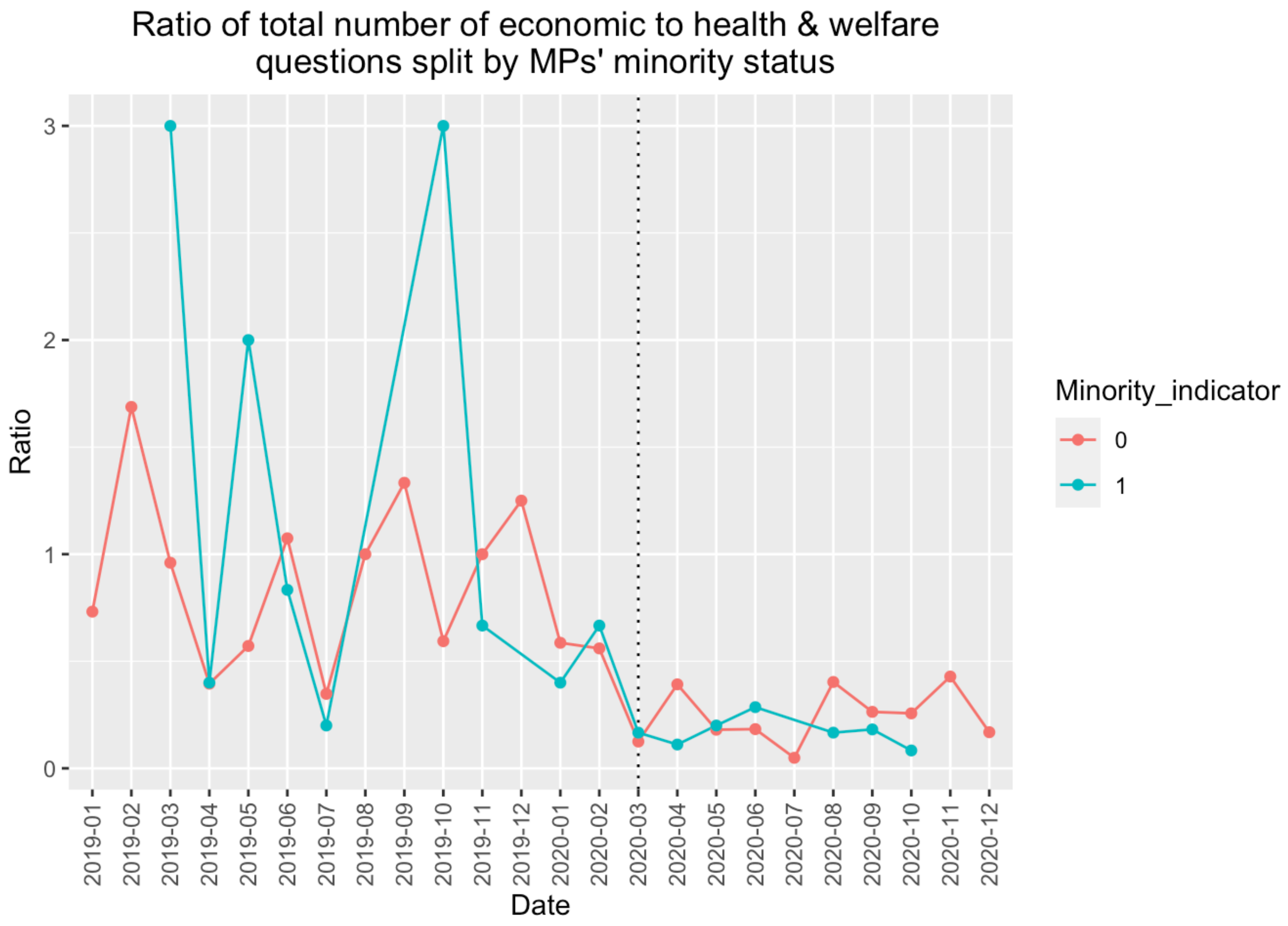
```
## [1] "Sample rows from health data set"
```

```
##   childcare children care gender women security abuse education sustainability
## 1         0        0    0      0     0        0     0         1              0
##   school service criminal safety prison girls homelessness housing plastic
## 1      0       0        0      0      0     0            0       0       0
##   theft immigration climate aid pollution patients mental university hospital
## 1     0           0       0   0         0        0      0          0        0
##   digital vulnerable disabled water welfare knife refugee old electricity
## 1       0          0        0     0       0     0       