

# R Group project

EU Group1

1/18/2021

- 1 Introduction to the Data Sets
- 2 Data tidy
  - 2.1 Survey—Demographic Items
  - 2.2 Survey—Other Items
  - 2.3 Content Analysis
- 3 Visualisations
  - 3.1 Lilian's Visualizaton
  - 3.2 Astrid's Visualizaton
  - 3.3 Janice's Visualisation
- 4 Bibliography

## 1 Introduction to the Data Sets

The present projects contains two datasets from the 2014 European Election Campaign Study a survey and a content analysis data set.

The survey was carried out in the Netherlands preceding the 2014 European Parliamentary elections. The survey was conducted in four waves. The first six months before, the second four months before and the third one month before the elections while the fourth waves was conducted immediately after the elections. The sample participants were drawn from the TNS NIPO database. The researchers enforced quotas on age, gender and education. The response rate for all waves was around 80% for each of the four waves around 1500 participants participated in the survey. The survey was conducted online.

Re-coded variables include apart from the demographic information (such as income, education and age among others) also other single measurements, such as the big five personality traits, trust towards the media, the vote certainty and vote intention. Several variables were repeatedly measured like a battery of items asking about the attitudes towards the EU and political knowledge which were both across 4 waves. Other variables like campaign exposure or campaign attention were assessed over 2 waves.

The content analysis data set is an analysis of newspapers, tv news and one online website (nu.nl) resulting in a total of 8 outlets being considered. The sampling was split into two periods, the first in December 2013 to April 2014 corresponds to the waves 1,2 and 3 of the survey.

For the essential variables (primary topic of the story and the emotion displayed among others) the front page was considered. The first five stories about the EU in each outlet were analyzed for every third day. In the later, second period (closer to the election) between April and May 2014 all stories about the EU in every outlet were considered for every second day. This time period corresponds to waves 3 and 4 in the survey. Some variables such as the type of story were only coded for newspaper stories, while others like the length of the news item were just coded for TV news stories. The coding included topics about the economy, the location, actors, attitudes of the media towards the EU and framing variables.

## 2 Data tidy

**Survey Dataset** - In This section we split the survey dataset to 2 main tibbles: demo\_EU and time\_EU. - The reason why we need to split them is that we have variables that are repeatedly measured over the 4 waves and demographic variables that are only measured once. This makes the main table messy and difficult to run analysis. - To split the tibbles and make sure they can be joined together again, we need primary keys and foreign keys. We decide to use respondent number as the primary key in demo\_EU, as it uniquely defines each participant.

Participants and waves together are the primary keys in time\_EU, as we need them both to uniquely define responses per participant per wave.

**Content Dataset** - In the content analysis dataset, we are only focusing on newstories in the Netherlands. So the unit is clear. We did not have to split the dataset into tibbles, but only selected the variables we are interested in.

```
#Load packages
library(tidyverse)
library(lubridate)
library(plotly)
library(htmlwidgets)
library(nanajar)
library(papaja)
```

## 2.1 Survey—Demographic Items

- In this section we will clean the demographic variables that are only measured once.
- We are mostly recoding the variables so that 1) their names make more sense 2) their values are unified and easy to run analysis on.

```
#Load data file
EU <- read_csv("All waves GENERAL.csv")
content <- read_csv("Dataset MCA EPE 2014 NL FINAL.csv")
```

```
#Select demographic variables
demo_EU <- EU %>%
  select(RESPNR, GSL, LFT, OPLA, NIELSENCBS, w1_q23, w1_q58_1:w1_q58_10, w1_q65, w2_q27_1:w2_q28_11, w4_q6)
#Rename the selected variables so we can see them more clearly when doing data wrangling.
namekey <- c(RESPNR = "RESPNR",
             GSL = "gender",
             LFT = "age",
             OPLA = "edu",
             NIELSENCBS = "region",
             w1_q23 = "past_voting",
             w1_q58_1 = "personality1",
             w1_q58_2 = "personality2",
             w1_q58_3 = "personality3",
             w1_q58_4 = "personality4",
             w1_q58_5 = "personality5",
             w1_q58_6 = "personality6",
             w1_q58_7 = "personality7",
             w1_q58_8 = "personality8",
             w1_q58_9 = "personality9",
             w1_q58_10 = "personality10",
             w1_q65 = "income",
             w2_q27_1 = "med_tru_tv1",
             w2_q27_2 = "med_tru_tv2",
             w2_q27_3 = "med_tru_tv3",
             w2_q27_4 = "med_tru_tv4",
             w2_q27_5 = "med_tru_tv5",
             w2_q27_6 = "med_tru_tv6",
             w2_q27_7 = "med_tru_tv7",
             w2_q27_8 = "med_tru_tv8",
             w2_q27_9 = "med_tru_tv9",
             w2_q28_1 = "med_tru_np1",
             w2_q28_2 = "med_tru_np2",
             w2_q28_3 = "med_tru_np3",
             w2_q28_4 = "med_tru_np4",
             w2_q28_5 = "med_tru_np5",
             w2_q28_6 = "med_tru_np6",
             w2_q28_7 = "med_tru_np7",
             w2_q28_8 = "med_tru_np8",
             w2_q28_9 = "med_tru_np9",
             w2_q28_10 = "med_tru_np10",
             w2_q28_11 = "med_tru_np11",
             w4_q6 = "turnout_EU")

names(demo_EU) <- namekey[names(demo_EU)]
```

### 2.1.1 Gender

```
#Change gender to English.
demo_EU <- demo_EU %>%
  mutate(gender = recode(gender,
                         "Vrouw" = "female",
                         "Man" = "male"))
```

## 2.1.2 Education

Recode education to numbers according to the following rules:

- Geen onderwijs/Basisonderwijs = 1,
- LBO/VBO/VMBO (kader- en beroepsgerichte leerweg)/MBO 1 = 2,
- MAVO/eerste 3 jaar HAVO en VWO/VMBO (theoretische en gemengde leerweg) = 3,
- MBO 2, 3, 4 of MBO oude structuur = 4,
- HAVO en VWO bovenbouw = 5,
- HBO-/WO-propedeuse/HBO-/WO-bachelor of kandidaats = 6,
- HBO-/WO-master of doctoraal = 7.

```
demo_EU <- demo_EU %>%
  mutate(edu = recode(edu,
    "Geen onderwijs/Basisonderwijs" = "1",
    "LBO/VBO/VMBO (kader- en beroepsgerichte leerweg)/MBO 1" = "2",
    "MAVO/eerste 3 jaar HAVO en VWO/VMBO (theoretische en gemengde leerweg)" = "3",
    "MBO 2, 3, 4 of MBO oude structuur" = "4",
    "HAVO en VWO bovenbouw" = "5",
    "HBO-/WO-propedeuse/HBO-/WO-bachelor of kandidaats" = "6",
    "HBO-/WO-master of doctoraal" = "7"))
```

## 2.1.3 Region

We want to just keep the major parts of the region so it would be easier for us to explore the data, and we also want to translate the region name to English.

Our recode is like this:

- Drie grote gemeenten (Amsterdam, Rotterdam, Den Haag) = Three large municipalities,
- West (Utrecht, Noord-Holland, Zuid-Holland excl. drie grote gemeenten en randgemeenten) = West,
- Noord (Groningen, Friesland, Drenthe) = North,
- Oost (Overijssel, Gelderland, Flevoland) = East,
- Zuid (Zeeland, Noord-Brabant, Limburg) = South,
- Randgemeenten (Amstelveen, Diemen, Landsmeer, Ouder-Amstel, Ridderkerk, Barendrecht, Albrandwaard, Krimpen a/d IJssel, C = Peripheral municipalities.

```
demo_EU <- demo_EU %>%
  mutate(region = recode(region,
    "Drie grote gemeenten (Amsterdam, Rotterdam, Den Haag)" = "Three large municipalities",
    "West (Utrecht, Noord-Holland, Zuid-Holland excl. drie grote gemeenten en randgemeenten)" = "West",
    "Noord (Groningen, Friesland, Drenthe)" = "North",
    "Oost (Overijssel, Gelderland, Flevoland)" = "East",
    "Zuid (Zeeland, Noord-Brabant, Limburg)" = "South",
    "Randgemeenten (Amstelveen, Diemen, Landsmeer, Ouder-Amstel, Ridderkerk, Barendrecht, Albrandwaard, Krimpen a/d IJssel, C" = "Peripheral municipalities"))
```

## 2.1.4 Past voting

Past voting behaviour should also be recoded into numeric values in order to make calculations easier.

```
#Recode this variable according to codebook.
demo_EU <- demo_EU %>%
  mutate(past_voting = recode(past_voting,
    "Elke keer" = "1",
    "Bijna elke keer" = "2",
    "Meer dan de helft van de keren" = "3",
    "Ongeveer de helft van de keren" = "4",
    "Minder dan de helft van de keren" = "5",
    "Vrijwel nooit" = "6",
    "Nooit" = "7",
    "Ik mocht nog nooit stemmen" = "8"))
```

## 2.1.5 Big 5 personality

```
#We need to recode `1 helemaal niet` = `1`, `3 een beetje` = `3`, `5 in redelijke mate` = `5`, `7 zeer goed` = `7`.
#Recode this variable (for the whole 10 questions) according to codebook.
demo_EU <- demo_EU %>%
  mutate_at(c("personality1", "personality2", "personality3", "personality4", "personality5", "personality6", "personality7", "personality8", "personality9", "personality10"),
    funs(parse_number(.,)))
```

## 2.1.6 Income

```
#Recode this variable according to codebook.
demo_EU <- demo_EU %>%
  mutate(income = recode(income,
    "Minder dan €150" = "1",
    "€150 tot €300" = "2",
    "€301 tot €500" = "3",
    "€501 tot €1000" = "4",
    "€1001 tot €1500" = "5",
    "€1501 tot €2000" = "6",
    "€2001 tot €2500" = "7",
    "€2501 tot €3000" = "8",
    "€3001 tot €5000" = "9",
    "€5001 tot €7500" = "10",
    "€7501 tot €10000" = "11",
    "€10001 of meer" = "12",
    "Wil geen antwoord geven" = "13",
    "weet niet" = "14"))
```

## 2.1.7 Media trust-TV

We need to recode

- 1 helemaal geen vertrouwen = 1,
- 7 veel vertrouwen = 7,
- ik ken deze nieuwsbron niet = 0.

```
demo_EU <- demo_EU %>%
  mutate_at(c("med_tru_tv1", "med_tru_tv2", "med_tru_tv3", "med_tru_tv4", "med_tru_tv5", "med_tru_tv6", "med_tru_tv7", "med_tru_tv8", "med_tru_tv9"),
    funs(parse_number(.,)))
```

## 2.1.8 Media trust-np, radio, internet

We need to recode

- 1 helemaal geen vertrouwen = 1,
- 7 veel vertrouwen = 7,
- ik ken deze nieuwsbron niet = 0.

```
demo_EU <- demo_EU %>%
  mutate_at(c("med_tru_np1", "med_tru_np2", "med_tru_np3", "med_tru_np4", "med_tru_np5", "med_tru_np6", "med_tru_np7", "med_tru_np8", "med_tru_np9", "med_tru_np10", "med_tru_np11"),
    funs(parse_number(.,)))
```

## 2.1.9 Voting Behavior: Turnout EU wave 4

Because "turnout EU" is only measured in one wave, we put this variable in demographic tibble.

```
#Clean Turnout EU wave 4(w4_q6)

#recode into factor with levels according to the documentation
#[1. Ik heb op 22 mei niet gestemd voor de Europese Parlementsverkiezingen.]
#[2. Ik dacht eraan te stemmen, maar deed het dit keer niet.]
#[3. Ik stem meestal wel, maar dit keer niet.]
#[4. Ik heb op 22 mei gestemd voor de Europese Parlementsverkiezingen.]

demo_EU<-
  demo_EU %>%
  mutate(
    turnout_EU = recode_factor(
      turnout_EU,
      "Ik heb op 22 mei niet gestemd voor de Europese Parlementsverkiezingen." = 1,
      "Ik dacht eraan te stemmen, maar deed het dit keer niet." = 2,
      "Ik stem meestal wel, maar dit keer niet." = 3,
      "Ik heb op 22 mei gestemd voor de Europese Parlementsverkiezingen." = 4
    )
  )

write.csv(demo_EU, "demo_EU.csv")
```

demo\_EU is cleaned

## 2.2 Survey — Other Items

- In this section, we select items that are measured in multiple waves.
- The variables have many problems in common. For example, we need to pivot\_longer so that the “w1/2/3/4” wave information their variable names are split out to its own column. Secondly, we need to recode the frequency values so that they are numbers and easy to run analysis on.
- We therefore created functions to deal with these common problems.

```
#Select needed variables.
time_EU <- EU %>%
  select(RESPNR,
    #Interest
    w1_q3_1:w1_q3_6, w2_q3_1:w2_q3_6, w3_q3_1:w3_q3_6, w4_q3_1:w4_q3_6,
    #Government satisfaction (General)
    w1_q6, w2_q5, w3_q9, w4_q11,
    #Government satisfaction (Specific Issues)
    w1_q7_1:w1_q7_4, w2_q6_1:w2_q6_4, w3_q10_1:w3_q10_4, w4_q12_1:w4_q12_4,
    #Political cynicism 4 waves
    w1_q10_1:w1_q10_4, w2_q7_1:w2_q7_4, w3_q11_1:w3_q11_4, w4_q13_1:w4_q13_4,
    #Political knowledge
    w1_q18, w2_q14, w3_q19, w4_q21,
    w1_q19, w2_q15, w3_q20, w4_q22,
    w1_q20, w2_q16, w3_q21, w4_q23,
    w1_q21, w2_q17, w3_q22, w4_q24,
    w1_q22, w2_q18, w3_q23, w4_q25,
    #EU attitudes-1
    w1_q26_1:w1_q26_9, w2_q22_1:w2_q22_9, w3_q28_1:w3_q28_9, w4_q31_1:w4_q31_9,
    #EU attitudes-2
    w1_q27_1:w1_q27_9, w2_q23_1:w2_q23_9, w3_q29_1:w3_q29_9, w4_q32_1:w4_q32_9,
    #Interpersonal communication: Politics & EU Politics
    w1_q44, w2_q41, w3_q40, w4_q44,
    w1_q45, w2_q42, w3_q41, w4_q45,
    w1_q46, w2_q43, w3_q42, w4_q46,
    #Self reported media exposure [TV] [Newspaper] [Internet]
    w1_q49_1:w1_q49_10, w2_q44_1:w2_q44_10, w3_q43_1:w3_q43_10, w4_q47_1:w4_q47_10,
    w1_q50_1:w1_q50_10, w2_q45_1:w2_q45_10, w3_q44_1:w3_q44_10, w4_q48_1:w4_q48_10,
    w1_q51_1:w1_q51_6, w2_q46_1:w2_q46_6, w3_q45_1:w3_q45_6, w4_q49_1:w4_q49_6,
    #Voting behavior, intention, turnout
    w1_q33, w2_q34, w3_q6,
    w1_q34, w2_q35, w3_q7,
    w1_q35, w2_q36, w3_q8,
    #Campaign
    w3_q5_1:w3_q5_12, w4_q5_1:w4_q5_12,
    w3_q47, w4_q53)
```

