European Social Survey: Exploratory data analysis

Data

```r
library(tidyverse)
library(essurvey)
library(labelled)
library(rstan)
library(rgdal)
library(doBy)
set_email('trubetskoy.vasa@gmail.com')
```

Round 8 contains questions refering to climate. We can extract these.

```r
round_8 <- import_rounds(8, )

base_variables <- c('idno', 'cntry')

interesting_variables <- c('eduyrs')

questions <- var_label(round_8) %>%
  as_data_frame() %>%
  gather(key = 'variable_name', value = 'description')
```

```
## Warning: `as_data_frame()` is deprecated, use `as_tibble()` (but mind the new semantics).
## This warning is displayed once per session.
```

```r
climate_questions <- questions %>%
  filter(str_detect(description, 'climate'))

responses <- round_8 %>%
  select(one_of(c(base_variables, climate_questions$variable_name, interesting_variables)))

# is this the right thing to do?
responses <- recode_missings(responses, c("Don't know", "Refusal"))

country_summaries <- responses %>%
  group_by(cntry) %>%
  summarise(mean_inctxff = mean(inctxff, na.rm = T))
```

Mapping

Starting point: https://gist.github.com/stared/fbca436c885c430a314a

```r
library(maptools)
```

```
## Checking rgeos availability: FALSE
##      Note: when rgeos is not available, polygon geometry     computations in maptools depend on gpcl
##      which has a restricted licence. It is disabled by default;
##      to enable gpclib, type gpclibPermit()
```

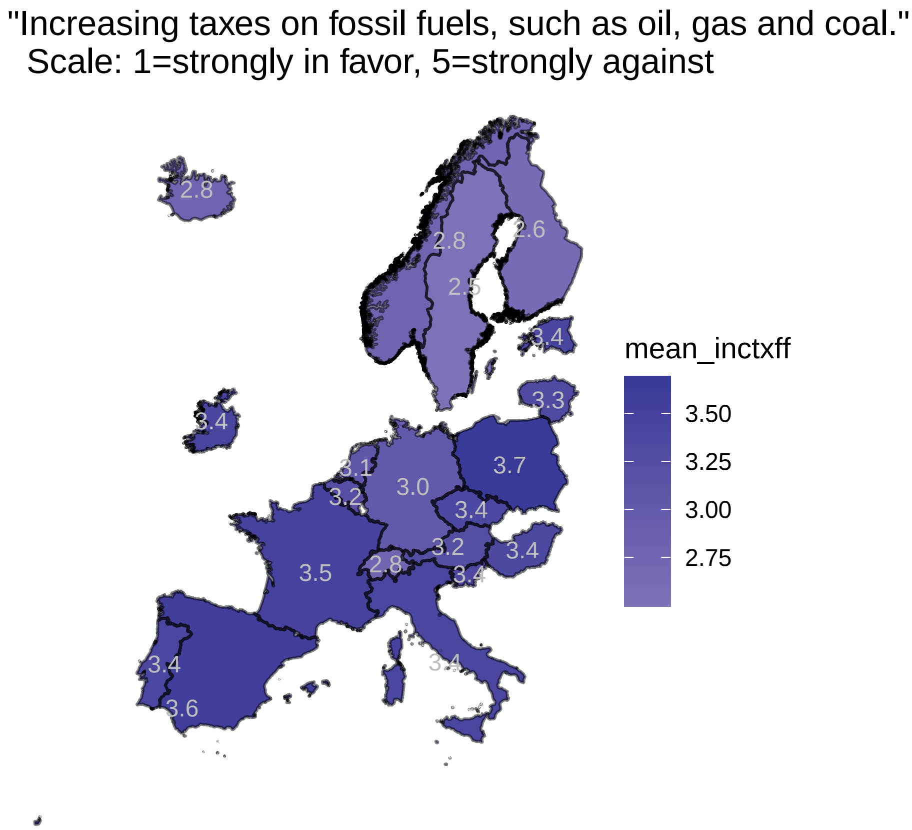
```r
gpclibPermit()
```

```
## Warning in gpclibPermit(): support for gpclib will be withdrawn from
## maptools at the next major release

## [1] TRUE
```

```
eurMap <- readOGR('../data/NUTS_RG_01M_2016_3035.shp')
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/vasa/Projects/the_climate_question/data/NUTS_RG_01M_2016_3035.shp", layer: "NUTS_RG_(
## with 2016 features
## It has 5 fields
```

```
eurMap <- subset(eurMap, nchar(as.character(NUTS_ID)) == 2)

eurMapDf <- fortify(eurMap, region='NUTS_ID')

# merge map and data
eurMapDataDf <- merge(eurMapDf, country_summaries, by.x="id", by.y="cntry")
# sort, so that polygons are drawn correctly
eurMapDataDf <- eurMapDataDf[order(eurMapDataDf$order),]

# limit data to main Europe
eurMapDataDf <- subset(eurMapDataDf, long > 2e6 & long < 6e6 & lat > 1e6 & lat < 6e6)

# add text; instead of mean I do middle (not to be to biased towards detailed coastlines)
middle = function (x) {
  (max(x) + min(x)) / 2
}
txtVal <- summaryBy(long + lat + mean_inctxff ~ id, data=eurMapDataDf, FUN=middle, keep.names=T)

p <- ggplot(data=eurMapDataDf) +
  geom_polygon(aes(x=long, y=lat, group=group, fill=mean_inctxff)) +
  geom_path(aes(x=long, y=lat, group=group), color='black', alpha=.5) +
  geom_text(aes(x=long, y=lat, label=sprintf("%.1f", mean_inctxff)), data=txtVal, col="gray", cex=3) +
  scale_fill_gradient2() +
  theme_void() +
  coord_equal() +
  ggtitle('"Increasing taxes on fossil fuels, such as oil, gas and coal."\n  Scale: 1=strongly in favor
p
```