0   0           0
##   Member_ID    Date  Member_name Gender  party            constituency
## 1      4389 2019-01 Ruth Cadbury      F Labour Brentford and Isleworth
##   RegionName WageMedianConst WageMedianRegion Minority_indicator
## 1     London             849              796                  0
```

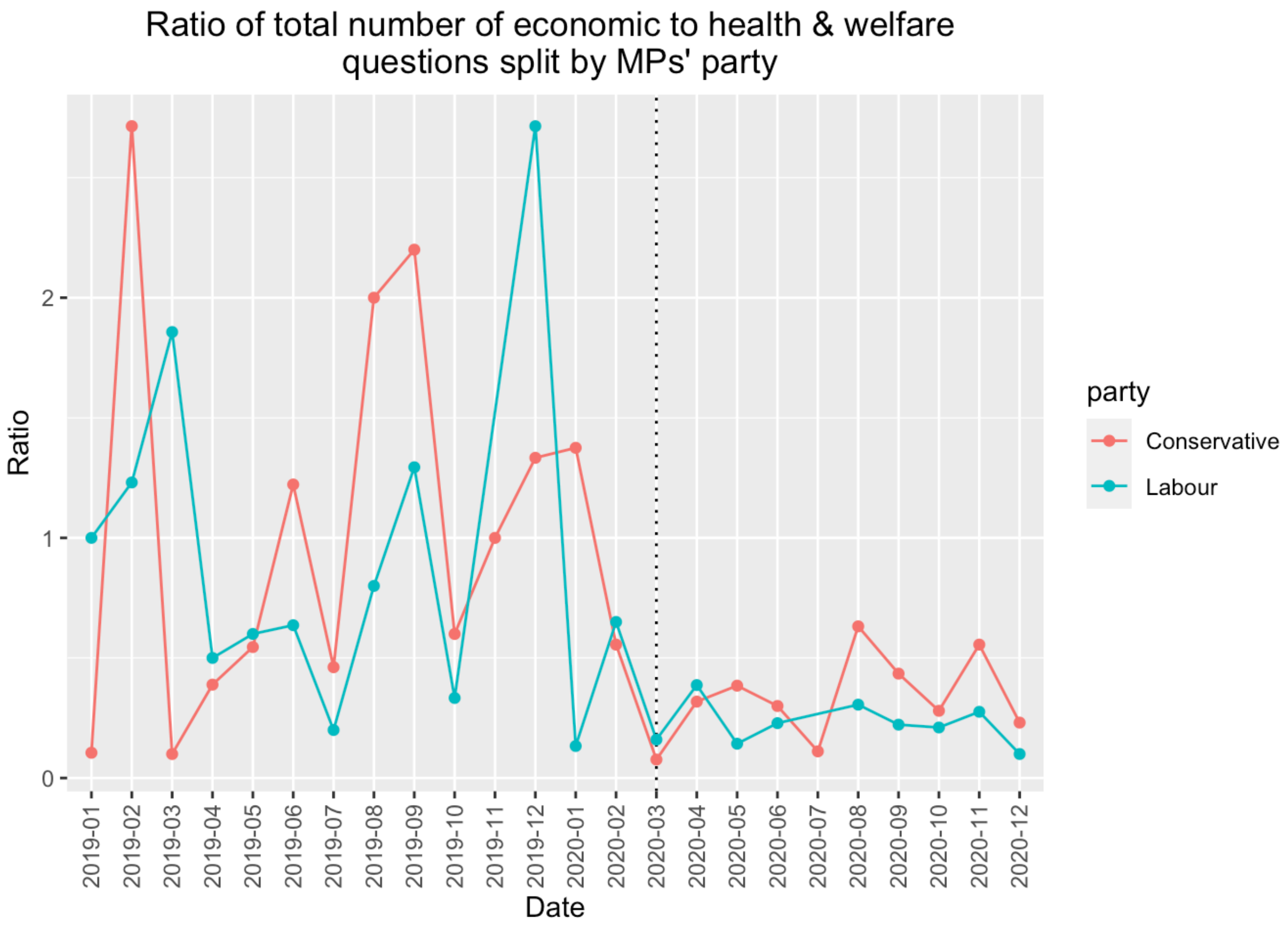
Analysis:



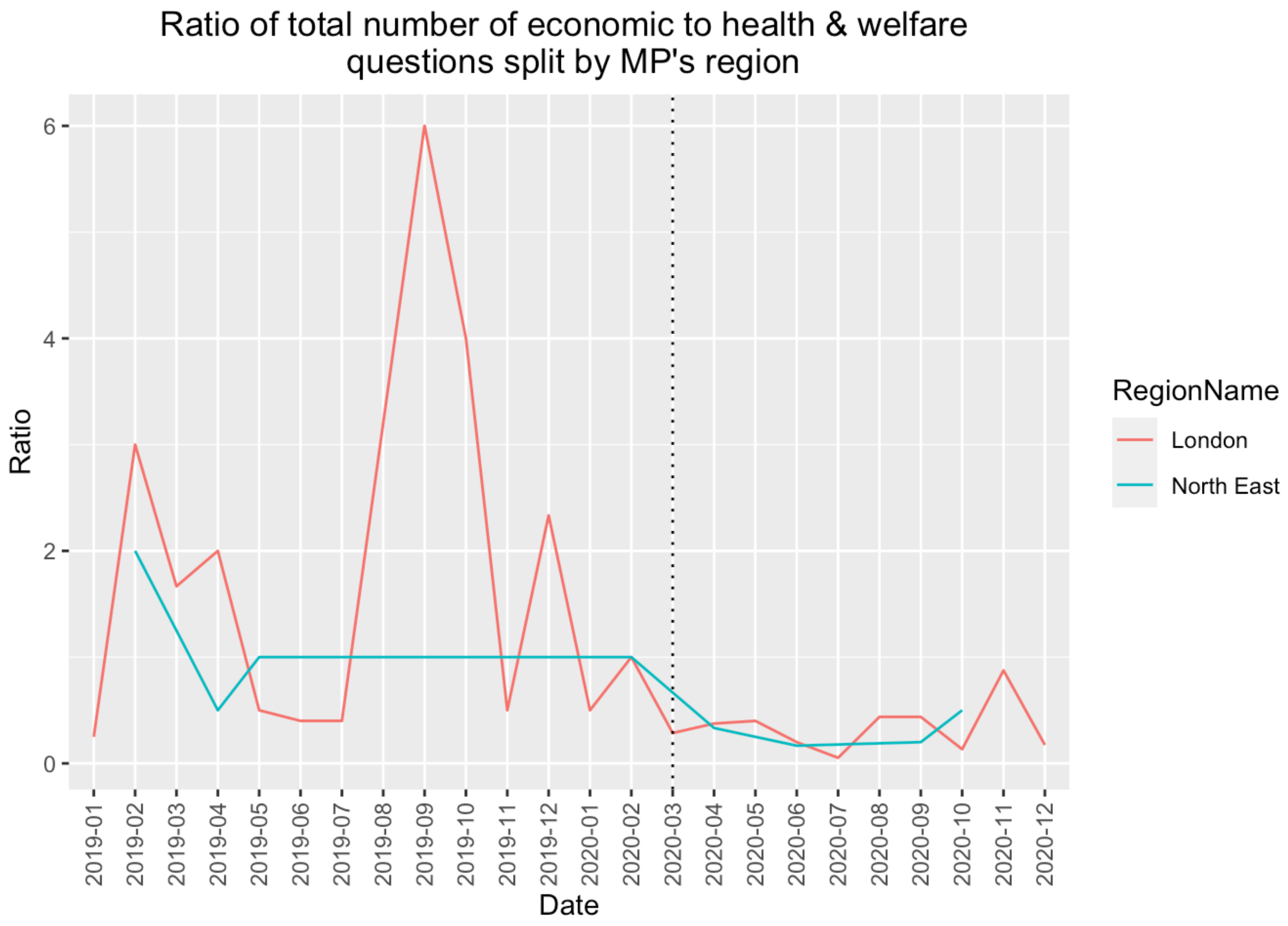
In the above plot, it is observed that men in general have marginally asked more economy-related questions in comparison to women. Post the onset of covid-19, both genders have actively asked more questions about health but we cannot see a big difference in the ratio of questions asked by the two genders.