## 2.2.1 All functions

As there are some variables that require the same processing steps it is easier to write two functions. The first takes the groups of the same variables that are currently in columns and pivots them so that if there are four waves there will be four rows for the same respondent (one for each wave). A new column is created to identify the waves. The second function recodes this column to only contain the number of the wave without any additional text.

```
#Make a function to pivot_longer the variables with waves
spread_waves <-
  function(var_tibble, new_var_name) {
    tibble <-
      pivot_longer(
        var_tibble,
        cols = starts_with("w"),
        names_to = "wave",
        values_to = new_var_name
      )
    return(tibble)
  }
```

```
#Make a function to recode waves.
recode_wave <- function(x) {
  case_when(
    grepl("w1", x) == TRUE ~ "1",
    grepl("w2", x) == TRUE ~ "2",
    grepl("w3", x) == TRUE ~ "3",
    grepl("w4", x) == TRUE ~ "4"
  )
}
```

```
#Make a function to recode campaign exposure
#Unique values of each variables checked
#Order confirm to survey documentary

recode_expo <- function(x) {
  recode_factor(x,
    "helemaal niet" = 1,
    "bijna nooit" = 2,
    "minder dan één keer per week" = 3,
    "ongeveer één keer per week" = 4,
    "verschillende keren per week" = 5,
    "bijna dagelijks" = 6,
    "dagelijks" = 7
  )
}
```

## 2.2.2 Interest

```
#Parse the variables to characters.
#To unite column types.

cols.char <-
  c("RESPNR", "w1_q3_1", "w1_q3_2", "w1_q3_3", "w1_q3_4", "w1_q3_5", "w1_q3_6")

time_EU[cols.char] <- sapply(time_EU[cols.char], as.character)
```

```
#Apply the function to all 6 interest variables:  
#Always check after you paste. e.g. waves, "interest"  
variables_1 <- time_EU %>% select(RESPNR, w1_q3_1, w2_q3_1, w3_q3_1, w4_q3_1)  
wave_int_1 <- spread_waves(variables_1, "intrest1") %>% mutate(wave = recode_wave(wave), intrest1 = parse_number(intrest  
1))  
  
variables_2 <- time_EU %>% select(RESPNR, w1_q3_2, w2_q3_2, w3_q3_2, w4_q3_2)  
wave_int_2 <- spread_waves(variables_2, "intrest2") %>% mutate(wave = recode_wave(wave), intrest2 = parse_number(intrest  
2))  
  
variables_3 <- time_EU %>% select(RESPNR, w1_q3_3, w2_q3_3, w3_q3_3, w4_q3_3)  
wave_int_3 <- spread_waves(variables_3, "intrest3") %>% mutate(wave = recode_wave(wave), intrest3 = parse_number(intrest  
3))  
  
variables_4 <- time_EU %>% select(RESPNR, w1_q3_4, w2_q3_4, w3_q3_4, w4_q3_4)  
wave_int_4 <- spread_waves(variables_4, "intrest4") %>% mutate(wave = recode_wave(wave), intrest4 = parse_number(intrest  
4))  
  
variables_5 <- time_EU %>% select(RESPNR, w1_q3_5, w2_q3_5, w3_q3_5, w4_q3_5)  
wave_int_5 <- spread_waves(variables_5, "intrest5") %>% mutate(wave = recode_wave(wave), intrest5 = parse_number(intrest  
5))  
  
variables_6 <- time_EU %>% select(RESPNR, w1_q3_6, w2_q3_6, w3_q3_6, w4_q3_6)  
wave_int_6 <- spread_waves(variables_6, "intrest6") %>% mutate(wave = recode_wave(wave), intrest6 = parse_number(intrest  
6))
```

```
#Join the 6 tibbles of each interest questions by wave and respondent number.  
wave_interest <- wave_int_1 %>%  
  left_join(wave_int_2, by = c("RESPNR", "wave")) %>%  
  left_join(wave_int_3, by = c("RESPNR", "wave")) %>%  
  left_join(wave_int_4, by = c("RESPNR", "wave")) %>%  
  left_join(wave_int_5, by = c("RESPNR", "wave")) %>%  
  left_join(wave_int_6, by = c("RESPNR", "wave"))  
  
#Cleaned and ready to join table is wave_interest
```

## 2.2.3 Government satisfaction

```
#Parse the variables to characters.  
#As there is only one general government satisfaction item there are no tibbles to join.  
  
cols.char_govsat <-  
  c("RESPNR", "w1_q6", "w2_q5", "w3_q9", "w4_q11")  
  
time_EU[cols.char_govsat] <- sapply(time_EU[cols.char_govsat], as.character)  
  
#apply both functions  
gov_sat <- time_EU %>% select(RESPNR, w1_q6, w2_q5, w3_q9, w4_q11)  
wave_gov_sat <- spread_waves(gov_sat, "gov_sat1") %>% mutate(wave = recode_wave(wave), gov_sat1 = parse_number(gov_sat1))
```

## 2.2.4 Government satisfaction specific

```
#Apply spread_waves, recode functions to clean it
gov_variables_1 <- time_EU %>% select(RESPNR,w1_q7_1,w2_q6_1, w3_q10_1, w4_q12_1)
wave_gov_specific_1 <- spread_waves(gov_variables_1, "gov_specific_1") %>% mutate(wave = recode_wave(wave), gov_specific_1 = parse_number(gov_specific_1))

gov_variables_2 <- time_EU %>% select(RESPNR, w1_q7_2,w2_q6_2, w3_q10_2, w4_q12_2)
wave_gov_specific_2 <- spread_waves(gov_variables_2, "gov_specific_2") %>% mutate(wave = recode_wave(wave),gov_specific_2 = parse_number(gov_specific_2))

gov_variables_3 <- time_EU %>% select(RESPNR, w1_q7_1,w2_q6_3, w3_q10_3, w4_q12_3)
wave_gov_specific_3<- spread_waves(gov_variables_3, "gov_specific_3") %>% mutate(wave = recode_wave(wave),gov_specific_3 = parse_number(gov_specific_3))

gov_variables_4 <- time_EU %>% select(RESPNR, w1_q7_1,w2_q6_4, w3_q10_4, w4_q12_4)
wave_gov_specific_4 <- spread_waves(gov_variables_4, "gov_specific_4") %>% mutate(wave = recode_wave(wave),gov_specific_4 = parse_number(gov_specific_4))
```

```
#Join each tibble together
wave_gov_sat_specific <- wave_gov_specific_1 %>%
  left_join(wave_gov_specific_2, by = c("RESPNR", "wave")) %>%
  left_join(wave_gov_specific_3, by = c("RESPNR", "wave")) %>%
  left_join(wave_gov_specific_4, by = c("RESPNR", "wave"))
```

## 2.2.5 Political cynicism

```
#Apply recode and spread waves function to clean
cynicism_variables_1 <- time_EU %>% select(RESPNR,w1_q10_1,w2_q7_1, w3_q11_1, w4_q13_1)
wave_cynicism_1 <- spread_waves(cynicism_variables_1, "pol_cynicism_1") %>% mutate(wave = recode_wave(wave),pol_cynicism_1 = parse_number(pol_cynicism_1))

cynicism_variables_2 <- time_EU %>% select(RESPNR,w1_q10_2,w2_q7_2, w3_q11_2, w4_q13_2)
wave_cynicism_2 <- spread_waves(cynicism_variables_2, "pol_cynicism_2") %>% mutate(wave = recode_wave(wave),pol_cynicism_2 = parse_number(pol_cynicism_2))

cynicism_variables_3 <- time_EU %>% select(RESPNR,w1_q10_3,w2_q7_3, w3_q11_3, w4_q13_3)
wave_cynicism_3 <- spread_waves(cynicism_variables_3, "pol_cynicism_3") %>% mutate(wave = recode_wave(wave),pol_cynicism_3 = parse_number(pol_cynicism_3))

cynicism_variables_4 <- time_EU %>% select(RESPNR,w1_q10_4,w2_q7_4, w3_q11_4, w4_q13_4)
wave_cynicism_4 <- spread_waves(cynicism_variables_4, "pol_cynicism_4") %>% mutate(wave = recode_wave(wave), pol_cynicism_4 = parse_number(pol_cynicism_4))
```

```
#Join all 4 small tibbles.
wave_pol_cynicism <- wave_cynicism_1 %>%
  left_join(wave_cynicism_2, by = c("RESPNR", "wave")) %>%
  left_join(wave_cynicism_3, by = c("RESPNR", "wave")) %>%
  left_join(wave_cynicism_4, by = c("RESPNR", "wave"))
```

## 2.2.6 Political knowledge

```
#Apply recode and spread waves function to clean for all items of political knowledge
pol_knowledge_1 <- time_EU %>% select(RESPNR,w1_q18,w2_q14, w3_q19, w4_q21)
wave_pol_knowledge_1 <- spread_waves(pol_knowledge_1, "pol_knowledge_1") %>% mutate(wave = recode_wave(wave))

wave_pol_knowledge_1 <-
  wave_pol_knowledge_1 %>% mutate(
    pol_knowledge_1 = recode(
      pol_knowledge_1,
      "Frans Timmermans" = 1,
      .default = 0
    )
  )
```

```
pol_knowledge_2 <- time_EU %>% select(RESPNR,w1_q19,w2_q15, w3_q20, w4_q22)
wave_pol_knowledge_2 <- spread_waves(pol_knowledge_2, "pol_knowledge_2") %>% mutate(wave = recode_wave(wave))

wave_pol_knowledge_2 <-
  wave_pol_knowledge_2 %>% mutate(
    pol_knowledge_2 = recode(
      pol_knowledge_2,
      "4 jaar" = 1,
      .default = 0
    )
  )
```

```
pol_knowledge_3 <- time_EU %>% select(RESPNR,w1_q20,w2_q16, w3_q21, w4_q23)
wave_pol_knowledge_3 <- spread_waves(pol_knowledge_3, "pol_knowledge_3") %>% mutate(wave = recode_wave(wave))

wave_pol_knowledge_3 <-
  wave_pol_knowledge_3 %>% mutate(
    pol_knowledge_3 = recode(
      pol_knowledge_3,
      "26" = 1,
      .default = 0
    )
  )
```

```
pol_knowledge_4 <- time_EU %>% select(RESPNR,w1_q21,w2_q17, w3_q22, w4_q24)
wave_pol_knowledge_4 <- spread_waves(pol_knowledge_4, "pol_knowledge_4") %>% mutate(wave = recode_wave(wave))

wave_pol_knowledge_4 <-
  wave_pol_knowledge_4 %>% mutate(
    pol_knowledge_4 = recode(
      pol_knowledge_4,
      "28" = 1,
      .default = 0
    )
  )
```

```
pol_knowledge_5 <- time_EU %>% select(RESPNR,w1_q22,w2_q18, w3_q23, w4_q25)
wave_pol_knowledge_5 <- spread_waves(pol_knowledge_5, "pol_knowledge_5") %>% mutate(wave = recode_wave(wave))

wave_pol_knowledge_5 <-
  wave_pol_knowledge_5 %>% mutate(
    pol_knowledge_5 = recode(
      pol_knowledge_5,
      "Martin Schulz" = 1,
      .default = 0
    )
  )
```

```
#Join all 5 small tibbles into one political knowledge tibble
wave_pol_knowledge <- wave_pol_knowledge_1 %>%
  left_join(wave_pol_knowledge_2, by = c("RESPNR", "wave")) %>%
  left_join(wave_pol_knowledge_3, by = c("RESPNR", "wave")) %>%
  left_join(wave_pol_knowledge_4, by = c("RESPNR", "wave")) %>%
  left_join(wave_pol_knowledge_5, by = c("RESPNR", "wave"))
```