Weirdly the UK is missing on this map. ostok119 ## Modeling

Questions

How does income affect climate views?

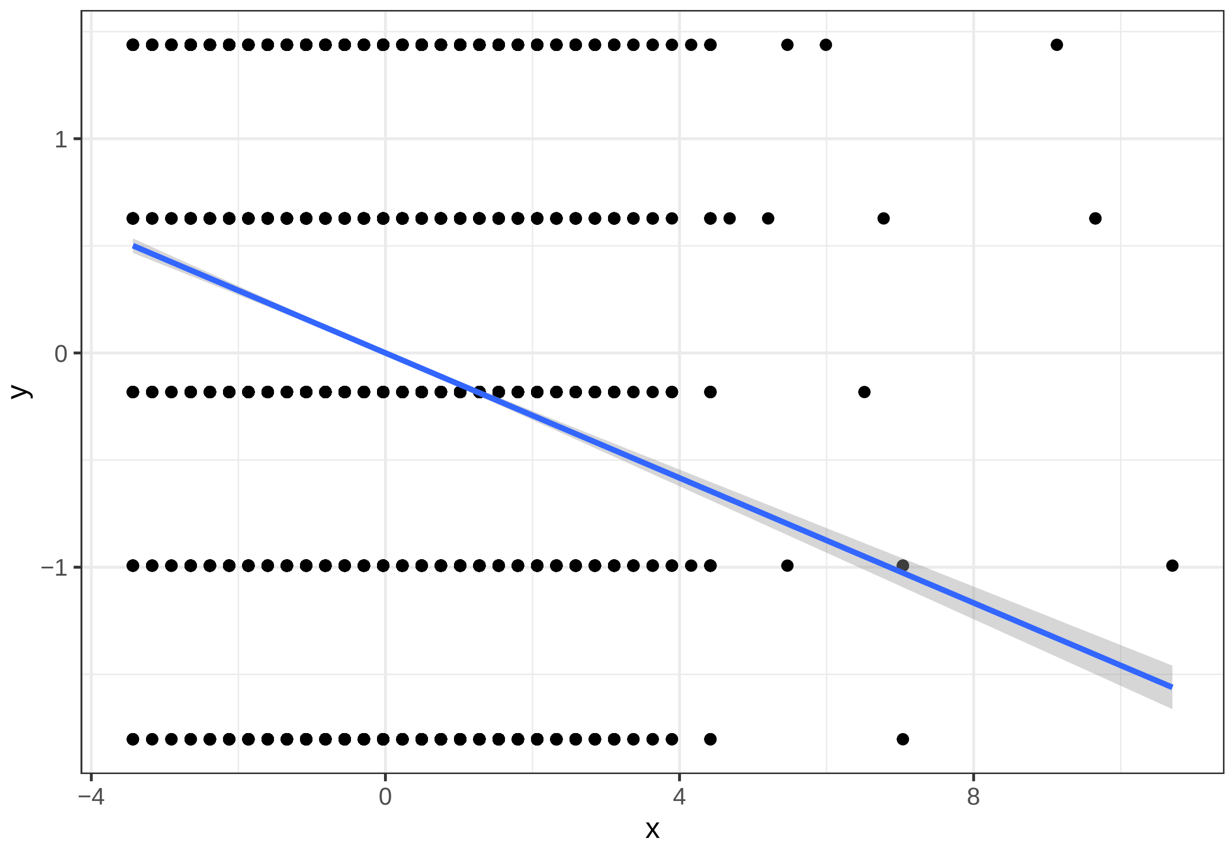
How does level of education affect climate views?