In the plot above, prior to the onset of covid-19, there are no clear trends between the kind of questions asked by MPs with a minority status compared to MPs without one. However, post the onset of the pandemic, MPs with minority status have asked more health-related questions when compared to MPs without one.

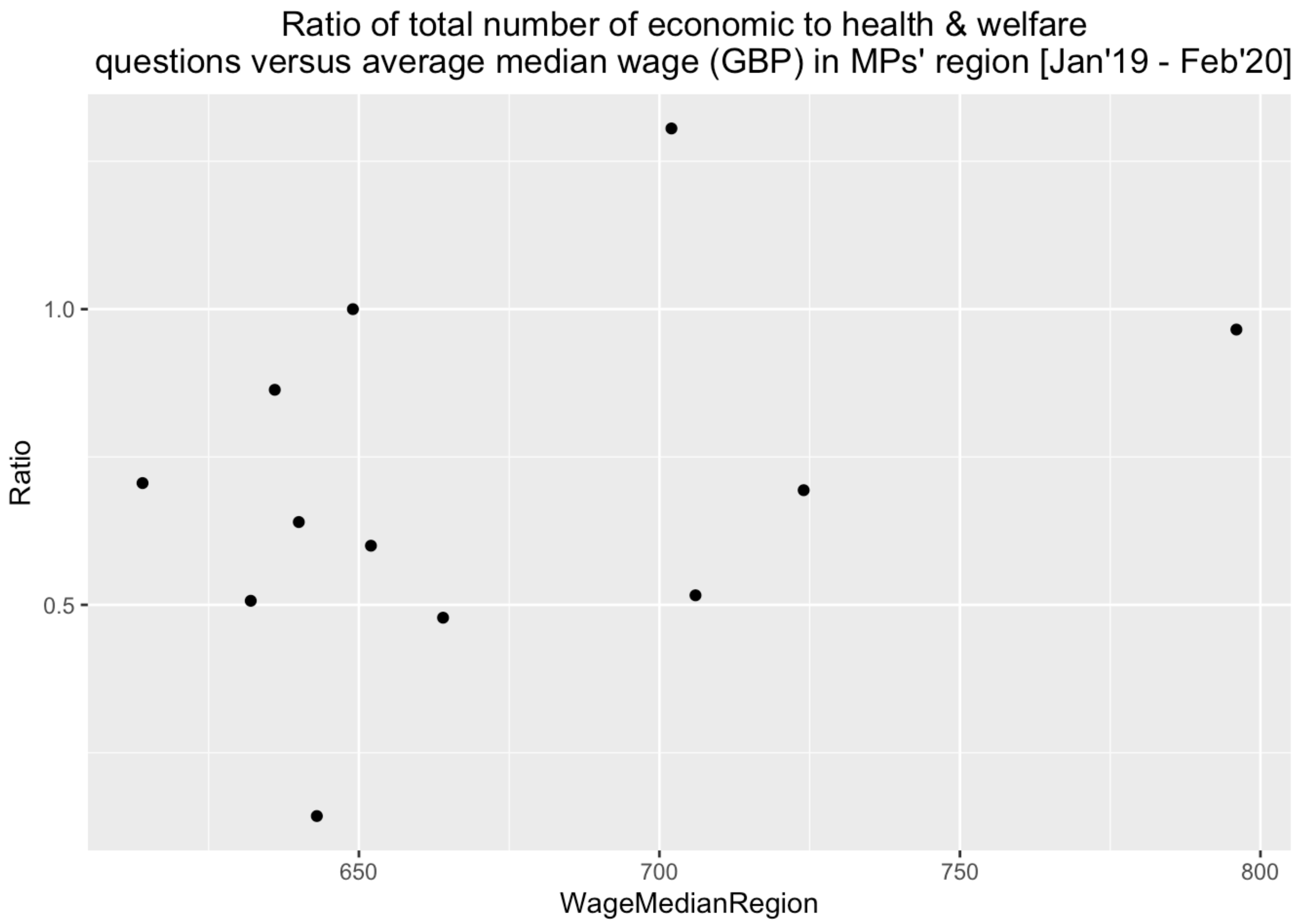


In the plot above, I have restricted the parties to just Conservative and Labour as these are two main parties in UK. It is evident from the plot that Conservative MPs have asked more economy-related questions in comparison to Labour MPs during both the time periods. This is perhaps intutive based off both party policies.