## 2.2.7 EU attitude question 1

```
#Apply recode and spread waves function
#Parse to only numbers
EU_att_1 <- time_EU %>% select(RESPNR,w1_q26_1, w2_q22_1, w3_q28_1, w4_q31_1)
wave_EU_att_1 <- spread_waves(EU_att_1, "EU_attitude_1") %>% mutate(wave = recode_wave(wave), EU_attitude_1 = parse_number(EU_attitude_1))

EU_att_2 <- time_EU %>% select(RESPNR,w1_q26_2, w2_q22_2, w3_q28_2, w4_q31_2)
wave_EU_att_2 <- spread_waves(EU_att_2, "EU_attitude_2") %>% mutate(wave = recode_wave(wave), EU_attitude_2 = parse_number(EU_attitude_2))

EU_att_3 <- time_EU %>% select(RESPNR,w1_q26_3, w2_q22_3, w3_q28_3, w4_q31_3)
wave_EU_att_3 <- spread_waves(EU_att_3, "EU_attitude_3") %>% mutate(wave = recode_wave(wave), EU_attitude_3 = parse_number(EU_attitude_3))

EU_att_4 <- time_EU %>% select(RESPNR,w1_q26_4, w2_q22_4, w3_q28_4, w4_q31_4)
wave_EU_att_4 <- spread_waves(EU_att_4, "EU_attitude_4") %>% mutate(wave = recode_wave(wave), EU_attitude_4 = parse_number(EU_attitude_4))

EU_att_5 <- time_EU %>% select(RESPNR,w1_q26_5, w2_q22_5, w3_q28_5, w4_q31_5)
wave_EU_att_5 <- spread_waves(EU_att_5, "EU_attitude_5") %>% mutate(wave = recode_wave(wave), EU_attitude_5 = parse_number(EU_attitude_5))

EU_att_6 <- time_EU %>% select(RESPNR,w1_q26_6, w2_q22_6, w3_q28_6, w4_q31_6)
wave_EU_att_6 <- spread_waves(EU_att_6, "EU_attitude_6") %>% mutate(wave = recode_wave(wave), EU_attitude_6 = parse_number(EU_attitude_6))

EU_att_7 <- time_EU %>% select(RESPNR,w1_q26_7, w2_q22_7, w3_q28_7, w4_q31_7)
wave_EU_att_7 <- spread_waves(EU_att_7, "EU_attitude_7") %>% mutate(wave = recode_wave(wave), EU_attitude_7 = parse_number(EU_attitude_7))

EU_att_8 <- time_EU %>% select(RESPNR,w1_q26_8, w2_q22_8, w3_q28_8, w4_q31_8)
wave_EU_att_8 <- spread_waves(EU_att_8, "EU_attitude_8") %>% mutate(wave = recode_wave(wave), EU_attitude_8 = parse_number(EU_attitude_8))

EU_att_9 <- time_EU %>% select(RESPNR,w1_q26_9, w2_q22_9, w3_q28_9, w4_q31_9)
wave_EU_att_9 <- spread_waves(EU_att_9, "EU_attitude_9") %>% mutate(wave = recode_wave(wave), EU_attitude_9 = parse_number(EU_attitude_9))
```

```
#Put all tibbles together
wave_EU_attitudes_1 <- wave_EU_att_1 %>%
  left_join(wave_EU_att_2, by = c("RESPNR", "wave")) %>%
  left_join(wave_EU_att_3, by = c("RESPNR", "wave")) %>%
  left_join(wave_EU_att_4, by = c("RESPNR", "wave")) %>%
  left_join(wave_EU_att_5, by = c("RESPNR", "wave")) %>%
  left_join(wave_EU_att_6, by = c("RESPNR", "wave")) %>%
  left_join(wave_EU_att_7, by = c("RESPNR", "wave")) %>%
  left_join(wave_EU_att_8, by = c("RESPNR", "wave")) %>%
  left_join(wave_EU_att_9, by = c("RESPNR", "wave"))
```

## 2.2.8 EU Attitudes question 2

```
#Apply recode and spread waves function and parse to only numbers

EU_att2_1 <- time_EU %>% select(RESPNR, w1_q27_1, w2_q23_1, w3_q29_1, w4_q32_1)
wave_att2_1 <- spread_waves(EU_att2_1, "EU_attitudes2_1") %>% mutate(wave = recode_wave(wave), EU_attitudes2_1 = parse_number(EU_attitudes2_1))

EU_att2_2 <- time_EU %>% select(RESPNR, w1_q27_2, w2_q23_2, w3_q29_2, w4_q32_2)
wave_att2_2 <- spread_waves(EU_att2_2, "EU_attitudes2_2") %>% mutate(wave = recode_wave(wave), EU_attitudes2_2 = parse_number(EU_attitudes2_2))

EU_att2_3 <- time_EU %>% select(RESPNR, w1_q27_3, w2_q23_3, w3_q29_3, w4_q32_3)
wave_att2_3 <- spread_waves(EU_att2_3, "EU_attitudes2_3") %>% mutate(wave = recode_wave(wave), EU_attitudes2_3 = parse_number(EU_attitudes2_3))

EU_att2_4 <- time_EU %>% select(RESPNR, w1_q27_4, w2_q23_4, w3_q29_4, w4_q32_4)
wave_att2_4 <- spread_waves(EU_att2_4, "EU_attitudes2_4") %>% mutate(wave = recode_wave(wave), EU_attitudes2_4 = parse_number(EU_attitudes2_4))

EU_att2_5 <- time_EU %>% select(RESPNR, w1_q27_5, w2_q23_5, w3_q29_5, w4_q32_5)
wave_att2_5 <- spread_waves(EU_att2_5, "EU_attitudes2_5") %>% mutate(wave = recode_wave(wave), EU_attitudes2_5 = parse_number(EU_attitudes2_5))

EU_att2_6 <- time_EU %>% select(RESPNR, w1_q27_6, w2_q23_6, w3_q29_6, w4_q32_6)
wave_att2_6 <- spread_waves(EU_att2_6, "EU_attitudes2_6") %>% mutate(wave = recode_wave(wave), EU_attitudes2_6 = parse_number(EU_attitudes2_6))

EU_att2_7 <- time_EU %>% select(RESPNR, w1_q27_7, w2_q23_7, w3_q29_7, w4_q32_7)
wave_att2_7 <- spread_waves(EU_att2_7, "EU_attitudes2_7") %>% mutate(wave = recode_wave(wave), EU_attitudes2_7 = parse_number(EU_attitudes2_7))

EU_att2_8 <- time_EU %>% select(RESPNR, w1_q27_8, w2_q23_8, w3_q29_8, w4_q32_8)
wave_att2_8 <- spread_waves(EU_att2_8, "EU_attitudes2_8") %>% mutate(wave = recode_wave(wave), EU_attitudes2_8 = parse_number(EU_attitudes2_8))

EU_att2_9 <- time_EU %>% select(RESPNR, w1_q27_9, w2_q23_9, w3_q29_9, w4_q32_9)
wave_att2_9 <- spread_waves(EU_att2_9, "EU_attitudes2_9") %>% mutate(wave = recode_wave(wave), EU_attitudes2_9 = parse_number(EU_attitudes2_9))
```

```
# Join all 9 items to one EU attitude tibble
wave_EU_attitudes_2 <- wave_att2_1 %>%
  left_join(wave_att2_2, by = c("RESPNR", "wave")) %>%
  left_join(wave_att2_3, by = c("RESPNR", "wave")) %>%
  left_join(wave_att2_4, by = c("RESPNR", "wave")) %>%
  left_join(wave_att2_5, by = c("RESPNR", "wave")) %>%
  left_join(wave_att2_6, by = c("RESPNR", "wave")) %>%
  left_join(wave_att2_7, by = c("RESPNR", "wave")) %>%
  left_join(wave_att2_8, by = c("RESPNR", "wave")) %>%
  left_join(wave_att2_9, by = c("RESPNR", "wave"))
```

## 2.2.9 Interpersonal communication: Politics [frequency]

```
int_com_pol <- time_EU %>% select(RESPNR, w1_q44, w2_q41, w3_q40, w4_q44)
wave_com_pol <- spread_waves(int_com_pol, "comm_pol") %>% mutate(wave = recode_wave(wave), comm_pol = parse_number(comm_pol))
```

## 2.2.10 Interpersonal communication: EU Politics [frequency]

```
int_com_EU1 <- time_EU %>% select(RESPNR, w1_q45, w2_q42, w3_q41, w4_q45)
wave_com_EU1 <- spread_waves(int_com_EU1, "comm_EU1") %>% mutate(wave = recode_wave(wave), comm_EU1 = parse_number(comm_EU1))
```

## 2.2.11 Interpersonal communication: EU EU2itics [position]

```
int_com_EU2 <- time_EU %>% select(RESPNR, w1_q46, w2_q43, w3_q42, w4_q46)
wave_com_EU2 <- spread_waves(int_com_EU2, "comm_EU2") %>% mutate(wave = recode_wave(wave), comm_EU2 = parse_number(comm_EU2))
```

## 2.2.12 Self reported media exposure [TV]

*#Apply recode and spread waves function and parse to only numbers*

```
exptv_1 <- time_EU %>% select(RESPNR, w1_q49_1, w2_q44_1, w3_q43_1, w4_q47_1) %>% unique()
wave_exptv_1 <- spread_waves(exptv_1, "exptv_1") %>% mutate(wave = recode_wave(wave), exptv_1 = parse_number(exptv_1))

exptv_2 <- time_EU %>% select(RESPNR, w1_q49_2, w2_q44_2, w3_q43_2, w4_q47_2) %>% unique()
wave_exptv_2 <- spread_waves(exptv_2, "exptv_2") %>% mutate(wave = recode_wave(wave), exptv_2 = parse_number(exptv_2))

exptv_3 <- time_EU %>% select(RESPNR, w1_q49_3, w2_q44_3, w3_q43_3, w4_q47_3) %>% unique()
wave_exptv_3 <- spread_waves(exptv_3, "exptv_3") %>% mutate(wave = recode_wave(wave), exptv_3 = parse_number(exptv_3))

exptv_4 <- time_EU %>% select(RESPNR, w1_q49_4, w2_q44_4, w3_q43_4, w4_q47_4) %>% unique()
wave_exptv_4 <- spread_waves(exptv_4, "exptv_4") %>% mutate(wave = recode_wave(wave), exptv_4 = parse_number(exptv_4))

exptv_5 <- time_EU %>% select(RESPNR, w1_q49_5, w2_q44_5, w3_q43_5, w4_q47_5) %>% unique()
wave_exptv_5 <- spread_waves(exptv_5, "exptv_5") %>% mutate(wave = recode_wave(wave), exptv_5 = parse_number(exptv_5))

exptv_6 <- time_EU %>% select(RESPNR, w1_q49_6, w2_q44_6, w3_q43_6, w4_q47_6) %>% unique()
wave_exptv_6 <- spread_waves(exptv_6, "exptv_6") %>% mutate(wave = recode_wave(wave), exptv_6 = parse_number(exptv_6))

exptv_7 <- time_EU %>% select(RESPNR, w1_q49_7, w2_q44_7, w3_q43_7, w4_q47_7) %>% unique()
wave_exptv_7 <- spread_waves(exptv_7, "exptv_7") %>% mutate(wave = recode_wave(wave), exptv_7 = parse_number(exptv_7))

exptv_8 <- time_EU %>% select(RESPNR, w1_q49_8, w2_q44_8, w3_q43_8, w4_q47_8) %>% unique()
wave_exptv_8 <- spread_waves(exptv_8, "exptv_8") %>% mutate(wave = recode_wave(wave), exptv_8 = parse_number(exptv_8))

exptv_9 <- time_EU %>% select(RESPNR, w1_q49_9, w2_q44_9, w3_q43_9, w4_q47_9) %>% unique()
wave_exptv_9 <- spread_waves(exptv_9, "exptv_9") %>% mutate(wave = recode_wave(wave), exptv_9 = parse_number(exptv_9))

exptv_10 <- time_EU %>% select(RESPNR, w1_q49_10, w2_q44_10, w3_q43_10, w4_q47_10) %>% unique()
wave_exptv_10 <- spread_waves(exptv_10, "exptv_10") %>% mutate(wave = recode_wave(wave), exptv_10 = parse_number(exptv_10))
```

*#Join the 10 items to one self reported media exposure of TV*

```
wave_exp_tv <- wave_exptv_1 %>%
  left_join(wave_exptv_2, by = c("RESPNR", "wave")) %>%
  left_join(wave_exptv_3, by = c("RESPNR", "wave")) %>%
  left_join(wave_exptv_4, by = c("RESPNR", "wave")) %>%
  left_join(wave_exptv_5, by = c("RESPNR", "wave")) %>%
  left_join(wave_exptv_6, by = c("RESPNR", "wave")) %>%
  left_join(wave_exptv_7, by = c("RESPNR", "wave")) %>%
  left_join(wave_exptv_8, by = c("RESPNR", "wave")) %>%
  left_join(wave_exptv_9, by = c("RESPNR", "wave")) %>%
  left_join(wave_exptv_10, by = c("RESPNR", "wave"))
```