How do climate views differ by country?

Model 01: linear everything

How unrealistic! We can take a look anyway:

```
preprocessed_data <- responses %>%
  select(c(eduyrs, inctxff)) %>%
  drop_na() %>%
  mutate(x = scale(eduyrs) %>% as.vector(), y = scale(inctxff) %>% as.vector())

ggplot(preprocessed_data, aes(x=x, y=y)) +
  geom_point() +
  geom_smooth(method='lm',formula=y~x) +
  theme_bw()
```



Yikes. Lets fit this thing anyway.

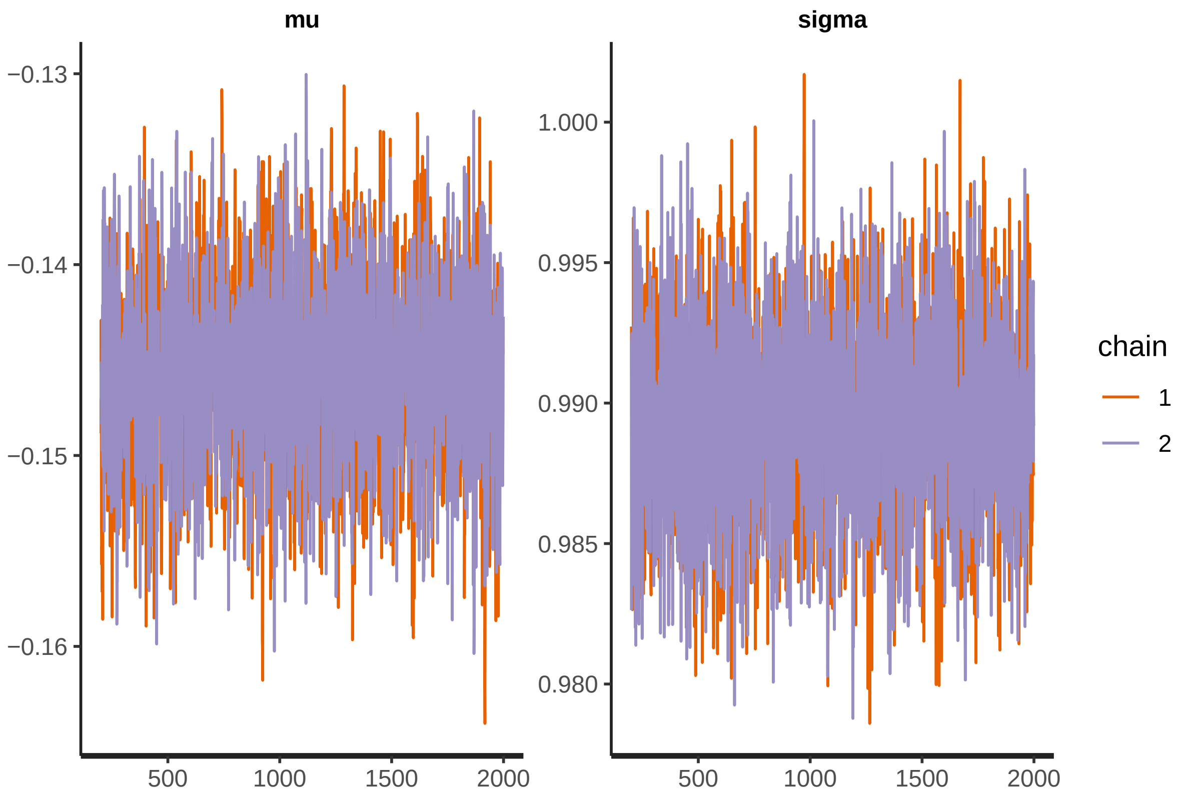
```
stan_data <- preprocessed_data %>%
  select(c(x, y)) %>%
  as.list()

stan_data[['N']] <- length(stan_data[['y']])

edu_inctxf_fit <-stan(file = '../models/linear_education_climate.stan',
  data = stan_data,
  iter = 2000,
  warmup = 200,
  chains = 2)
```

```
##
## SAMPLING FOR MODEL 'linear_education_climate' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.006648 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 66.48 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  201 / 2000 [ 10%]  (Sampling)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Sampling)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Sampling)
```

```
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Sampling)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 2.33629 seconds (Warm-up)
## Chain 1:                18.6054 seconds (Sampling)
## Chain 1:                20.9416 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'linear_education_climate' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.001788 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 17.88 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  201 / 2000 [ 10%]  (Sampling)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Sampling)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Sampling)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Sampling)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 2.09905 seconds (Warm-up)
## Chain 2:                18.4826 seconds (Sampling)
## Chain 2:                20.5817 seconds (Total)
## Chain 2:
```