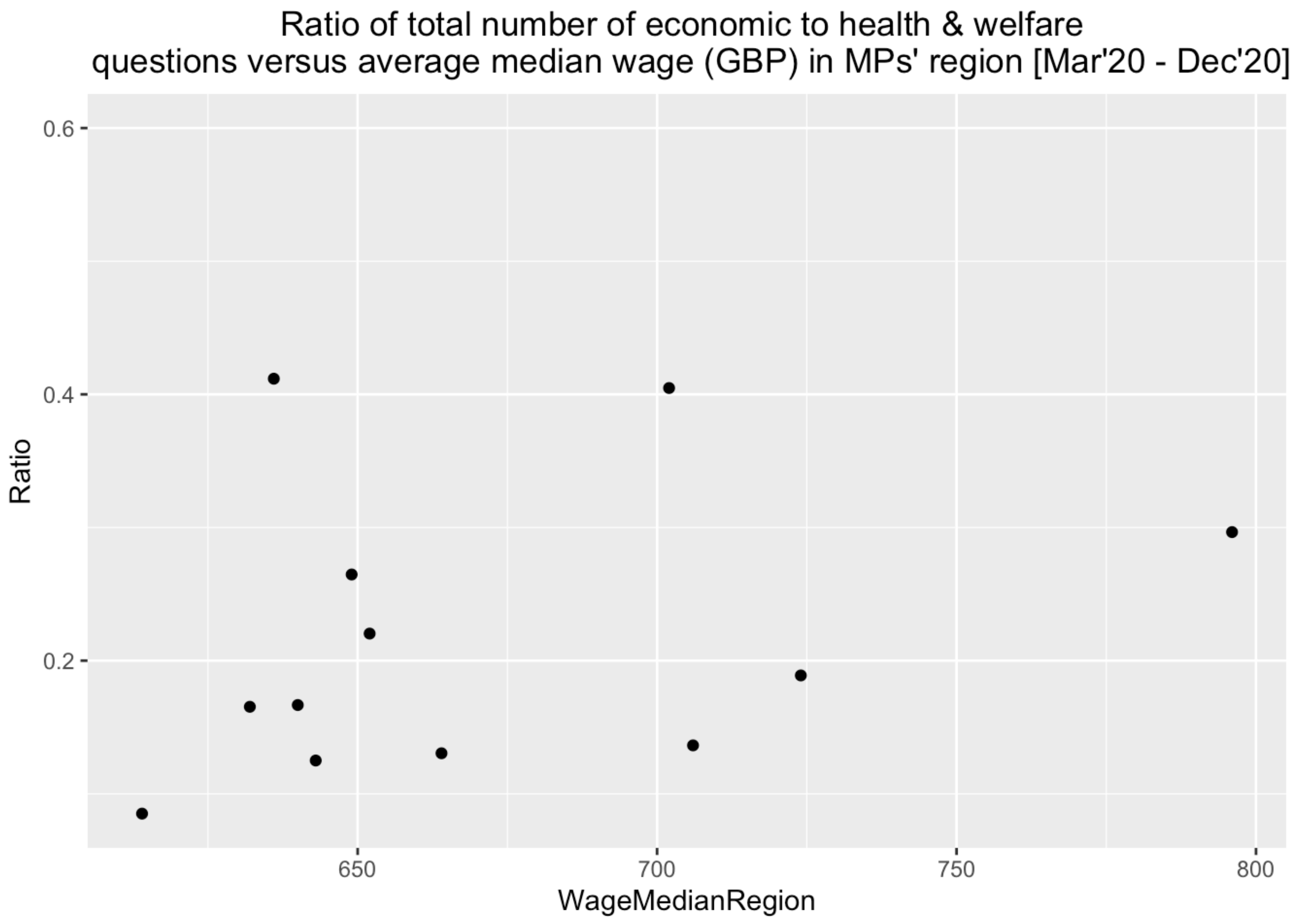


```
## geom_point: na.rm = FALSE
## stat_identity: na.rm = FALSE
## position_identity
```

In the above plot, I have restricted the regions to London and North East as London has the highest median wage and North East has the lowest median wage. It is clearly evident from the graph that MPs representing London have by far asked more economy-related questions throughout both the time-periods in comparison to MPs representing North East.



Through the plot, it can be seen that there is no clear correlation with regards to the kind of questions asked and the median wage of regions that the MPs represent. Most of the questions asked are closer to health and welfare category along with some outliers.



Through the plot, it can be seen that there is no clear correlation with regards to the kind of questions asked and the median wage of the regions that the MPs represent. Post the onset of covid-19, all of the questions asked are related to health and welfare category.

Through the analysis I can conclude that men, MPs hailing from ethnic minorities and MPs hailing from conservative party and MPs representing London constituencies are more inclined towards asking economic-related questions rather than health and welfare questions. Limitations: I have not conducted any statistically significant analysis. The observations would be more reliable if I had obtained a greater number of questions from the API.

Appendix: All code in this assignment

```
#setwd("/Users/aashikachoudhary/Desktop/MY472/Final_Assignment-MY472-")
#write.csv(economic_data, "economic_data.csv", row.names = FALSE)
#write.csv(health_data, "health_data.csv", row.names = FALSE)

economic_data_csv <- read.csv("economic_data.csv")
health_data_csv <- read.csv("health_data.csv")

# A function which takes a list of "start_date" and "end_date" as inputs to give a
list of oral_questions asked by British MPs
```

```
#oral_questions <- function(start_dates, end_dates) {
# Creating an empty list to store results for each date range
 # all_questions <- list()
  # For loop to iterate over dates
 # for (i in seq_along(start_dates)) {
    # Defining the url
   # url <- "https://oralquestionsandmotions-api.parliament.uk/oralquestions/list?
parameters.answeringDateStart="
    # Redefining the url along with pasting value from the url, the start and the
end date
   # url <- paste0(url, start_dates[i], "&parameters.answeringDateEnd=", end_date
s[i], "&parameters.take=100")
    # Converting the url into json
   # json.m <- fromJSON(url)

    # Check if 'Response' is present in json.m and has the expected data structure
   # if ("Response" %in% names(json.m) && is.data.frame(json.m$Response)) {
      # If the condition in the "if" statement is true then assigning data to the
following
    #  data <- json.m$Response[, c("QuestionText", "TabledWhen", "AskingMemberI
d")]
    #  all_questions[[i]] <- data
   # } else {
      # Handle unexpected structure or missing 'Response'
     # warning(paste("Unexpected structure in response for dates:", start_dates[
i], "to", end_dates[i]))