## 2.2.13 Self reported media exposure [newspaper]

```
#Apply recode and spread waves function and parse to only numbers

expnp_1 <- time_EU %>% select(RESPNR, w1_q50_1, w2_q45_1, w3_q44_1, w4_q48_1) %>% unique()
wave_exppn_1 <- spread_waves(expnp_1, "expnp_1") %>% mutate(wave = recode_wave(wave), expnp_1 = parse_number(expnp_1))

expnp_2 <- time_EU %>% select(RESPNR, w1_q50_2, w2_q45_2, w3_q44_2, w4_q48_2) %>% unique()
wave_exppn_2 <- spread_waves(expnp_2, "expnp_2") %>% mutate(wave = recode_wave(wave), expnp_2 = parse_number(expnp_2))

expnp_3 <- time_EU %>% select(RESPNR, w1_q50_3, w2_q45_3, w3_q44_3, w4_q48_3) %>% unique()
wave_exppn_3 <- spread_waves(expnp_3, "expnp_3") %>% mutate(wave = recode_wave(wave), expnp_3 = parse_number(expnp_3))

expnp_4 <- time_EU %>% select(RESPNR, w1_q50_4, w2_q45_4, w3_q44_4, w4_q48_4) %>% unique()
wave_exppn_4 <- spread_waves(expnp_4, "expnp_4") %>% mutate(wave = recode_wave(wave), expnp_4 = parse_number(expnp_4))

expnp_5 <- time_EU %>% select(RESPNR, w1_q50_5, w2_q45_5, w3_q44_5, w4_q48_5) %>% unique()
wave_exppn_5 <- spread_waves(expnp_5, "expnp_5") %>% mutate(wave = recode_wave(wave), expnp_5 = parse_number(expnp_5))

expnp_6 <- time_EU %>% select(RESPNR, w1_q50_6, w2_q45_6, w3_q44_6, w4_q48_6) %>% unique()
wave_exppn_6 <- spread_waves(expnp_6, "expnp_6") %>% mutate(wave = recode_wave(wave), expnp_6 = parse_number(expnp_6))

expnp_7 <- time_EU %>% select(RESPNR, w1_q50_7, w2_q45_7, w3_q44_7, w4_q48_7) %>% unique()
wave_exppn_7 <- spread_waves(expnp_7, "expnp_7") %>% mutate(wave = recode_wave(wave), expnp_7 = parse_number(expnp_7))

expnp_8 <- time_EU %>% select(RESPNR, w1_q50_8, w2_q45_8, w3_q44_8, w4_q48_8) %>% unique()
wave_exppn_8 <- spread_waves(expnp_8, "expnp_8") %>% mutate(wave = recode_wave(wave), expnp_8 = parse_number(expnp_8))

expnp_9 <- time_EU %>% select(RESPNR, w1_q50_9, w2_q45_9, w3_q44_9, w4_q48_9) %>% unique()
wave_exppn_9 <- spread_waves(expnp_9, "expnp_9") %>% mutate(wave = recode_wave(wave), expnp_9 = parse_number(expnp_9))

expnp_10 <- time_EU %>% select(RESPNR, w1_q50_10, w2_q45_10, w3_q44_10, w4_q48_10) %>% unique()
wave_exppn_10 <- spread_waves(expnp_10, "expnp_10") %>% mutate(wave = recode_wave(wave), expnp_10 = parse_number(expnp_10))
```

```
# Join 10 items to self reported media exposure for news paper tibble
wave_exp_np <- wave_exppn_1 %>%
  left_join(wave_exppn_2, by = c("RESPNR", "wave")) %>%
  left_join(wave_exppn_3, by = c("RESPNR", "wave")) %>%
  left_join(wave_exppn_4, by = c("RESPNR", "wave")) %>%
  left_join(wave_exppn_5, by = c("RESPNR", "wave")) %>%
  left_join(wave_exppn_6, by = c("RESPNR", "wave")) %>%
  left_join(wave_exppn_7, by = c("RESPNR", "wave")) %>%
  left_join(wave_exppn_8, by = c("RESPNR", "wave")) %>%
  left_join(wave_exppn_9, by = c("RESPNR", "wave")) %>%
  left_join(wave_exppn_10, by = c("RESPNR", "wave"))
```

## 2.2.14 Self reported media exposure [internet]

```
#Apply recode, spread waves function and parse to numbers

expnet1 <- time_EU %>% select(RESPNR, w1_q51_1, w2_q46_1, w3_q45_1, w4_q49_1)
wave_exppn_1 <- spread_waves(expnet1, "expnet_1") %>% mutate(wave = recode_wave(wave), expnet_1 = parse_number(expnet_1))

expnet2 <- time_EU %>% select(RESPNR, w1_q51_2, w2_q46_2, w3_q45_2, w4_q49_2)
wave_exppn_2 <- spread_waves(expnet2, "expnet_2") %>% mutate(wave = recode_wave(wave), expnet_2 = parse_number(expnet_2))

expnet3 <- time_EU %>% select(RESPNR, w1_q51_5, w2_q46_5, w3_q45_5, w4_q49_5)
wave_exppn_3 <- spread_waves(expnet3, "expnet_3") %>% mutate(wave = recode_wave(wave), expnet_3 = parse_number(expnet_3))

expnet4 <- time_EU %>% select(RESPNR, w1_q51_6, w2_q46_6, w3_q45_6, w4_q49_6)
wave_exppn_4 <- spread_waves(expnet4, "expnet_4") %>% mutate(wave = recode_wave(wave), expnet_4 = parse_number(expnet_4))
```

```
#Join the 4 tibbles to one self reported media exposure specifically for internet
wave_exp_net <- wave_expnets_1 %>%
  left_join(wave_expnets_2, by = c("RESPNR", "wave")) %>%
  left_join(wave_expnets_3, by = c("RESPNR", "wave")) %>%
  left_join(wave_expnets_4, by = c("RESPNR", "wave"))
```

## 2.2.15 Voting intention: turnout intention

```
#Select relevant variables for turnout intention:
turnout_intention <- time_EU %>% select(RESPNR, w1_q33, w2_q34, w3_q6)

#Use function recode_wave and spread_waves to clean the variables:
wave_turnout <-
  spread_waves(turnout_intention, "Turnout_intention") %>%
  mutate(
    wave = recode_wave(wave),
    Turnout_intention = parse_number(Turnout_intention)
  )

#Cleaned and ready to join table is wave_turnout
```

## 2.2.16 Voting intention

```
#Select relevant variables for voting intention:
voting_intention <- time_EU %>% select(RESPNR, w1_q34, w2_q35, w3_q7)

#Use function recode_wave and spread_waves to clean the variables:
wave_vote_int <-
  spread_waves(voting_intention, "Vote_int") %>%
  mutate(wave = recode_wave(wave))

#Cleaned and ready to join table is wave_vote_int
```

## 2.2.17 Voting intention: vote intention certainty

```
#Select relevant variables for voting intention certainty:
voting_certain <- time_EU %>% select(RESPNR, w1_q35, w2_q36, w3_q8)

#Use function recode_wave and spread_waves to clean the variables:
wave_vote_certain <-
  spread_waves(voting_certain, "Vote_int_certain") %>%
  mutate(wave = recode_wave(wave),
         Vote_int_certain = parse_number(Vote_int_certain))

#Cleaned and ready to join table is wave_vote_certain
```

## 2.2.18 Campaign Exposure

```
#Select relevant variables for campaign exposure in wave 3 and wave 4
#Use spread_waves function and recode_wave function to clean them
#Do the same for all 12 variables used to measure campaign exposure in wave 3 and 4.

variables_expo_1 <- time_EU %>% select(RESPNR, w3_q5_1, w4_q5_1)
wave_cam_1 <- spread_waves(variables_expo_1, "exposure1") %>% mutate(wave = recode_wave(wave))

variables_expo_2 <- time_EU %>% select(RESPNR, w3_q5_2, w4_q5_2)
wave_cam_2 <- spread_waves(variables_expo_2, "exposure2") %>% mutate(wave = recode_wave(wave))

variables_expo_3 <- time_EU %>% select(RESPNR, w3_q5_3, w4_q5_3)
wave_cam_3 <- spread_waves(variables_expo_3, "exposure3") %>% mutate(wave = recode_wave(wave))

variables_expo_4 <- time_EU %>% select(RESPNR, w3_q5_4, w4_q5_4)
wave_cam_4 <- spread_waves(variables_expo_4, "exposure4") %>% mutate(wave = recode_wave(wave))

variables_expo_5 <- time_EU %>% select(RESPNR, w3_q5_5, w4_q5_5)
wave_cam_5 <- spread_waves(variables_expo_5, "exposure5") %>% mutate(wave = recode_wave(wave))

variables_expo_6 <- time_EU %>% select(RESPNR, w3_q5_6, w4_q5_6)
wave_cam_6 <- spread_waves(variables_expo_6, "exposure6") %>% mutate(wave = recode_wave(wave))

variables_expo_7 <- time_EU %>% select(RESPNR, w3_q5_7, w4_q5_7)
wave_cam_7 <- spread_waves(variables_expo_7, "exposure7") %>% mutate(wave = recode_wave(wave))

variables_expo_8 <- time_EU %% select(RESPNR, w3_q5_8, w4_q5_8)
wave_cam_8 <- spread_waves(variables_expo_8, "exposure8") %>% mutate(wave = recode_wave(wave))

variables_expo_9 <- time_EU %>% select(RESPNR, w3_q5_9, w4_q5_9)
wave_cam_9 <- spread_waves(variables_expo_9, "exposure9") %>% mutate(wave = recode_wave(wave))

variables_expo_10 <- time_EU %>% select(RESPNR, w3_q5_10, w4_q5_10)
wave_cam_10 <- spread_waves(variables_expo_10, "exposure10") %>% mutate(wave = recode_wave(wave))

variables_expo_11 <- time_EU %>% select(RESPNR, w3_q5_11, w4_q5_11)
wave_cam_11 <- spread_waves(variables_expo_11, "exposure11") %>% mutate(wave = recode_wave(wave))

variables_expo_12 <- time_EU %>% select(RESPNR, w3_q5_12, w4_q5_12)
wave_cam_12 <- spread_waves(variables_expo_12, "exposure12") %>% mutate(wave = recode_wave(wave))
```

```
#Join all 12 tibbles by respondent number and wave
wave_campaign_exposure <- wave_cam_1 %>%
  left_join(wave_cam_2, by = c("RESPNR", "wave")) %>%
  left_join(wave_cam_3, by = c("RESPNR", "wave")) %>%
  left_join(wave_cam_4, by = c("RESPNR", "wave")) %>%
  left_join(wave_cam_5, by = c("RESPNR", "wave")) %>%
  left_join(wave_cam_6, by = c("RESPNR", "wave")) %>%
  left_join(wave_cam_7, by = c("RESPNR", "wave")) %>%
  left_join(wave_cam_8, by = c("RESPNR", "wave")) %>%
  left_join(wave_cam_9, by = c("RESPNR", "wave")) %>%
  left_join(wave_cam_10, by = c("RESPNR", "wave")) %>%
  left_join(wave_cam_11, by = c("RESPNR", "wave")) %>%
  left_join(wave_cam_12, by = c("RESPNR", "wave"))

#Cleaned and ready to join table is wave_campaign_exposure
```

```
#Apply the recode function (see function section above) to clean exposure items.
wave_campaign_exposure <-
  wave_campaign_exposure %>%
  mutate(
    exposure1 = recode_expo(exposure1),
    exposure2 = recode_expo(exposure2),
    exposure3 = recode_expo(exposure3),
    exposure4 = recode_expo(exposure4),
    exposure5 = recode_expo(exposure5),
    exposure6 = recode_expo(exposure6),
    exposure7 = recode_expo(exposure7),
    exposure8 = recode_expo(exposure8),
    exposure9 = recode_expo(exposure9),
    exposure10 = recode_expo(exposure10),
    exposure11 = recode_expo(exposure11),
    exposure12 = recode_expo(exposure12),
  )
```

## 2.2.19 Campaign attention

```
#Select relevant variables for campaign attention:
campaign_att <- time_EU %>% select(RESPNR, w3_q47, w4_q53)

#Use function recode_wave and spread_waves to clean the variables:
wave_campaign_att <-
  spread_waves(campaign_att, "Campaign_attention") %>%
  mutate(wave = recode_wave(wave),
         Campaign_attention = parse_number(Campaign_attention))

#Cleaned and ready to join table is wave_campaign_att
```

```
#Put the cleaned tibble back to the main table.
time_EU_cleaned <- wave_interest %>%
  left_join(wave_turnout, by = c("RESPNR", "wave")) %>%
  left_join(wave_vote_int, by = c("RESPNR", "wave")) %>%
  left_join(wave_vote_certain, by = c("RESPNR", "wave")) %>%
  left_join(wave_campaign_exposure, by = c("RESPNR", "wave")) %>%
  left_join(wave_campaign_att, by = c("RESPNR", "wave")) %>%
  left_join(wave_gov_sat, by = c("RESPNR", "wave")) %>%
  left_join(wave_gov_sat_specific, by = c("RESPNR", "wave")) %>%
  left_join(wave_pol_cynicism, by = c("RESPNR", "wave")) %>%
  left_join(wave_pol_knowledge, by = c("RESPNR", "wave")) %>%
  left_join(wave_EU_attitudes_1, by = c("RESPNR", "wave")) %>%
  left_join(wave_EU_attitudes_2, by = c("RESPNR", "wave")) %>%
  left_join(wave_com_pol, by = c("RESPNR", "wave")) %>%
  left_join(wave_com_EU1, by = c("RESPNR", "wave")) %>%
  left_join(wave_com_EU2, by = c("RESPNR", "wave")) %>%
  left_join(wave_exp_tv, by = c("RESPNR", "wave")) %>%
  left_join(wave_exp_np, by = c("RESPNR", "wave")) %>%
  left_join(wave_exp_net, by = c("RESPNR", "wave"))
```

```
write.csv(time_EU_cleaned, "time_EU_cleaned.csv")
```

**time\_EU\_cleaned** is cleaned

## 2.3 Content Analysis

### 2.3.1 Select key

```
#keep only the variables we want.
content <- content %>% select(V2, V3a:V4, V11:V13)
```

```
#check if news story number is unique and can be used as key to link tibbles.
sum(unique(content$V2)==FALSE)
```

```
## [1] 0
```

## 2.3.2 Date

```
#year is in 2-digit, change to 4 digit. check value of years.  
unique(content$V3c)
```

```
## [1] 14 13
```

```
content <- content %>%  
  mutate(V3c = recode(V3c, "14" = 2014, "13" = 2013), date = make_date(day = V3a, month = V3b, year = V3c)) %>% select(-c(V3a:V3c))  
#remove the old 3 variables of day, month and year.  
  
#check first 5 dates to see how it is.  
head(content$date, 5)
```

```
## [1] "2014-05-06" "2014-05-06" "2014-05-06" "2014-05-06" "2014-05-06"
```

## 2.3.3 News outlet

```
#check V4 unit to see what is wrong  
unique(content$V4)
```

```
## [1] "21"           "22"           "23"           "NRC Handelsblad"  
## [5] "de Volkskrant" "de Telegraaf"   "NOS Journaal"   "RTL Nieuws"  
## [9] "nu.nl"
```

*#nu.nl is not in the outlet list. I code it as 116.*

```
content <- content %>%  
  mutate(V4 = recode(V4, "21" = 21, "22" = 22, "23" = 23, "NRC Handelsblad" = 111, "de Volkskrant" = 112, "de Telegraaf" = 113, "NOS Journaal" = 114, "RTL Nieuws" = 115, "nu.nl" = 116))  
  
#check recoded results.  
unique(content$V4)
```

```
## [1] 21 22 23 111 112 113 114 115 116
```

