Assignment #3

Katie Levesque, Sarah Unbehaun, Meerim Ruslanova

April 14, 2016

Quick recap

This project aims to investigate the relationship between happiness levels across German Federal States (Bundeslaender) and among individuals over the time horizon from 1990 to 2012. More specifically, this projects explores whether the state level emissions, as well as personal characteristics, affect life satisfaction of German citizens? The hypotheses state:

> *H1: Bundeslaender with higher emissions will have lower reported levels of health, well-being, or life satisfaction.*
>
> *H2: Reported levels of health, well-being or life satisfaction will reflect changes in emissions in line with hypothesis above, i.e. as emissions decline, reported levels of health, well-being, and life satisfaction will rise.*

Data

The individual-level data is provided by the German Socio-Economic Panel Data GSOEP conducted by the German Institute for Economic Research DIW. Due to the confidentiality restrictions, DIW could only supply a shortened sample with prior specified variables in a *.dta* format. Therefore, the GSOEP dataset is stored on the local drives and GitHub Climate-Happiness Repository. The original GSOEP file is cleaned and transformed into a shorter dataset with the help of the State Do-File. The short dataset contains the information on the main variables: reported levels of life satisfaction (on a scale from 0 to 10), subjective concerns about the environment, age, gender, employment, family status, and state residence of a respondent. Detailed labels and description of the variables are given in the GSOEP codebook. Likewise, all GSOEP related files are stored on the GitHub server.

The state-level data, on the other hand, is gathered from three web-based sources: State Initiative for Core Indicators LIKI, Statista.com, Environmental-Economic Accounting of the Bundeslaender UGRdL and Agency for Renewable Agency of North Rhine-Westphalia AfEE.

A university subscription to *Statista.com* enabled access to historic state emissions from 1990 to 2012 for most of the Bundeslaender except North Rhine-Westphalia (NRW). Since the website allows data downloads only in *Excel* and provides no unique URLs for each of them, 15 individual files were downloaded manually on a local machine, while manipulations were conducted with the help of R loops. The information on NRW was slightly involved more intensive research and data handling. Finally, the NRW annual emissions were gathered and combined from the UGRdL (from 1990 to 2000) and *AfEE* (from 2000 to 2012) with R web-scraping functions. Fortunately, emissions are measured in the same units (annually emitted Carbon dioxide tons per capita). Hence, the yielded data frame of emissions is comprehensive and consistent, although there are missing observations on some years.

Simultaneously, the state efforts to curb their emissions and preserve local environment are reflected in their renewable energy indicators. This information is measured in percentage of renewables in the annual primary energy consumption, final energy consumption, and electricity consumption. The indicators had to be downloaded manually from *LIKI* in three separate excel files, which later on were cleaned, transformed and reshaped into suitable data frames in R.

After the names of the Bundeslaender and the time frame of the three produced data frames match, they are easily merged in R into a final data set.

Descriptive Statistics

```
##      Baden-WÃ¼rttemberg                  Bayern                  Berlin
##                  15935                   20889                    6328
##            Brandenburg                  Bremen                 Hamburg
##                   8334                    1075                    1468
##                 Hessen Mecklenburg-Vorpommern           Niedersachsen
##                  10886                    4073                    7930
##    Nordrhein-Westfalen         Rheinland-Pfalz                Saarland
##                  22048                    7460                    1208
##                Sachsen          Sachsen-Anhalt      Schleswig-Holstein
##                  11955                    8765                    4317
##             ThÃ¼ringen
##                   8159
```