      #The i(th) elment of "all_questions" list is assigned to a vector containing
three NA values
     # all_questions[[i]] <- c(NA, NA, NA)
   # }
 # }
  # Filter out all elements which are not null and assigning it to "all_questions"
 # all_questions <- Filter(Negate(is.null), all_questions)
  # "all_questions" only contains non null elements and stop the execution of the
error message
  # if (length(all_questions) == 0) {
   # stop("No valid data found.")
 # }
  # returning the elements of "all questions" into a single dataframe
 # return(do.call(rbind, all_questions))
# }

# Oral Questions

# Create a sequence of dates from January 1st, 2019, to January 1st, 2020
# dates <- seq(as.Date("2019-01-01"), as.Date("2021-01-01"), by = "months")
# Extract the first and last day of each month
# start_date <- floor_date(dates, "month")
# end_date <- ceiling_date(dates + months(1) - days(1), "month")
# Call the questions function with vectors of start and end dates
# oral_result <- oral_questions(start_date, end_date)


# Written Question
# A function which takes a list of "start_date" and "end_date" as inputs to give a
list of written_questions asked by British MPs
# written_questions <- function(start_dates, end_dates) {
  # Creating an empty list to store results for each date range
 # all_results <- list()
  # For loop to iterates over dates
 # for (i in seq_along(start_dates)) {
  #  start_date <- start_dates[i]
   # end_date <- end_dates[i]
    # Defining the second url
   # url <- "https://questions-statements-api.parliament.uk/api/writtenquestions/q
uestions?tabledWhenFrom="
    # Redefining the url along with pasting value from the url, the start and the
end date
   # url <- paste0(url, start_date, "&tabledWhenTo=", end_date, "&questionStatus=A
llQuestions&take=100")
    # Converting the url into json
   # data <- fromJSON(url)
   # results <- data$results
   # value <- results$value
   # value <- value[, c("questionText", "askingMemberId", "dateTabled")]

    # Append the results to the list
   # all_results[[i]] <- value
#  }

  # Combine all results into a single data frame
  # final_result <- do.call(rbind, all_results)

 # return(final_result)
# }

# Call the questions function with vectors of start and end dates
# written_result <- written_questions(start_date, end_date)
# Reassigning the column names of "written_result" data frame
# written_result <- written_result %>%
# rename(
#  QuestionText = questionText,
 # AskingMemberId = askingMemberId,
#  TabledWhen = dateTabled
# )
# Creating a database called as "total_data" which binds oral_result and written_r
esult by row
# total_data <- rbind(oral_result, written_result)

# Convert into lower case
# total_data$QuestionText <- tolower(total_data$QuestionText)
# Creating a text corpus of column "QuestionText" from the "total_data" data fram
e.
# corpus <- corpus(total_data$QuestionText)
# Creating tokens of the "corpus" data frame
# corpus_tokens <- tokens(corpus)
# Converting tokensied corpus to document feature matrix
# dfm_result <- dfm(corpus_tokens)

# Creating a vocabulary of economic words
# economic_vocab <- c("economy", "trade", "credit", "dollar", "cash", "deficit", "
loss", "surplus", "business", "industry", "poverty", "money", "cost", "payment", "
market", "rent", "revenue", "oil", "infrastructure", "wages", "salary", "unemploym
ent", "pension", "workers", "job", " billion", "tax", "bank", "gdp", "fiscal", "bu
dget", "pay", "income", "brexit", "inflation", "growth", "manufacturing", "service
s", "rate", "recession", "stock")
# Creating a dictionary of economic_vocab words
# economic_questions <- dictionary(list(selection_economic_words = economic_voca
b))
# Looking up for specific words contained in "dfm_result" document from "economic_
questions" variable
# economic_questions_lookup <- dfm_lookup(dfm_result, economic_questions)
# Selecting specific words from the "dfm_result" data frame
# doc_economic_questions <- dfm_select(dfm_result, pattern = economic_vocab)
# Converting "economic_questions" document into a data frame
# economic_questions_data <- as.data.frame(as.matrix(doc_economic_questions))
# Creating a dataframe called as "df1" which binds economic_questions_data and Mem
ber_ID from "total data" data frame
# df1 <- cbind(economic_questions_data, Member_ID = total_data$AskingMemberId)
# Creating a dataframe called as "df" which binds df1 and TabledWhen column from "
total data" data frame
# df <- cbind(df1, Date = total_data$TabledWhen)
# Creating "economic_data" as the subset of "df" containing only words from econom
ic_vocab
# economic_data <- df[rowSums(df[, 1:ncol(economic_questions_data)] == 1) > 0, ]
# Changing the format of the Date column of the "economic_data" dataset and limiti
ng it only to year and month
# economic_data$Date <- substr(economic_data$Date, 1, 7)

# Removing irrelevant dates from the "economic_data" dataset
# dates_to_remove <- c("2018-12", "2021-01", "2021-02")
# economic_data <- subset(economic_data, !Date %in% dates_to_remove)


# Creating a vocabulary of health and welfare words
# health_vocab <- c("security", "aid", "hospital", "water", "education", "sanitati
on", "electricity", "care", "children", "old", "gender", "women", "equity", "patie
nts", "girls", "immigration", "sustainability", "childcare", "abuse", "criminal",
"prison", "service", "refugee", "homelessness", "safety", "vulnerable", "disable
d", "plastic", "climate", "theft", "doctor", "school", "welfare", "mental", "digit
al", "housing", "university", "water", "warm", "pollution", "knife")
# Creating a dictionary of health_vocab words
# heath_welfare_questions <- dictionary(list(selection_heath_words = health_voca
b))
# Looking up for specific words contained in "dfm_result" document from "health_qu
estions" variable