## 2.3.4 News evaluation

```
#check values of V11  
unique(content$V11)
```

```
## [1] NA           "Balanced/mixed"  
## [3] "Mentioned but not evaluated" "Rather positive"  
## [5] "Not applicable / not mentioned" "Rather negative"  
## [7] "Negative"      "Positive"
```

```
#re-code V11 to numeric.
content <-
  content %>% mutate(
    V11 = recode(
      V11,
      "Negative" = 1,
      "Rather negative" = 2,
      "Balanced/mixed" = 3,
      "Rather positive" = 4,
      "Positive" = 5,
      "Mentioned but not evaluated" = 0,
      "Not applicable / not mentioned" = 9
    )
  )

#check recoded results.
unique(content$V11)
```

```
## [1] NA 3 0 4 9 2 1 5
```

```
#check values of V11
unique(content$V12)
```

```
## [1] NA                      "Not applicable / not mentioned"
## [3] "Mentioned but not evaluated" "Rather negative"
## [5] "Positive"                 "Negative"
## [7] "Rather positive"          "Balanced/mixed"
```

```
content <-
  content %>% mutate(
    V12 = recode(
      V12,
      "Negative" = 1,
      "Rather negative" = 2,
      "Balanced/mixed" = 3,
      "Rather positive" = 4,
      "Positive" = 5,
      "Mentioned but not evaluated" = 0,
      "Not applicable / not mentioned" = 9
    )
  )

#check recoded results.
unique(content$V12)
```

```
## [1] NA 9 0 2 5 1 4 3
```

```
#check values of V13
unique(content$V13)
```

```
## [1] NA                      "Not applicable / not mentioned"
## [3] "Mentioned but not evaluated" "Negative"
## [5] "Balanced/mixed"             "Rather positive"
## [7] "Positive"                  "Rather negative"
```

```

content <-
  content %>% mutate(
    V13 = recode(
      V13,
      "Negative" = 1,
      "Rather negative" = 2,
      "Balanced/mixed" = 3,
      "Rather positive" = 4,
      "Positive" = 5,
      "Mentioned but not evaluated" = 0,
      "Not applicable / not mentioned" = 9
    )
  )
#check recoded results.
unique(content$V13)

```

```
## [1] NA 9 0 1 3 4 5 2
```

```
write_csv(content, "content.csv")
```

**content** is cleaned

```

#Clean the environment
rm(campaign_att, cynicism_variables_1, cynicism_variables_2, cynicism_variables_3, cynicism_variables_4, EU, EU_att_1, EU_att_2, EU_att_3, EU_att_4, EU_att_5, EU_att_6, EU_att_7, EU_att_8, EU_att_9, EU_att2_1, EU_att2_2, EU_att2_3, EU_att2_4, EU_att2_5, EU_att2_6, EU_att2_7, EU_att2_8, EU_att2_9, expnet1, expnet2, expnet3, expnet4, expnp_1, expnp_2, expnp_3, expnp_4, expnp_5, expnp_6, expnp_7, expnp_8, expnp_9, expnp_10, exptv_1, exptv_2, exptv_3, exptv_4, exptv_5, exptv_6, exptv_7, exptv_8, exptv_9, exptv_10, gov_sat, gov_variables_1, gov_variables_2, gov_variables_3, gov_variables_4, int_com_EU1, int_com_EU2, int_com_pol, pol_knowledge_1, pol_knowledge_2, pol_knowledge_3, pol_knowledge_4, pol_knowledge_5, time_EU, turno_ut_intention, variables_1, variables_2, variables_3, variables_4, variables_5, variables_6, variables_expo_1, variables_expo_2, variables_expo_3, variables_expo_4, variables_expo_5, variables_expo_6, variables_expo_7, variables_expo_8, variables_expo_9, variables_expo_10, variables_expo_11, variables_expo_12, voting_certain, voting_intention, wave_att2_1, wave_att2_2, wave_att2_3, wave_att2_4, wave_att2_5, wave_att2_6, wave_att2_7, wave_att2_8, wave_att2_9, wave_cam_1, wave_cam_2, wave_cam_3, wave_cam_4, wave_cam_5, wave_cam_6, wave_cam_7, wave_cam_8, wave_cam_9, wave_cam_10, wave_cam_11, wave_cam_12, wave_campaign_att, wave_campaign_exposure, wave_com_EU1, wave_com_EU2, wave_com_pol, wave_cynicism_1, wave_cynicism_2, wave_cynicism_3, wave_cynicism_4, wave_EU_att_1, wave_EU_att_2, wave_EU_att_3, wave_EU_att_4, wave_EU_att_5, wave_EU_att_6, wave_EU_att_7, wave_EU_att_8, wave_EU_att_9, wave_EU_attitudes_1, wave_EU_attitudes_2, wave_exp_net, wave_exp_np, wave_exp_tv, wave_expn_1, wave_expn_2, wave_expn_3, wave_expn_4, wave_expn_5, wave_expn_6, wave_expn_7, wave_expn_8, wave_expn_9, wave_expn_10, wave_exptv_1, wave_exptv_2, wave_exptv_3, wave_exptv_4, wave_exptv_5, wave_exptv_6, wave_exptv_7, wave_exptv_8, wave_exptv_9, wave_exptv_10, wave_gov_sat, wave_gov_specific, wave_gov_specific_1, wave_gov_specific_2, wave_gov_specific_3, wave_gov_specific_4, wave_int_1, wave_int_2, wave_int_3, wave_int_4, wave_int_5, wave_int_6, wave_interest, wave_pol_cynicism, wave_pol_knowledge, wave_pol_knowledg_e_1, wave_pol_knowledge_2, wave_pol_knowledge_3, wave_pol_knowledge_4, wave_pol_knowledge_5, wave_turnout, wave_vote_certain, wave_vote_int, cols.char_govsat, namekey, recode_expo, recode_wave, spread_waves)

```

## 3 Visualisations

3 CSV files remain (demo\_EU, time\_EU\_cleaned and content) which each of the group members will draw from to create the individual visualisations. The first demo\_EU containing all the demographic variables in the survey that were only captured once, time\_EU\_cleaned contains the cleaned variables from the survey that the respondents had to answer several times throughout the duration of the 2014 European Election Campaign Study. The third file containing all the relevant variables from the content analysis data set.

### 3.1 Lilian's Visualizaton

#### 3.1.1 Read data file

```

content <- read_csv("content.csv")
survey_data <- read_csv("time_EU_cleaned.csv")

```

#### 3.1.2 Create Reverse Coding Function

```
recode_reverse <- function(x) {
  recode(x,
    "7" = 1,
    "6" = 2,
    "5" = 3,
    "4" = 4,
    "3" = 5,
    "2" = 6,
    "1" = 7
  )
}
```

### 3.1.3 Reverse code the items in EU attitude and political cynicism.

- Some of the items are asked reversely in the survey.

```
survey_data <-
  survey_data %>%
  mutate(
    EU_attitudes2_5 = recode_reverse(EU_attitudes2_5),
    EU_attitudes2_6 = recode_reverse(EU_attitudes2_6),
    EU_attitudes2_7 = recode_reverse(EU_attitudes2_7),
    EU_attitudes2_8 = recode_reverse(EU_attitudes2_8),
    EU_attitudes2_9 = recode_reverse(EU_attitudes2_9),
    pol_cynicism_3 = recode_reverse(pol_cynicism_3),
    pol_cynicism_4 = recode_reverse(pol_cynicism_4)
  )
```

### 3.1.4 Calculate mean score of political cynicism, political knowledge, and EU attitude for each participants.

```
survey_data <-
  survey_data %>%
  mutate(general_EU_att = rowMeans(select(., 40:57)),
         pol_cy = rowMeans(select(., 31:34)),
         pol_kn = rowMeans(select(., 35:39))
  )

#DV: sum score of EU attitudes1&2, 4 waves.
```

### 3.1.5 Categorize political knowledge

- Political knowledge is measure by 5 items. I calculated the mean score of 5 items.
- The mean ranges from 1 to 0. Therefore I use 0.5 as a threshold for high and low political knowledge.

```
survey_data <-
  survey_data %>%
  mutate(
    pol_kn_cat = case_when(
      pol_kn <= 0.5 ~ "Low political knowledge",
      pol_kn > 0.5 ~ "High political knowledge"
    )
  )
```

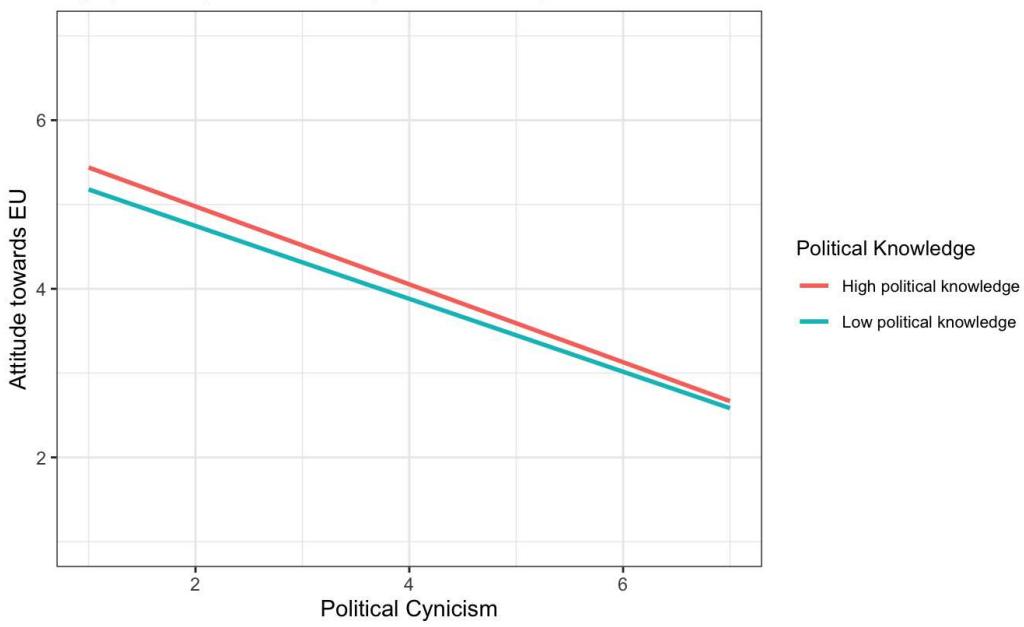
### 3.1.6 Create Plot

- As suggested in class, the over-time difference is very small and can be due to non significant relationships.
- So I decide to not facet on waves, but just show a general relationship. And I will run a regression model to test significance.
- Also, the problem with zoomed-in axis is fixed.

```
survey_data %>%
  ggplot(mapping = aes(x= pol_cy, y = general_EU_att, color = pol_kn_cat)) +
  geom_smooth(method = lm, se = FALSE) +
  labs(
    x = "Political Cynicism",
    y = "Attitude towards EU",
    title = "How Political Cynicism and Political Knowledge Influence Attitudes Towards EU.",
    subtitle = "High political cynicism and low political knowledge indicate low EU attitude.",
    caption = "Data from 2014 European Election Campaign Study (ASCoR, Claes de Vreese) \n Author: Lilian",
    color = "Political Knowledge"
  ) +
  ylim(1, 7) +
  theme_bw(base_size = 11) +
  theme(legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        plot.caption = element_text(face= "italic", size = 8),
        plot.subtitle = element_text(size = 11)
  )
```

### How Political Cynicism and Political Knowledge Influence Attitudes Towards EU.

High political cynicism and low political knowledge indicate low EU attitude.



#### 3.1.7 Run a regression model

- Results show significant effect of political cynicism and political knowledge.
- Political cynicism negatively affects EU attitudes. Higher the cynicism, lower the EU attitudes. With one unit increase of political cynicism, EU attitudes go down 0.46 unit.
- EU attitudes of people with low political knowledge is 0.29 unit lower than those with high political knowledge.
- The interaction effect between political cynicism and political knowledge is not significant.

```

model_re <- lm(general_EU_att ~ pol_cy * pol_kn_cat, data = survey_data)
# Dependent variable: EU attitudes;
# Predictors: political cynicism and political knowledge.

results_re <- apa_print(model_re)
# Get summary of results.

results_re$table$predictor <-
  recode(results_re$table$predictor,
    "Pol cy" = "Political cynicism",
    "Pol kn catlow political knowledge" = "Political knowledge (low)",
    "Pol cy $\times$ Pol kn catLow political knowledge" = "Political cynicism by political knowledge"
  )
# Recode variable names.

apa_table(results_re$table,
  caption = "How Political Cynicism And Political Knowledge Influences Attitudes Towards EU.",
  note = "Data from 2014 European Election Campaign Study (ASCoR, Claes de Vreese)",
  font_size = "small",
  escape = TRUE)

```

(#tab:Regression)

*How Political Cynicism And Political Knowledge Influences Attitudes Towards EU.*

predictor	b	95% CI	t(6920)	p
Intercept	5.90	[5.77, 6.03]	86.76	< .001
Political cynicism	-0.46	[-0.49, -0.43]	-32.38	< .001
Political knowledge (low)	-0.29	[-0.45, -0.13]	-3.45	.001
Political cynicism by political knowledge	0.03	[0.00, 0.06]	1.69	.090

*Note.* Data from 2014 European Election Campaign Study (ASCoR, Claes de Vreese)

## 3.2 Astrid's Visualizaton

In this project, I want to study the possible relationship between how many pieces of news mention EU on media, people's attitude towards EU, and their turnout intention, as well as their final turnout in EU election. My research question is:

*How does the change of the percentage of (positive) news mentioning EU influence people's turnout (intention) in each wave, and how does people's attitude towards EU influence the relationship?*

This research question includes four variables, which are listed below:

1. wave , in **time\_EU\_cleaned**.
2. percentage of (positive) news mentioning EU , in **content**.
3. turnout (intention) percentage , in **time\_EU\_cleaned** and **demo\_EU**.
4. mean EU attitude , in **time\_EU\_cleaned**.