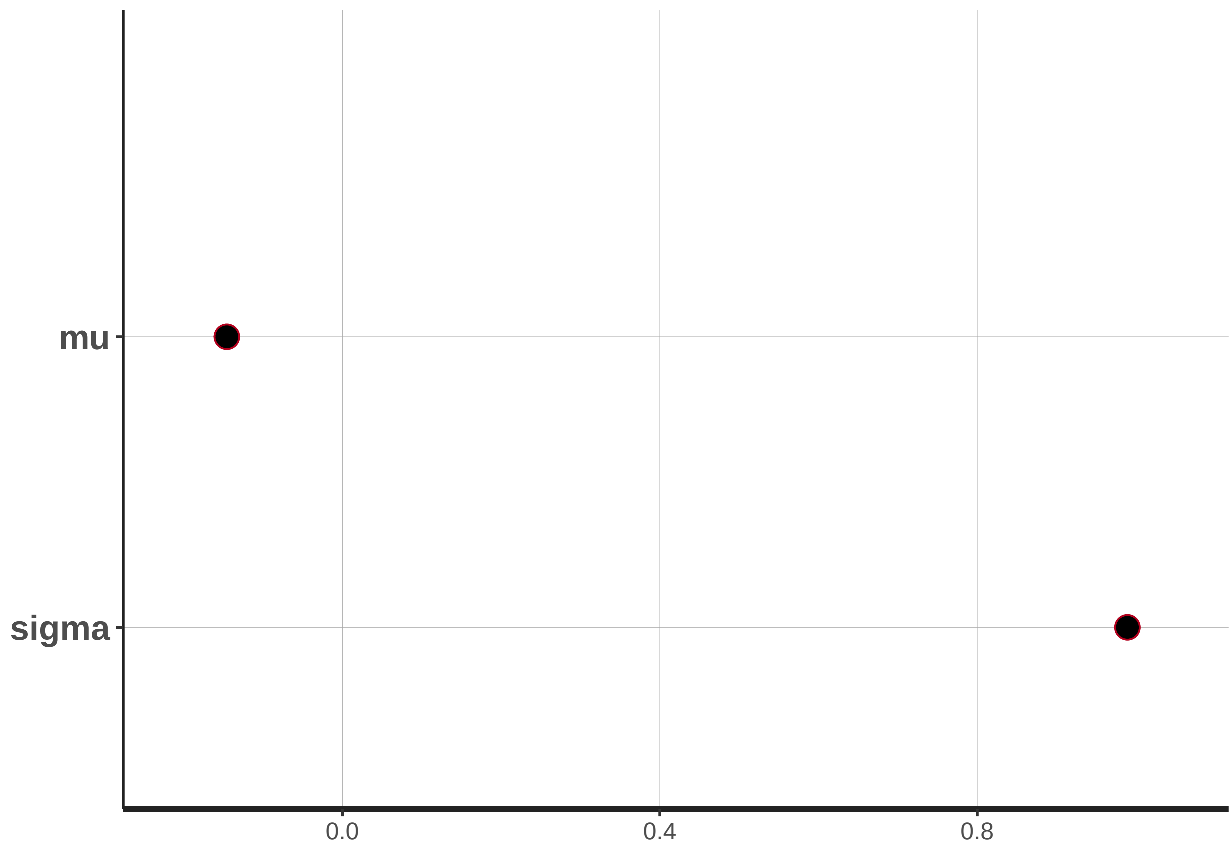
Look at the trace and parameter estimates.

```r
traceplot(edu_inctxf_fit)
```




```
plot(edu_inctxf_fit)
```

```
## ci_level: 0.8 (80% intervals)
## outer_level: 0.95 (95% intervals)
```



Model 02: ordinal response? ordinal predictor?

Model 03: survey adjustments?

Model 04: hierarchical effects?