# heath_welfare_lookup <- dfm_lookup(dfm_result, heath_welfare_questions)
# Selecting specific words from the "dfm_result" data frame
# doc_heath_welfare <- dfm_select(dfm_result, pattern = health_vocab)
# Converting "health_questions" document into a data frame
# health_questions_data <- as.data.frame(as.matrix(doc_heath_welfare))
# Creating a data frame called as "df2" which binds economic_questions_data and Me
mber_ID from "total data" data frame
# df2 <- cbind(health_questions_data, Member_ID = total_data$AskingMemberId)
# Creating a dataframe called as "df_2" which binds df2 and TabledWhen column from
"total data" data frame
# df_2 <- cbind(df2, Date = total_data$TabledWhen)
# Creating "health_data" as the subset of "df_2" containing only health_vocab word
s
# health_data <- df_2[rowSums(df_2[, 1:ncol(health_questions_data)] == 1) > 0, ]
# Changing the format of the Date column of the "health_data" dataset and limiting
it only to year and month
# health_data$Date <- substr(health_data$Date, 1, 7)

# Removing irrelevant dates from the "health_data" dataset
# dates_to_remove <- c("2018-12", "2021-01", "2021-02")
# health_data <- subset(health_data, !Date %in% dates_to_remove)


#Creating a function called as "get_member_name" which gets the name of the parlia
ment member with the input of "member_id"
# get_member_name <- function(member_id) {
  # Constructing the API endpoint URL
 #  api_url <- paste0("https://members-api.parliament.uk/api/Members/", member_id)
  # Making the HTTP request
  # response <- GET(api_url)
  # Checking the HTTP status if it is successful
 # if (status_code(response) == 200) {
    # Making the content parse
   # member_info <- content(response, "parsed")
    # Extracting information from parsed "member_info"
   # member_info <- member_info$value
   # member_name <- member_info$nameDisplayAs
 # }
  # Pausing each execution by 0.5 seconds
  #Sys.sleep(0.5)
  # returning "member_name" as output of the function
# return(member_name)
# }
# Creating a function called as "get_member_gender" which gets the name of the par
liament member with the input of "member_id"
# get_member_gender <- function(member_id) {
  # Constructing the API endpoint URL
 # api_url <- paste0("https://members-api.parliament.uk/api/Members/", member_id)
  # Making the HTTP request
 # response <- GET(api_url)
  # Checking the HTTP status if it is successful
 # if (status_code(response) == 200) {
    # Making the content parse
 #   member_info <- content(response, "parsed")
    # Extracting information from parsed "member_info"
  #  member_info <- member_info$value
   # gender <- member_info$gender
  }
  # Pausing each execution by 0.5 seconds
 # Sys.sleep(0.5)
  # returning "gender" as output of the function
#  return(gender)
 # }

# Creating a function called as "get_member_party" which gets the name of the parl
iament member with the input of "member_id"
# get_member_party <- function(member_id) {
  # Constructing the API endpoint URL
 # api_url <- paste0("https://members-api.parliament.uk/api/Members/", member_id)
  # Making the HTTP request
  # response <- GET(api_url)
  # Checking the HTTP status if it is successful
 # if (status_code(response) == 200) {
    # Making the content parse
  #  member_info <- content(response, "parsed")
    # Extracting information from parsed "member_info"
   # member_info <- member_info$value
   # member <- member_info$latestParty
   # party <- member$name
 # }
  # Pausing each execution by 0.5 seconds
#  Sys.sleep(0.5)
  # returning "party" as output of the function
 # return(party)
# }

# Creating a function called as "get_member_constituency" which gets the name of t
he parliament member with the input of "member_id"
# get_member_constituency <- function(member_id) {
  # Constructing the API endpoint URL
 # api_url <- paste0("https://members-api.parliament.uk/api/Members/", member_id)
  # Making the HTTP request
 # response <- GET(api_url)
```

```
  # Extracting information from parsed "member_info"
 # if (status_code(response) == 200) {
    # Making the content parse
   # member_info <- content(response, "parsed")
    # Extracting information from parsed "member_info"
   # member_info <- member_info$value
   # member <- member_info$latestHouseMembership
     # constituency <- member$membershipFrom
 # }
  # Pausing each execution by 0.5 seconds
 # Sys.sleep(0.5)
  # returning "constituency" as output of the function
 # return(constituency)
# }

# Creating a new column in the economic_data data set which gets the member_name f
or the entire dataset by using "get_member_name" function
# economic_data$Member_name <- sapply(economic_data$Member_ID, function(x)get_memb
er_name(x))
# Creating a new column in the economic_data data set which gets gender for the en
tire dataset by using "get_member_gender" function
# economic_data$Gender <- sapply(economic_data$Member_ID, function(x)get_member_ge
nder(x))
# Creating a new column in the economic_data data set which gets the party for the
entire dataset by using "get_member_party" function
# economic_data$party <- sapply(economic_data$Member_ID, function(x)get_member_par
ty(x))
# Creating a new column in the economic_data data set which gets the constituency
for the entire dataset by using "get_member_constituency" function
# economic_data$constituency <- sapply(economic_data$Member_ID, function(x)get_mem
ber_constituency(x))

# Reading "wages.csv" document which contains information on the average wage of e
ach constituency of UK
# Extracting wages document from "https://commonslibrary.parliament.uk/constituenc
y-data-wages/" link
# average_income <- read.csv("Wages.csv")
# Renaming the constituency column in the "average_income" data set
# average_income <- average_income %>%
#  rename(constituency = ConstituencyName)
#Selecting relevant columns in "average_income" data set
# average_income <- average_income[, c(2, 4, 9, 10)]

# Joining "average_income" data set to"economic_data" data set by constituency
# economic_data <- left_join(economic_data, average_income, by = "constituency")

# Creating a new column in the "health_data" data set which gets the member_name f
or the entire dataset by using "get_member_name" function