My plan is that, after the process of data wrangling, I put all these four (five) variables in a plot to see if they have the same pattern. I assume that:

- the change of the percentage of (positive) news mentioning EU would influence people's attitude, since people's opinion would be influenced by what they read on media. Therefore, the pattern of percentage of (positive) news mentioning EU and mean EU attitude should be similar.
- people's attitude towards EU would impact their turnout rate negatively, because if one holds negative attitude towards an institution, this attitude may turn to a behavior (voting here) to change the institution, while people holding positive attitude may feel satisfied and feel that it is okay to vote or not. Thus, here I expect a contrary pattern of mean EU attitude and turnout (intention) percentage on the final plot.

I will explain the process of data wrangling for each variable in the following sections.

### 3.2.1 Wave

Wave is an indication of time change in the survey. The four waves of the survey represent 13-26/Dec/2013, 20-30/Mar/2014, 17-28/Apr/2014, and 26/May-02/Jun/2014, respectively.

### 3.2.2 Percentage of (positive) news mentioning EU

Percentage of (positive) news mentioning EU is a series of data calculated through data wrangling. This variable is from the content analysis tibble. As the content analysis tibble only shows the date of the news, while I want to study the change of percentage in four waves, I set a criterion to divide the news into four waves.

I want to explore how the news influence people's attitude and intention, so I want to divide the news according to people's memory capacity — when each wave of survey started, participants should still had memory about the news. According to the forgetting curve (Murre and Dros, 2015), people lose 90% of memory of new information within a week. Therefore, I set the start of each division of the news one week earlier than each wave of survey took place, and the end of each division is the last day of each survey.

The content analysis covers the news between 2013/Dec/02 and 2014/May/22, so the start and the end date of the first and the content analysis division follows these two dates.

The detailed dates can be found in the following tibble:

```
tribble(
  ~Wave, ~Survey_taking_period, ~Content_analysis_division,
  #----, -----, -----
  1, "2013/Dec/13-Dec/26", "2013/Dec/06-Dec/26",
  2, "2014/Mar/20-Mar/30", "2014/Mar/13-Mar/30",
  3, "2014/Apr/17-Apr/28", "2014/Apr/10-Apr/28",
  4, "2014/May/26-Jun/02", "2014/May/19-May/22"
)
```

Wave	Survey_taking_period	Content_analysis_division
	<dbl> <chr>	<chr>
1	2013/Dec/13-Dec/26	2013/Dec/06-Dec/26
2	2014/Mar/20-Mar/30	2014/Mar/13-Mar/30
3	2014/Apr/17-Apr/28	2014/Apr/10-Apr/28
4	2014/May/26-Jun/02	2014/May/19-May/22

4 rows

After dividing four waves, I calculate the percentage of news that mention EU among all news in a certain wave, and also the percentage of news holding positive view towards EU among all news that mention EU.

Because the survey data only includes participants from the Netherlands, here I only select news from NL media.

```
#Select related variables.
content <- content %>%
  select(V2, V11, date, V4) %>%
  filter(V4 %in% c(111, 112, 113, 114, 115, 116)) %>% #Select news from NL media.
  select(-V4) %>%
  unique()

#Mutate four waves according to date.
content <- content %>%
  mutate("1" = date >= as.Date("2013-12-06") & date <= as.Date("2013-12-26")) %>%
  mutate("2" = date >= as.Date("2014-03-13") & date <= as.Date("2014-03-30")) %>%
  mutate("3" = date >= as.Date("2014-04-10") & date <= as.Date("2014-04-28")) %>%
  mutate("4" = date >= as.Date("2014-05-19") & date <= as.Date("2014-05-22"))

#Only select news that is in four waves.
content <- filter(content, `1` == TRUE | `2` == TRUE | `3` == TRUE | `4` == TRUE)
```

```
#Create "wave" variable.
content <- content %>%
  pivot_longer(c(`1`, `2`, `3`, `4`), names_to = "wave", values_to = "news") %>%
  filter(news == TRUE) %>%
  select(-(news))
```

```
#Count the frequency of news holding different opinion in each wave.
fre <- content %>%
  group_by(wave, V11) %>%
  summarize(count = n())
```

According to my criteria of dividing waves:

- there are 330 passages belong to wave 1. 13 % of which mention EU, and 11 % contains rather positive or positive evaluation.
- There are 364 passages belong to wave 2. 15 % of which mention EU, and 5 % contains rather positive or positive evaluation.
- There are 727 passages belong to wave 3. 12 % of which mention EU, and 6 % contains rather positive or positive evaluation.
- There are 169 passages belong to wave 4. 34 % of which mention EU, and 16 % contains rather positive or positive evaluation.

I create a tibble for the data.

```
#Create tibble for frequency of news mentioning EU and news holding positive attitude towards EU in each wave.
news_fre <- tribble(
~wave, ~mention_EU, ~positive_EU,
#----, -----
1, round((fre$count[1] + fre$count[2] + fre$count[3]+ fre$count[4] + fre$count[5] + fre$count[6])/(fre$count[1] + fre$count[2] + fre$count[3]+ fre$count[4] + fre$count[5] + fre$count[6] + fre$count[7] + fre$count[8])*100), round((fre$count[5] + fre$count[6])/(fre$count[1] + fre$count[2] + fre$count[3]+ fre$count[4] + fre$count[5] + fre$count[6])*100),
2, round((fre$count[9] + fre$count[10] + fre$count[11]+ fre$count[12] + fre$count[13])/ (fre$count[9] + fre$count[10] + fre$count[11]+ fre$count[12] + fre$count[13] + fre$count[14] + fre$count[15])*100), round((fre$count[13])/ (fre$count[9] + fre$count[10] + fre$count[11]+ fre$count[12] + fre$count[13])*100),
3, round((fre$count[16] + fre$count[17] + fre$count[18] + fre$count[19]+ fre$count[20] + fre$count[21])/ (fre$count[16] + fre$count[17] + fre$count[18] + fre$count[19]+ fre$count[20] + fre$count[21] + fre$count[22] + fre$count[23])*100), round((fre$count[20] + fre$count[21])/ (fre$count[16] + fre$count[17] + fre$count[18] + fre$count[19]+ fre$count[20] + fre$count[21])*100),
4, round((fre$count[24] + fre$count[25] + fre$count[26]+ fre$count[27] + fre$count[28] + fre$count[29])/ (fre$count[24] + fre$count[25] + fre$count[26]+ fre$count[27] + fre$count[28] + fre$count[29] + fre$count[30] + fre$count[31])*100), round((fre$count[28] + fre$count[29])/ (fre$count[24] + fre$count[25] + fre$count[26]+ fre$count[27] + fre$count[28] + fre$count[29])*100)
)

rm(fre)
```

From table `news_fre` I find that the percentage of positive news in each wave is sometimes too small to analyze. Therefore, in this visualization, I only use the percentage of news mentioning EU.

### 3.2.3 Percentage turnout (intention)

In the first three waves of survey, people's turnout intention in EU election was measured. In this visualization, I also add the real turnout measured in wave 4, to see whether there is any difference. In order to compare this variable in four waves, which were measured in two different set of questions, I calculate the percentage of participants' who (intended to) vote in each wave:

- In wave 4, the calculation is simple and straight: I calculate the percentage of those who said they voted in the election (`turnout_EU == 4`) after removing the missing values.
- In wave 1-3, after removing the missing values, I calculate the percentage of people who selected 5-7 (thus more willing to vote) in each wave.

#### 3.2.3.1 Turnout (wave 4)

```
#Select related variables.
demo_EU <- demo_EU %>%
  select(RESPNR, turnout_EU)
```

```
#Count how many people chose "4" (voted).
turnout4 <- demo_EU %>%
  group_by(turnout_EU) %>%
  summarize(count = n()) %>%
  filter(turnout_EU != "NA")
```

#### 3.2.3.2 Turnout intention (wave 1-3)

```
time_EU_cleaned <- time_EU_cleaned %>%
  select(RESPNR, wave, Turnout_intention, EU_attitude_1:EU_attitudes2_9)

#Count how many people chose "5"- "7" (more willing to vote).
turnout1_3 <- time_EU_cleaned %>%
  select(RESPNR, wave, Turnout_intention) %>%
  group_by(wave, Turnout_intention) %>%
  filter(Turnout_intention != "NA") %>%
  summarise(count = n())
```

```
#Create a tibble gather the data.
turnout <- tribble(
  ~wave, ~turnout_percentage,
  #----, -----
  1, round((turnout1_3$count[5] + turnout1_3$count[6] + turnout1_3$count[7]) / (turnout1_3$count[1] + turnout1_3$count[2]
  + turnout1_3$count[3] + turnout1_3$count[4] + turnout1_3$count[5] + turnout1_3$count[6] + turnout1_3$count[7]) * 100),
  2, round((turnout1_3$count[12] + turnout1_3$count[13] + turnout1_3$count[14]) / (turnout1_3$count[8] + turnout1_3$count[
  9] + turnout1_3$count[10] + turnout1_3$count[11] + turnout1_3$count[12] + turnout1_3$count[13] + turnout1_3$count[14]) * 1
  00),
  3, round((turnout1_3$count[19] + turnout1_3$count[20] + turnout1_3$count[21]) / (turnout1_3$count[15] + turnout1_3$count[
  16] + turnout1_3$count[17] + turnout1_3$count[18] + turnout1_3$count[19] + turnout1_3$count[20] + turnout1_3$count[21]) *
  100),
  4, round(turnout4$count[4] / sum(turnout4$count) *100)
)
rm(turnout1_3, turnout4)
```

### 3.2.4 EU attitude

In the survey, EU attitude was measured by 18 items in two blocks. I reverse-code four items, and calculate the mean attitude in each survey. To better study the moderation relationship, I divide EU attitude into three groups: positive (attitude), medium, and negative.

```
#Reverse-code four items according to their contents.
time_EU_cleaned <- time_EU_cleaned %>%
  mutate_at(c("EU_attitudes2_5","EU_attitudes2_6", "EU_attitudes2_7", "EU_attitudes2_8"),
  funs(recode(., `1` = 7, `2` = 6, `3` = 5, `4` = 4, `5` = 3, `6` = 2, `7` = 1, .default = NaN))) %>%
#Calculate mean attitude for each participant in each wave.
mutate(mean_att = rowMeans(select(., 4:21)))
```

I divide participants' mean attitude (ranges from 1 to 7) into three groups:

- negative: 1-3;
- medium: 3-5;
- positive: 5-7.

```
#Create three Levels.
time_EU_cleaned <- time_EU_cleaned %>%
  mutate("Positive" = mean_att >= 5) %>%
  mutate("Medium" = mean_att >= 3 & mean_att < 5) %>%
  mutate("Negative" = mean_att < 3) %>%
#Create "wave" variable.
pivot_longer(c(`Positive`, `Medium`, `Negative`), names_to = "attitude_level", values_to = "value") %>%
  filter(value == TRUE) %>%
  select(-value) %>%
  unique()

#Calculate the mean attitude for each Level in each wave.
attitude <- time_EU_cleaned %>%
  group_by(wave, attitude_level) %>%
  summarize(mean_att_level = mean(mean_att))

#Pivot the tibble so it can join with other tibbles.
attitude <- attitude %>%
  mutate(wave = parse_double(wave)) %>% #Change the type of a variable so can it be used as a key to join with other tibbles.
  pivot_wider(names_from = attitude_level, values_from = mean_att_level)
```

### 3.2.5 Join table and draw plot

```
sum <- news_fre %>%
  select(-positive_EU) %>%
  left_join(attitude, by = "wave") %>%
  left_join(turnout, by = "wave")

#rm(news_fre, attitude, turnout)
```

In sum, I find that different groups of variables have different scales, and thus it would be hard for them to be presented on the same plot. I decide to rescale the variables, so they get the similar values, and when I create the plot, I rescale the y-axis, so the right value are presented. I rescale the following variables:

- The values for the attitude group are too small comparing with turnout\_percentage , so I multiply them by 11.
- The values for mention\_EU are also small, thus I multiply it by 2.

```
#Recalculate the variables, so they can be better presented on the plot.
```

```
sum <- sum %>%
  mutate(medium = (Medium * 11)) %>%
  mutate(negative = (Negative * 11)) %>%
  mutate(positive = (Positive * 11)) %>%
  mutate(news = (mention_EU *2))
```

```
# Set aesthetic.
```

```
names <- c("", "Wave 1", "Wave 2", "Wave 3", "Wave 4", "")
ycolor <- c("Turnout intention" = "#BA3F51", "Percentage of news" = "#B1C5DE", "Mean attitude" = "#353B6F")
yfill <- c("Percentage of news" = "#B1C5DE")
ylinetype <- c("Negative attitude" = "twodash", "Medium attitude" = "dashed", "Positive attitude" = "solid")
y1title <- element_text(color = "#BA3F51")
y2title <- element_text(color = "#B1C5DE")
```

```
# Draw plot
```

```
sum %>%
  ggplot() +
```

```
#Main part of the plot
```

```
geom_col(mapping = aes(x = wave, y = news, fill = "Percentage of news", color = "Percentage of news"), show.legend = FALSE,
  alpha = 0.3) + #News percentage
  geom_line(mapping = aes(x = wave, y = negative, color = "Mean attitude", linetype = "Negative attitude"), size = 1.25) +
  #Negative attitude
  geom_line(mapping = aes(x = wave, y = medium, color = "Mean attitude", linetype = "Medium attitude"), size = 1.25) + #Medium attitude
  geom_line(mapping = aes(x = wave, y = positive, color = "Mean attitude", linetype = "Positive attitude"), size = 1.25) +
  #Positive attitude
  geom_line(mapping = aes(x = wave, y = turnout_percentage, color = "Turnout intention"), size = 2) +
```

```
#Set axis name.
```

```
scale_x_continuous(labels = names, name = "") +
```

```
#Set the scale and name on the second y-axis for vote intention, because it is zoomed to better fit the plot.
```

```
scale_y_continuous("Percentage of people who (willing to) vote (%)", sec.axis = sec_axis(~ . /2, name = "Percentage of news mentioning EU (%))") +
```

```
#Zoom the plot.
```

```
coord_cartesian(ylim = c(20, 70)) +
```

```
#Set the labels and titles.
```

```
labs(color = "Category", linetype = "Attitude", title = "How does the percentage of news mentioning EU influence people's turnout (intention)?", subtitle = "And how does people's attitude towards EU influence the relationship?", caption = "Data source: European Election Campaign Study, Center for Politics and Communication,\nAmsterdam School of Communication Research ASCoR, University of Amsterdam, September 16, 2014") +
```

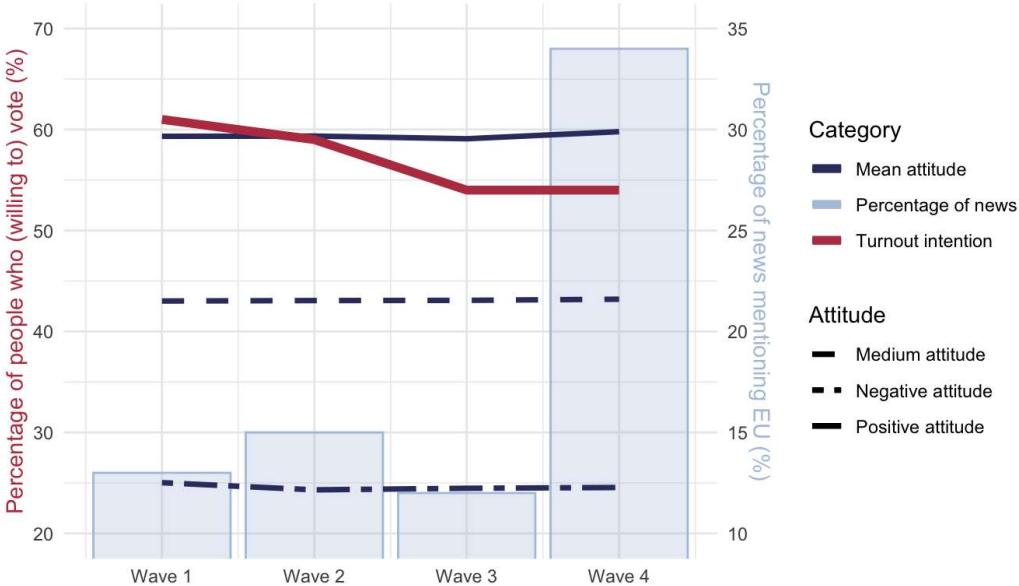
```
#Set color and line type.
```

```
scale_color_manual(values = ycolor) +
  scale_fill_manual(values = yfill) +
  scale_linetype_manual(values = ylinetype) +
```

```
#Set theme.
```

```
theme_minimal() +
  theme(plot.title = element_text(size = "12"), plot.subtitle = element_text(size = "11"), axis.title.y.left = y1title, axis.title.y.right = y2title, plot.caption = element_text(face = "italic", hjust = 0))
```

How does the percentage of news mentioning EU influence people's turnout (intention)?  
And how does people's attitude towards EU influence the relationship?



Data source: European Election Campaign Study, Center for Politics and Communication,  
Amsterdam School of Communication Research ASCoR, University of Amsterdam, September 16, 2014

```
#Save the plot.
ggsave("Visualization_Astrid.pdf")
```

```
## Saving 7 x 5 in image
```

```
rm(y1title, y2title, names, ycolor, yfill, ylinetype)
```

### 3.2.6 Conclusion

Several findings can be drawn from this plot:

1. No matter for people with positive, medium, or negative attitude, their **mean attitude did not change a lot during four waves**. This may show that people's attitude towards EU is hard to change in a shorter period (for example, six months in this survey).
2. Comparing with news in wave 1-3, news in wave 4 had greater possibility of mentioning EU. This pattern indicates that EU was not a frequently mentioned topic in Netherlands media in that year, but **2014 EU election allowed EU to be a focus on media**.
3. Main relationship: From the plot I find that the **change of news coverage for EU does not influence people's turnout intention and real turnout**. Although news coverage for EU fluctuated between wave 1-3 and increased in wave 4, the percentage of participants' that were willing to vote decreased slowly but continuously through four waves.
4. Moderation effect: As stated, people's mean attitude did not change much during the four waves. People's unchanged attitude implies **the lack of relationship between news coverage and attitude**. This finding is in line with the finding of Mutz and Soss (1997), that news has limited ability in influencing public opinion.

## 3.3 Janice's Visualisation

```
time_EU_cleaned <- read_csv("time_EU_cleaned.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
##
## — Column specification —
## cols(
##   .default = col_double(),
##   Vote_int = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```

visual <- time_EU_cleaned %>% select(wave, expnet_1:expnet_4, expnp_1:expnp_10, exptv_1:exptv_10, EU_attitude_1: EU_attitude_9, EU_attitudes2_1:EU_attitudes2_9) %>% as_tibble()

visual<- time_EU_cleaned %>% transmute(wave = as.integer(wave), EU_attitude_1 = as.integer(EU_attitude_1),EU_attitude_2 = as.integer(EU_attitude_2), EU_attitude_3 = as.integer(EU_attitude_3), EU_attitude_4 = as.integer(EU_attitude_4), EU_attitude_5 = as.integer(EU_attitude_5), EU_attitude_6 = as.integer(EU_attitude_6), EU_attitude_7 = as.integer(EU_attitude_7), EU_attitude_8 = as.integer(EU_attitude_8), EU_attitude_9 = as.integer(EU_attitude_9), expnet_1 = as.integer(expnet_1), expnet_2 = as.integer(expnet_2), expnet_3 = as.integer(expnet_3), expnet_4 = as.integer(expnet_4), exptv_1 = as.integer(exptv_1), exptv_2 = as.integer(exptv_2), exptv_3 = as.integer(exptv_3), exptv_4 = as.integer(exptv_4), exptv_5 = as.integer(exptv_5), exptv_6 = as.integer(exptv_6), exptv_7 = as.integer(exptv_7), exptv_8 = as.integer(exptv_8), exptv_9 = as.integer(exptv_9), exptv_10 = as.integer(exptv_10), expnp_1 = as.integer(expnp_1), expnp_2 = as.integer(expnp_2), expnp_3 = as.integer(expnp_3), expnp_4 = as.integer(expnp_4), expnp_5 = as.integer(expnp_5), expnp_6 = as.integer(expnp_6), expnp_7 = as.integer(expnp_7), expnp_8 = as.integer(expnp_8), expnp_9 = as.integer(expnp_9), expnp_10 = as.integer(expnp_10) ) %>%
  transmute ( personal_EUatt = rowMeans (.[c(
    "EU_attitude_1",
    "EU_attitude_2",
    "EU_attitude_3",
    "EU_attitude_4",
    "EU_attitude_5",
    "EU_attitude_6",
    "EU_attitude_7",
    "EU_attitude_8",
    "EU_attitude_9"
  )]*0.7),
  wave,expnet_2,exptv_1,exptv_2,expnp_1,expnp_2,expnp_6
)

```

```
cont <- read_csv("content.csv")
```

```

## 
## — Column specification ——————
## cols(
##   V2 = col_double(),
##   V4 = col_double(),
##   V11 = col_double(),
##   V12 = col_double(),
##   V13 = col_double(),
##   date = col_date(format = "")
## )

```

```

cont <- select(cont, V4, V11:V13, date )
outlets <- read_csv("Outlet list.csv")

```

```

## 
## — Column specification ——————
## cols(
##   Country = col_character(),
##   Outlet = col_character(),
##   Code = col_double()
## )

```

```

cont <- cont %>% filter (V11!=0 & V11!=9 | V12!=0 & V12!=9 | V13!=0 & V13!=9) %>%
  replace_with_na(replace = list (V11 = c(9,0), V12 = c(9,0), V13 = c(9,0)))

# Pivot the common media outlet data ready for joining with outlets (to get their names)
visual2 <-
  visual %>% pivot_longer(
    cols = c(
      "expnet_2",
      "exptv_1",
      "exptv_2",
      "expnp_1",
      "expnp_2",
      "expnp_6"
    ),
    names_to = "media",
    values_to = "media_val"
  ) %>%
  mutate(
    media = recode(
      media,
      "expnet_2" = 116,
      "expnp_1" = 113,
      "expnp_2" = 111,
      "expnp_6" = 112,
      "exptv_1" = 115,
      "exptv_2" = 114
    )
  ) %>% drop_na()

# Join media to outlets via the media code, calculate means of EU Attitude values and drop unneeded columns
# take distinct values to skip duplicates
vis4 <- visual2 %>%
  left_join(cont, by = c("media" = "V4")) %>%
  left_join(outlets, by = c("media" = "Code")) %>%
  mutate(mediaEU_attitude = rowMeans(.[,c("V11", "V12", "V13")], na.rm = TRUE)) %>%
  select(-c("V11", "V12", "V13", "Country")) %>%
  distinct(.keep_all = TRUE) %>%
  pivot_longer(cols = c("mediaEU_attitude", "personal_EUatt"), names_to = "Eu_stance", values_to = "mediavspersonal")

# Split media publications into those which could have been read at the times of the survey waves
# Calculate and save mediawave numbers analog to the survey waves

end_w1=dmy("26/12/2013")
end_w2=dmy("30/03/2014")
end_w3=dmy("28/04/2014")
end_w4=dmy("02/06/2014")
vis4 <- vis4 %>% mutate (mediawave = case_when (date <= end_w1 ~ 1,
                                                ((date >= end_w1) & (date <= end_w2)) ~ 2,
                                                ((date >= end_w2) & (date <= end_w3)) ~ 3,
                                                ((date >= end_w3) & (date <= end_w4)) ~ 4)) %>%
  distinct( .keep_all = TRUE)

# Prepare multiple plot data for the generalized function
wavecol=c(I("#bebada"),I("purple"),I("red"),I("orange"))
medwavecol=c(I("darkcyan"),I("blue"),I("darkred"),I("chocolate"))
xaxis_name=c("", "", "", "<b>News Outlet")
yaxis_name=c("<b>Opinion on the EU", "", "", "")

# Texts for the legends
mednames = c(
  'Media Stories <br> (until 26/12/2013)',
  'Media Stories <br> (27/12/2013 - 30/03/2014)',
  'Media Stories <br> (31/03/2014 - 28/04/2014)',
  'Media Stories <br> (29/04/2014 - 02/06/2014')
)
names = c(
  'Consumer Survey <br> (13/12/2013 - 26/12/2013)',
  'Consumer Survey <br> (20/03/2014 - 30/03/2014)',
  'Consumer Survey <br> (17/04/2014 - 28/04/2014)',
  'Consumer Survey <br> (26/05/2014 - 02/06/2014')
)

# Function to create split violin plots with media Left and survey right
# The 4 plots encompass the 4 survey waves and media-content data partitioned according