# health_data$Member_name <- sapply(health_data$Member_ID, function(x)get_member_n
ame(x))
```

```
# Creating a new column in the "health_data" data set which gets gender for the en
tire dataset by using "get_member_gender" function
# health_data$Gender <- sapply(health_data$Member_ID, function(x)get_member_gende
r(x))
# Creating a new column in the "health_data" data set which gets the party for the
entire dataset by using "get_member_party" function
# health_data$party <- sapply(health_data$Member_ID, function(x)get_member_party(
x))
# Creating a new column in the "health_data" data set which gets the constituency
for the entire dataset by using "get_member_constituency" function
# health_data$constituency <- sapply(health_data$Member_ID, function(x)get_member_
constituency(x))

# Joining "average_income" data set to"health_data" data set by constituency
# health_data <- left_join(health_data, average_income, by = "constituency")

# Extracting the minority status of MPs in UK from wikipedia
# wikipedia_link <- "https://en.wikipedia.org/wiki/List_of_ethnic_minority_politic
ians_in_the_United_Kingdom"
# Converting the wikipedia_link into html_wikipedia
# html_wikipedia <- read_html(wikipedia_link)
# html_wikipedia
# Selecting tables present in the wikipedia page
# wiki_table <- html_table(html_wikipedia, fill = TRUE)
# Choosing the particular table with relavnt information on all MPs with a minorit
y status
# wiki_table_data <- data.frame(wiki_table[6])
# Deleting any extra number, brackets or alphabets contained in the name column of
the table
# wiki_table_data$Name <- gsub("[0-9]+|\\[|\\]", "", wiki_table_data$Name)
# Assigning "wiki_mps" as the dataset which has the names of the MPs with a minori
ty status
# wiki_mps <- wiki_table_data$Name

# Creating a new column in the "economic_data" data set with default value 0
# economic_data <- mutate(economic_data, Minority_indicator = "0")
# Setting MembershipIndicator to 1 for MPs present in the Wikipedia list
# economic_data$Minority_indicator[economic_data$Member_name %in% wiki_mps] <- "1"

# Creating a new column in the "health_data" data set with default value 0
# health_data <- mutate(health_data, Minority_indicator = "0")
# Setting MembershipIndicator to 1 for MPs present in the Wikipedia list
# health_data$Minority_indicator[health_data$Member_name %in% wiki_mps] <- "1"

#Gender
# Creating a new data set which contains the total count of questions asked by bot
h male and female as grouped by date
economic_totalgender_count <- economic_data_csv %>% group_by(Date, Gender) %>%
   summarise(num_economic_questions=n())
```

```
# Creating a new data set which contains the total count of questions asked by bot
h male and female as grouped by date
health_totalgender_count <- health_data_csv %>% group_by(Date, Gender) %>%
  summarise(num_health_questions=n())

# Merging total gender count of both economic and health data grouped by date and
gender into a new data set
economic_health_gender <- merge(economic_totalgender_count, health_totalgender_cou
nt, by = c("Date", "Gender"))
# Creating a new column in this data set which is a ratio of total number of econo
mic to health questions segregated by gender
economic_health_gender$Ratio <- economic_health_gender$num_economic_questions/econ
omic_health_gender$num_health_questions

# Creating a plot which has Date on x-axis, ration on y-axis and it is grouped by
gender
ggplot(economic_health_gender, aes(x = Date, y = Ratio, col = Gender, group = Gend
er)) +
  geom_line()+
  theme(axis.text.x = element_text(angle=90, vjust=.5, hjust=1)) +
  geom_vline(xintercept = "2020-03", linetype = "dotted", color = "black") +
  ggtitle("Ratio of total number of economic to health & welfare \n questions spli
t by MPs' gender") +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_point()


#Minority
# Creating a new data set which contains the total count of questions asked by bot
h minority MPs as grouped by date
economic_totalminority_count <- economic_data_csv %>% group_by(Date, Minority_indi
cator) %>%
  summarise(num_economic_questions=n())

# Creating a new data set which contains the total count of questions asked by bot
h minority MPs as grouped by date
health_totalminority_count <- health_data_csv %>% group_by(Date, Minority_indicato
r) %>%
  summarise(num_health_questions=n())

# Merging total minority count of both economic and health data grouped by date an
d minority into a new data set
economic_health_minority <- merge(economic_totalminority_count, health_totalminori
ty_count, by = c("Date", "Minority_indicator"))
# Creating a new column in this data set which is a ratio of total number of econo
mic to health questions segregated by minority
economic_health_minority$Ratio <- economic_health_minority$num_economic_questions/
economic_health_minority$num_health_questions
economic_health_minority$Minority_indicator <- as.character(economic_health_minori
ty$Minority_indicator)
```

```
# Creating a plot which has Date on x-axis, ration on y-axis and is classified by
minority
ggplot(economic_health_minority, aes(x = Date, y = Ratio, col = Minority_indicato
r, group = Minority_indicator)) +
  geom_line()+
  theme(axis.text.x = element_text(angle=90, vjust=.5, hjust=1)) +
  geom_vline(xintercept = "2020-03", linetype = "dotted", color = "black") +
  ggtitle("Ratio of total number of economic to health & welfare \n questions spli
t by MPs' minority status") +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_point()


#Party
# Creating a new data set which contains the total count of questions asked by bot
h minority MPs as grouped by party where I have choosen only Labour and Conservati
ve parties
economic_totalparty_count <- economic_data_csv %>% filter(party == "Labour" | part
y == "Conservative") %>% group_by(Date, party) %>%
  summarise(num_economic_questions=n())

# Creating a new data set which contains the total count of questions asked by bot
h minority MPs as grouped by party where I have choosen only Labour and Conservati
ve parties
health_totalparty_count <- health_data_csv %>% filter(party == "Labour" | party ==
"Conservative") %>% group_by(Date, party) %>%
  summarise(num_health_questions=n())

# Merging total party count of both economic and health data grouped by date and p
arty into a new data set
economic_health_party <- merge(economic_totalparty_count, health_totalparty_count,
by = c("Date", "party"))
# Creating a new column in this data set which is a ratio of total number of econo
mic to health questions segregated by party
economic_health_party$Ratio <- economic_health_party$num_economic_questions/econom
ic_health_party$num_health_questions

# Creating a plot which has Date on x-axis, ration on y-axis and is classified by
party
ggplot(economic_health_party, aes(x = Date, y = Ratio, col = party, group = part
y)) +
  geom_line()+
  theme(axis.text.x = element_text(angle=90, vjust=.5, hjust=1)) +
  geom_vline(xintercept = "2020-03", linetype = "dotted", color = "black") +
  ggtitle("Ratio of total number of economic to health & welfare \n questions spli
t by MPs' party") +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_point()
```

```
# Pre-covid Region
# Assigning the values of dates as 1 or 0 based on the specification
economic_data_csv$covid_period <- with(economic_data_csv, ifelse(Date <"2020-03",
"0",
                                                              ifelse(Date >="2020-0
3", "1", "1")))

health_data_csv$covid_period <- with(health_data_csv, ifelse(Date <"2020-03", "0",
                                                          ifelse(Date >="2020-03",
"1", "1")))

# Creating a new data set which contains the total count of questions asked by bot
h male and female as grouped by date
economic_totalregion_count <- economic_data_csv %>% filter(covid_period == "0") %
>%  group_by(RegionName, WageMedianRegion) %>%
  summarise(num_economic_questions=n())

# Creating a new data set which contains the total count of questions asked by bot
h male and female as grouped by date
health_totalregion_count <- health_data_csv %>% filter(covid_period == "0") %>% gr
oup_by(RegionName, WageMedianRegion) %>%
  summarise(num_health_questions=n())

# Merging total gender count of both economic and health data grouped by date and
gender into a new data set
economic_health_region <- merge(economic_totalregion_count, health_totalregion_cou
nt, by = c("RegionName", "WageMedianRegion"))
# Creating a new column in this data set which is a ratio of total number of econo
mic to health questions segregated by gender
economic_health_region$Ratio <- economic_health_region$num_economic_questions/econ
omic_health_region$num_health_questions

# Creating a plot which has Date on x-axis, ration on y-axis and it is grouped by
gender
ggplot(economic_health_region) +
  geom_point(aes(x = WageMedianRegion, y = Ratio))+
  ggtitle("Ratio of total number of economic to health & welfare \n questions vers
us average median wage (GBP) in MPs' region [Jan 2019 - Feb 2020]") +
theme(plot.title = element_text(hjust = 0.5))


# Post-covid Region
# Creating a new data set which contains the total count of questions asked by bot
h male and female as grouped by date
economic_totalregion_count <- economic_data_csv %>% filter(covid_period == "1") %
>%  group_by(RegionName, WageMedianRegion) %>%
  summarise(num_economic_questions=n())

# Creating a new data set which contains the total count of questions asked by bot
```

```
h male and female as grouped by date
health_totalregion_count <- health_data_csv %>% filter(covid_period == "1") %>% gr
oup_by(RegionName, WageMedianRegion) %>%
  summarise(num_health_questions=n())

# Merging total gender count of both economic and health data grouped by date and
gender into a new data set
economic_health_region <- merge(economic_totalregion_count, health_totalregion_cou
nt, by = c("RegionName", "WageMedianRegion"))
# Creating a new column in this data set which is a ratio of total number of econo
mic to health questions segregated by gender
economic_health_region$Ratio <- economic_health_region$num_economic_questions/econ
omic_health_region$num_health_questions

# Creating a plot which has Date on x-axis, ration on y-axis and it is grouped by
gender
ggplot(economic_health_region) +
  geom_point(aes(x = WageMedianRegion, y = Ratio))+
  ggtitle("Ratio of total number of economic to health & welfare \n questions vers
us average median wage (GBP) in MPs' region [March 2020 - December 2020]") +
  theme(plot.title = element_text(hjust = 0.5))



# Region
economic_totalregion_count <- economic_data_csv %>% filter(RegionName == "London"
| RegionName == "North East") %>% group_by(Date, RegionName) %>%
  summarise(num_economic_questions=n())

# Creating a new data set which contains the total count of questions asked by bot
h male and female as grouped by date
health_totalregion_count <- health_data_csv %>% filter(RegionName == "London" | Re
gionName == "North East") %>% group_by(Date, RegionName) %>%
  summarise(num_health_questions=n())

# Merging total gender count of both economic and health data grouped by date and
gender into a new data set
economic_health_region <- merge(economic_totalregion_count, health_totalregion_cou
nt, by = c("Date", "RegionName"))
# Creating a new column in this data set which is a ratio of total number of econo
mic to health questions segregated by gender
economic_health_region$Ratio <- economic_health_region$num_economic_questions/econ
omic_health_region$num_health_questions

# Creating a plot which has Date on x-axis, ration on y-axis and it is grouped by
gender
ggplot(economic_health_region, aes(x = Date, y = Ratio, col = RegionName, group =
RegionName)) +
  geom_line()+
  theme(axis.text.x = element_text(angle=90, vjust=.5, hjust=1)) +
  geom_vline(xintercept = "2020-03", linetype = "dotted", color = "black") +
  ggtitle("Ratio of total number of economic to health & welfare \n questions spli
t by MP's region") +
  theme(plot.title = element_text(hjust = 0.5))
  geom_point()
```