```

```
# to date, where the dates are chosen to match the survey waves

makefigure <- function(data,wavenr,wavecol, medcol, mednames, names,xaxis_name,yaxis_name )
{
  wavefilt=filter(data, wave==wavenr)
  mediawavefilt=filter(data, mediawave==wavenr) #) & media_val>0)

fig0 <- wavefilt %>%
  plot_ly(type = 'violin')

fig <- add_trace(fig0,
  x = ~Outlet[mediawavefilt$Eu_stance == 'mediaEU_attitude'],
  y = ~mediavspersonal[mediawavefilt$Eu_stance == 'mediaEU_attitude'],
  legendgroup = 'mediaEU_attitude',
  scalegroup = 'mediaEU_attitude',
  name = mednames[wavenr],
  side = 'negative',
  box = list(visible = T),meanline = list(visible = T),color = I(medwavecol[wavenr]),
  # points = NULL,
  # pointpos = pointposL[wavenr],
  # jitter = 0,
  scalemode = 'width',
  meanline = list(visible = T),color = I(medwavecol[wavenr]),
  marker = list(
    line = list(
      width = 2,
      color = I(medwavecol[wavenr])
    ),
    symbol = 'line-ns'
  )
)
fig <- fig %>%
  add_trace(
  x = ~Outlet[wavefilt$Eu_stance == 'personal_EUatt'],
  y = ~mediavspersonal[wavefilt$Eu_stance== 'personal_EUatt'],
  legendgroup = 'personal_EUatt',
  scalegroup = 'personal_EUatt',
  name = names[wavenr],
  side = 'positive',
  box = list(visible = T),meanline = list(visible = T ),color = I(wavecol[wavenr]),
  # points = 'all',
  # pointpos = pointposR[wavenr],
  # jitter = 0,
  scalemode = 'width',
  meanline = list(visible = T),color = I("black"),
  marker = list(
    line = list(
      width = 2,
      color = I(wavecol[wavenr])
    ),
    symbol = 'line-ns'
  )
)
fig <- layout(fig,
  title = "<b>Positivity/Negativity Towards the EU (a comparison of Media Stories and Consumer Opinion)</b>",
  xaxis = list(title = xaxis_name[wavenr], zeroline=T),
  yaxis = list(title = yaxis_name[wavenr], zeroline=T),
  violingap = 0.0,
  violingapgap = 0.5,
  violinmode = 'overlay',
  pad = 2
)
return(fig)
}

fig1 = makefigure(vis4,1,wavecol,medwavecol, mednames,names,xaxis_name,yaxis_name)
fig2 = makefigure(vis4,2,wavecol,medwavecol, mednames,names,xaxis_name,yaxis_name)
fig3 = makefigure(vis4,3,wavecol,medwavecol, mednames,names,xaxis_name,yaxis_name)
fig4 = makefigure(vis4,4,wavecol,medwavecol, mednames,names,xaxis_name,yaxis_name)
```

```
## Warning: Ignoring 358 observations
```

```
## Warning: Ignoring 6822 observations
```

```
## Warning: Can't display both discrete & non-discrete data on same axis
```

```
## Warning: 'layout' objects don't have these attributes: 'violingroupgap', 'violinmode', 'pad'
## Valid attributes include:
## 'font', 'title', 'uniformtext', 'autosize', 'width', 'height', 'margin', 'computed', 'paper_bgcolor', 'plot_bgcolor',
'separators', 'hidesources', 'showlegend', 'colorway', 'datarevision', 'uirevision', 'editrevision', 'selectionrevision',
'template', 'modebar', 'newshape', 'activeshape', 'meta', 'transition', '_deprecated', 'clickmode', 'dragmode', 'hovermode',
'hoverdistance', 'spikedistance', 'hoverlabel', 'selectdirection', 'grid', 'calendar', 'xaxis', 'yaxis', 'ternary', 'scene',
'geo', 'mapbox', 'polar', 'radialaxis', 'angularaxis', 'direction', 'orientation', 'editType', 'legend', 'annotations',
'shapes', 'images', 'updatemenus', 'sliders', 'colorscale', 'coloraxis', 'metasrc', 'barmode', 'bargap', 'mapType'
```

```
## Warning: Ignoring 383 observations
```

```
## Warning: Ignoring 6624 observations
```

```
## Warning: Can't display both discrete & non-discrete data on same axis
```

```
## Warning: 'layout' objects don't have these attributes: 'violingroupgap', 'violinmode', 'pad'
## Valid attributes include:
## 'font', 'title', 'uniformtext', 'autosize', 'width', 'height', 'margin', 'computed', 'paper_bgcolor', 'plot_bgcolor',
'separators', 'hidesources', 'showlegend', 'colorway', 'datarevision', 'uirevision', 'editrevision', 'selectionrevision',
'template', 'modebar', 'newshape', 'activeshape', 'meta', 'transition', '_deprecated', 'clickmode', 'dragmode', 'hovermode',
'hoverdistance', 'spikedistance', 'hoverlabel', 'selectdirection', 'grid', 'calendar', 'xaxis', 'yaxis', 'ternary', 'scene',
'geo', 'mapbox', 'polar', 'radialaxis', 'angularaxis', 'direction', 'orientation', 'editType', 'legend', 'annotations',
'shapes', 'images', 'updatemenus', 'sliders', 'colorscale', 'coloraxis', 'metasrc', 'barmode', 'bargap', 'mapType'
```

```
## Warning: Ignoring 465 observations
```

```
## Warning: Ignoring 5868 observations
```

```
## Warning: Can't display both discrete & non-discrete data on same axis
```

```
## Warning: 'layout' objects don't have these attributes: 'violingroupgap', 'violinmode', 'pad'
## Valid attributes include:
## 'font', 'title', 'uniformtext', 'autosize', 'width', 'height', 'margin', 'computed', 'paper_bgcolor', 'plot_bgcolor',
'separators', 'hidesources', 'showlegend', 'colorway', 'datarevision', 'uirevision', 'editrevision', 'selectionrevision',
'template', 'modebar', 'newshape', 'activeshape', 'meta', 'transition', '_deprecated', 'clickmode', 'dragmode', 'hovermode',
'hoverdistance', 'spikedistance', 'hoverlabel', 'selectdirection', 'grid', 'calendar', 'xaxis', 'yaxis', 'ternary', 'scene',
'geo', 'mapbox', 'polar', 'radialaxis', 'angularaxis', 'direction', 'orientation', 'editType', 'legend', 'annotations',
'shapes', 'images', 'updatemenus', 'sliders', 'colorscale', 'coloraxis', 'metasrc', 'barmode', 'bargap', 'mapType'
```

```
## Warning: Ignoring 626 observations
```

```
## Warning: Ignoring 5886 observations
```

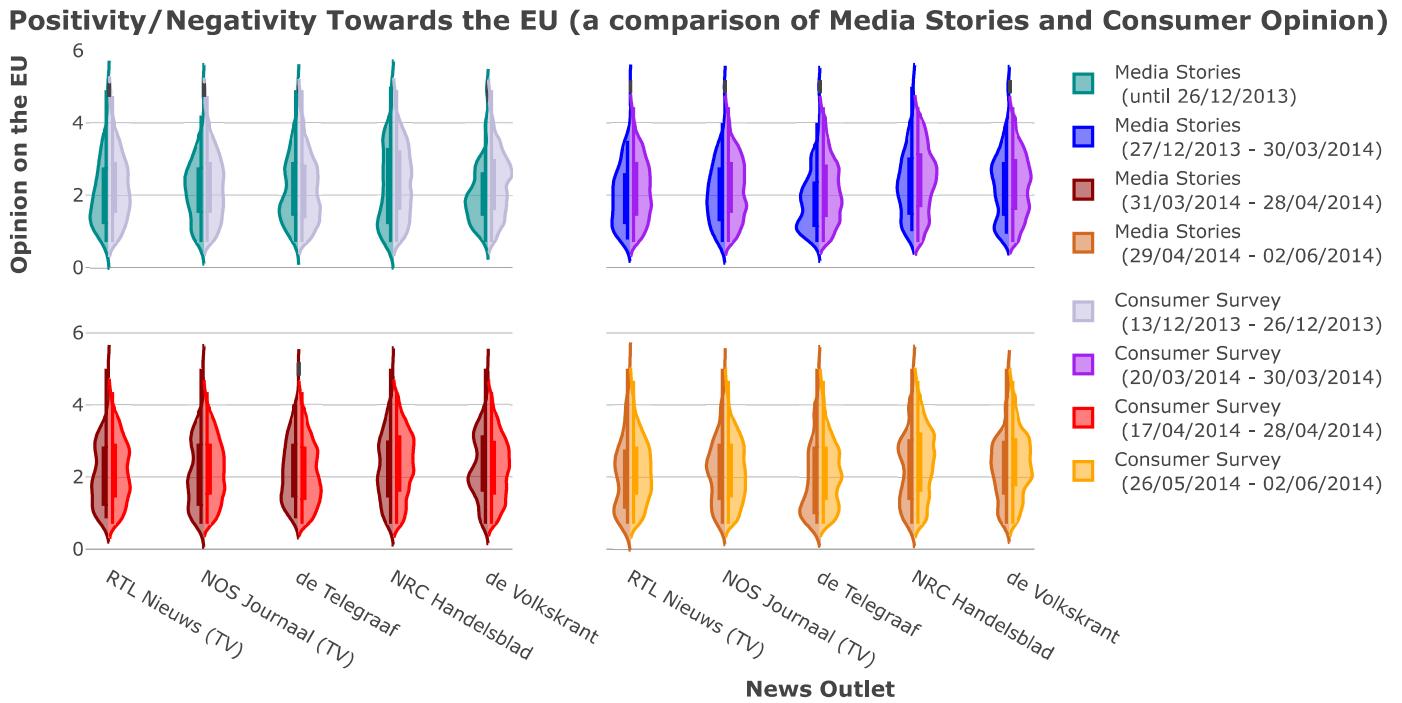
```
## Warning: Can't display both discrete & non-discrete data on same axis
```

```
## Warning: 'layout' objects don't have these attributes: 'violingroupgap', 'violinmode', 'pad'
## Valid attributes include:
## 'font', 'title', 'uniformtext', 'autosize', 'width', 'height', 'margin', 'computed', 'paper_bgcolor', 'plot_bgcolor',
'separators', 'hidesources', 'showlegend', 'colorway', 'datarevision', 'uirevision', 'editrevision', 'selectionrevision',
'template', 'modebar', 'newshape', 'activeshape', 'meta', 'transition', '_deprecated', 'clickmode', 'dragmode', 'hovermode',
'hoverdistance', 'spikedistance', 'hoverlabel', 'selectdirection', 'grid', 'calendar', 'xaxis', 'yaxis', 'ternary', 'scene',
'geo', 'mapbox', 'polar', 'radialaxis', 'angularaxis', 'direction', 'orientation', 'editType', 'legend', 'annotations',
'shapes', 'images', 'updatemenus', 'sliders', 'colorscale', 'coloraxis', 'metasrc', 'barmode', 'bargap', 'mapType'
```

```
## Warning: Can't display both discrete & non-discrete data on same axis
```

```
## Warning: Can't display both discrete & non-discrete data on same axis
```

```
## Warning: 'layout' objects don't have these attributes: 'violingroupgap', 'violinmode', 'pad'
## Valid attributes include:
## 'font', 'title', 'uniformtext', 'autosize', 'width', 'height', 'margin', 'computed', 'paper_bgcolor', 'plot_bgcolor',
## 'separators', 'hidesources', 'showlegend', 'colorway', 'datarevision', 'uirevision', 'editrevision', 'selectionrevision',
## 'template', 'modebar', 'newshape', 'activeshape', 'meta', 'transition', '_deprecated', 'clickmode', 'dragmode', 'hovermode',
## 'hoverdistance', 'spikedistance', 'hoverlabel', 'selectdirection', 'grid', 'calendar', 'xaxis', 'yaxis', 'ternary', 'scene',
## 'geo', 'mapbox', 'polar', 'radialaxis', 'angularaxis', 'direction', 'orientation', 'editType', 'legend', 'annotations',
## 'shapes', 'images', 'updatemenus', 'sliders', 'colorscale', 'metasrc', 'barmode', 'bargap', 'mapType'
```



### 3.3 Analysis of the Survey Regarding EU Positivity/Negativity and Media Stories at the Time of the Survey Waves

The 2013/2014 survey conducted in Belgium and the Netherlands provides results which help us to analyse some common assumptions about the media and in particular to reveal the opinions of consumers of those media over a period of approximately 6 months in the run up and subsequent to the 2014 EU elections held on 25th May. An indication of the answers to two questions can be illustrated in the violin plot visualisation regard to the questions.

Q1: Does positivity/negativity of the EU by media outlets affect positivity/negativity of EU opinion by their consumers? Q2: How does positivity/negativity of the media and of the consumers towards the EU change as the EU parliamentary election approaches?

The so-called split violin plot contains a great deal of information: \* The density of the media positivity (on the left) versus the consumer opinion (on the right) can be shown by the width of the violin at different heights \* The width of each side of the violin indicates at the top the most positivity towards the EU, a broader violin at the bottom indicates negativity towards the EU \* The time period for the survey and media content analysis is split into 4 periods, which are distinguished by the different colours and grouping of the violins

Through the split in the violin, two sets of data can be directly compared, representing the EU positivity of stories from each outlet (on the left side of the violin) and the EU positivity of consumers of media for outlet (on the right hand side of the violin plot). Additionally, to help answer the second question above, as the positivity/negativity towards the EU changes as the election approaches, four subplots were created in order to show the evolution over time with the exact dates in the legend showing the evolution from wave one (i.e. six months before the election) to wave four (immediately after the election). It is worth noting that within each violin, on each side there is a darker rectangle which providing additional statistical information. A mouse-over action reveals precise numerical details to each and every side of each violin, namely the minimum, maximum, median and both the first and third quartile (also intuitively visible by the vertical position and length of the box).

### 3.3 Analysis of Media Story Development

Regarding the media stories (left side of each violin, seen best in isolation by clicking to deselect the consumer survey legend group) it is immediately apparent that most outlets begin (green quadrant) with a fairly balanced, if less than enthusiastic stance towards the EU. NRC Handelsblad has the widest spread of positive and negative articles and also the highest mean positivity. De Volksrant is most negative of all, with a mean of 2.055 compared with ca. 2.1 - 2.3 from the other outlets. Moving to the second (blue) phase Q1/2014, there are small modifications of the outlets' stances to be seen, but most noticeably in the case of de Volkskrant, which becomes markedly more positive (mean 2.3), NRC Handelsblad has a surge of mild positivity, whilst de Telegraaf drops from a mean of 2.2 to 1.98 becoming the clear leader in negativity.

The other 3 outlets remain neutral and balanced throughout, including for the following 2 phases, whilst de Telegraaf regains some optimism in April only to return to the initial lower Q1 level after the election. De Volkskrant loses the EU optimism marginally in April and again, marginally post-election.

### 3.3 Analysis of Consumer Opinion

Turning to the consumer survey (right side of each violin, seen best in isolation by clicking to deselect the media story legend group) there is a very homogenous spread of opinion between the levels of 1.5 and 3 and with a higher mean than the media stories might imply. As with the outlet stories the widest spread of opinion was by the NRC Handelsblad. It is most notable that readers of de Volkskrant show more initial positivity than the stories from that outlet and maintain this throughout the period of the survey. The slight relative negativity of NRC Handelsblad is maintained throughout.

### 3.3 In Summary

The results show that there is a correlation between media opinion and consumer opinion, but the rather homogeneous and objectively neutral results (nowhere was a mean significantly below 2 or above 2.4) could indicate that the correlation is more by chance than born of influence through the media articles. Consumers exhibit a significant spread in their views of the European Union and are equally spread regarding negativity/positivity. The media outlets also exhibit a fairly even reporting regarding positivity towards the EU. Notable exceptions are de Volkskrant and de Telegraaf, which both seem to be more negative in their reporting initially. For de Volkskrant this changes over time as the reporting gets more positive the more the election approaches. One could even argue that the outlet modifies its opinion to match the positive stance of its readership. The opinion of de Volkskrant readers does, in any case, not seem to show much of an evolution over time with the bulk of the readership having a somewhat neutral opinion of the EU, but more positive than the outlet in 3 of the 4 measurement periods.

In general it can be said that the surveyed people seem to be uninterested or ambivalent towards the EU. It is however possible that this relationship has changed with significant events having occurred after the surveyed period. Both the Brexit referendum in 2016, almost two full years after the survey was conducted and the SARS COVID-19 may have changed the opinion towards the EU again.

## 4 Bibliography

Murre, J., & Dros, J. (2015). Replication and Analysis of Ebbinghaus' Forgetting Curve. *PLOS ONE*, 10(7), e0120644. doi: 10.1371/journal.pone.0120644

Mutz, D., & Soss, J. (1997). Reading Public Opinion: The Influence of News Coverage on Perceptions of Public Sentiment. *Public Opinion Quarterly*, 61(3), 431. doi: 10.1086/297807