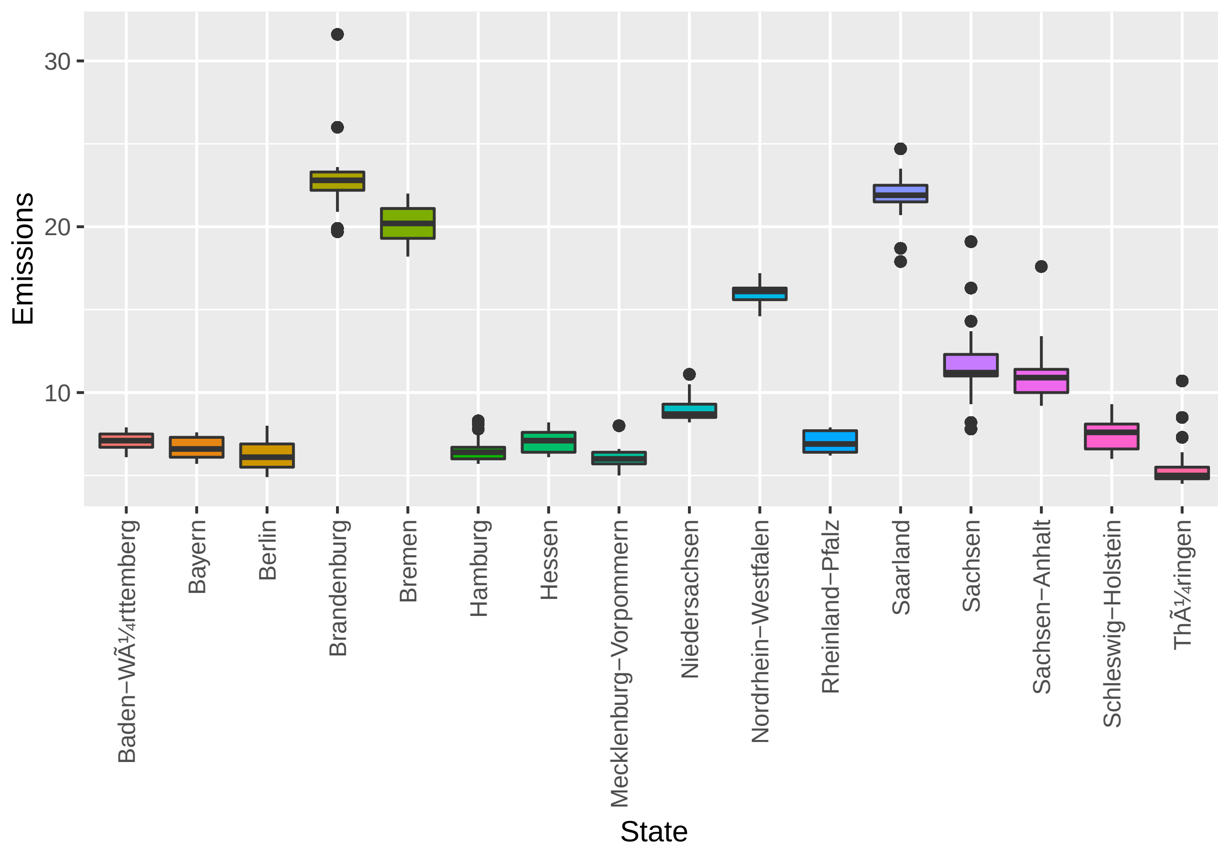
The data has a wide range in terms of the number of observations for each federal state. If we look a little closer, separating them by year, we see that some states are missing observations for a few years:

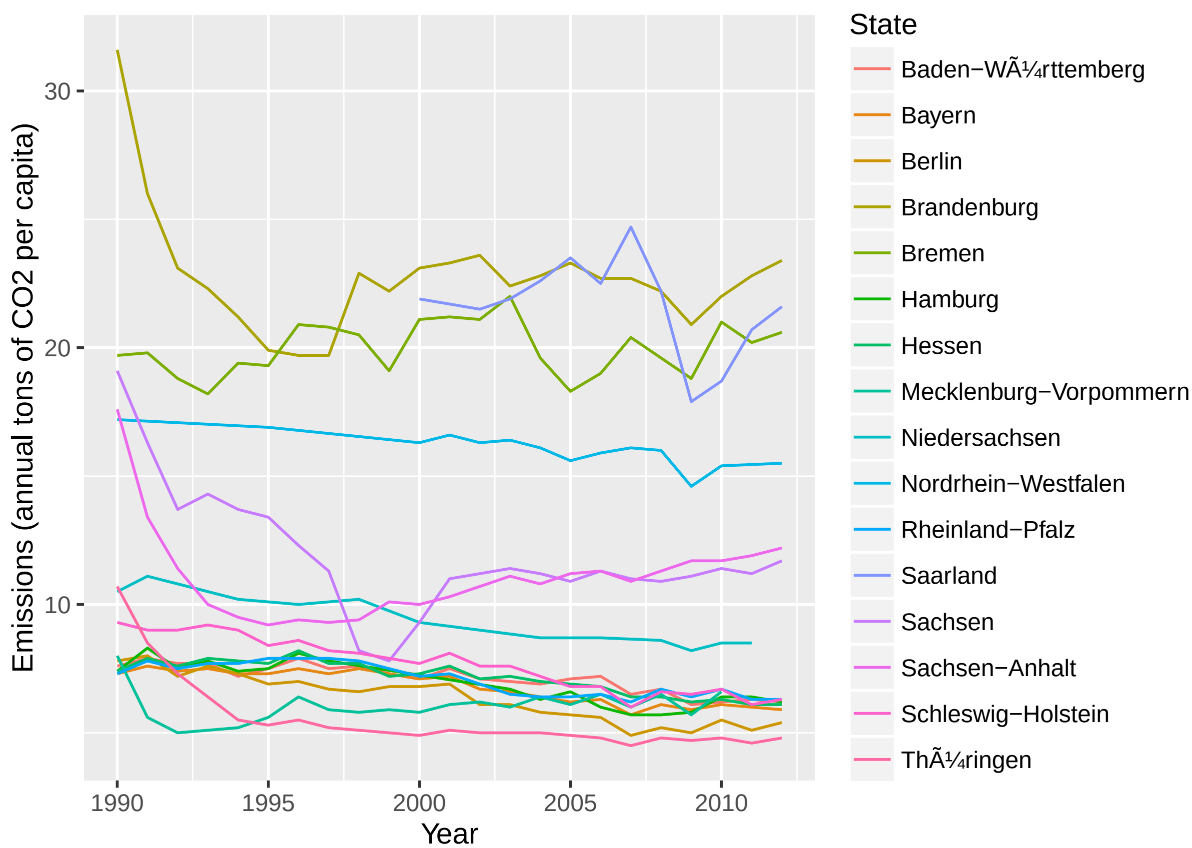
```
##                          Year
## State                     1990 1991 1992 1993 1994 1995 1996 1997 1998 1999
##   Baden-WÃ¼rttemberg       407  378  344  335  355  378  537  524  570  530
##   Bayern                   536  517  512  492  515  561  643  596  692  652
##   Berlin                   251  233  234  233  226  233  225  215  222  212
##   Brandenburg              371  337  315  307  294  276  273  262  275  268
##   Bremen                    34   32   33   30   32   35   32   31   34   24
##   Hamburg                   41   37   37   33   36   33   42   41    0    0
##   Hessen                   361  336  329  312  300  325  360  330  390  353
##   Mecklenburg-Vorpommern   219  204  178  170  175  172  157  153  171  165
##   Niedersachsen            368  347    0    0  310    0  403    0  462    0
##   Nordrhein-Westfalen      768    0    0    0    0  738    0    0    0    0
##   Rheinland-Pfalz          263  252  241  241  262  284  287  293  330  300
##   Saarland                   0    0    0    0    0    0    0    0    0    0
##   Sachsen                  568  522  491  472  457  428  419  395  432  419
##   Sachsen-Anhalt           433  407  396  372  353  335  309  308  318  317
##   Schleswig-Holstein       101  109   98   97  100   93  100   96  129  111
##   ThÃ¼ringen               382  356  346  329  327  314  317  321  347  331
##                          Year
## State                     2000 2001 2002 2003 2004 2005 2006 2007 2008 2009
##   Baden-WÃ¼rttemberg      1116  970 1055  957  887  862  965  847  789  851
##   Bayern                  1449 1243 1387 1265 1189 1088 1187 1098 1010 1142
##   Berlin                   377  344  361  346  323  305  300  291  276  299
##   Brandenburg              462  451  463  453  434  399  416  394  366  409
##   Bremen                    71   53   74   74   69   54   47   48   48   64
##   Hamburg                    0    0    0  127  125  122  127  117  111  110
##   Hessen                   750  625  726  629  585  541  587  528  505  558
##   Mecklenburg-Vorpommern   239  207  223  221  207  195  235  204  200  202
##   Niedersachsen            843    0  849    0  743    0  778    0  694  773
##   Nordrhein-Westfalen     2156 1895 2034 1859 1747 1616 1719 1575 1482 1567
##   Rheinland-Pfalz          489  432  442  392  382  353  370  322  310  317
##   Saarland                 133  113  114  108  100   97   97   86   73   73
##   Sachsen                  716  632  640  614  595  552  586  543  510  550
##   Sachsen-Anhalt           503  481  469  436  424  406  410  399  357  361
##   Schleswig-Holstein       295  266  318  272  274  256  271  246  209  222
##   ThÃ¼ringen               496  436  420  404  390  357  367  341  324  338
```

```
##                            Year
## State                      2010 2011 2012
##   Baden-WÃ¼rttemberg        730  731  817
##   Bayern                    998  995 1122
##   Berlin                    268  259  295
##   Brandenburg               358  363  388
##   Bremen                     58   46   52
##   Hamburg                   109  106  114
##   Hessen                    488  450  518
##   Mecklenburg-Vorpommern    176    0    0
##   Niedersachsen             700  660    0
##   Nordrhein-Westfalen      1410    0 1482
##   Rheinland-Pfalz           281  289  328
##   Saarland                   66   72   76
##   Sachsen                   475  453  486
##   Sachsen-Anhalt            319  314  338
##   Schleswig-Holstein        198  211  245
##   ThÃ¼ringen                304  291  321
```

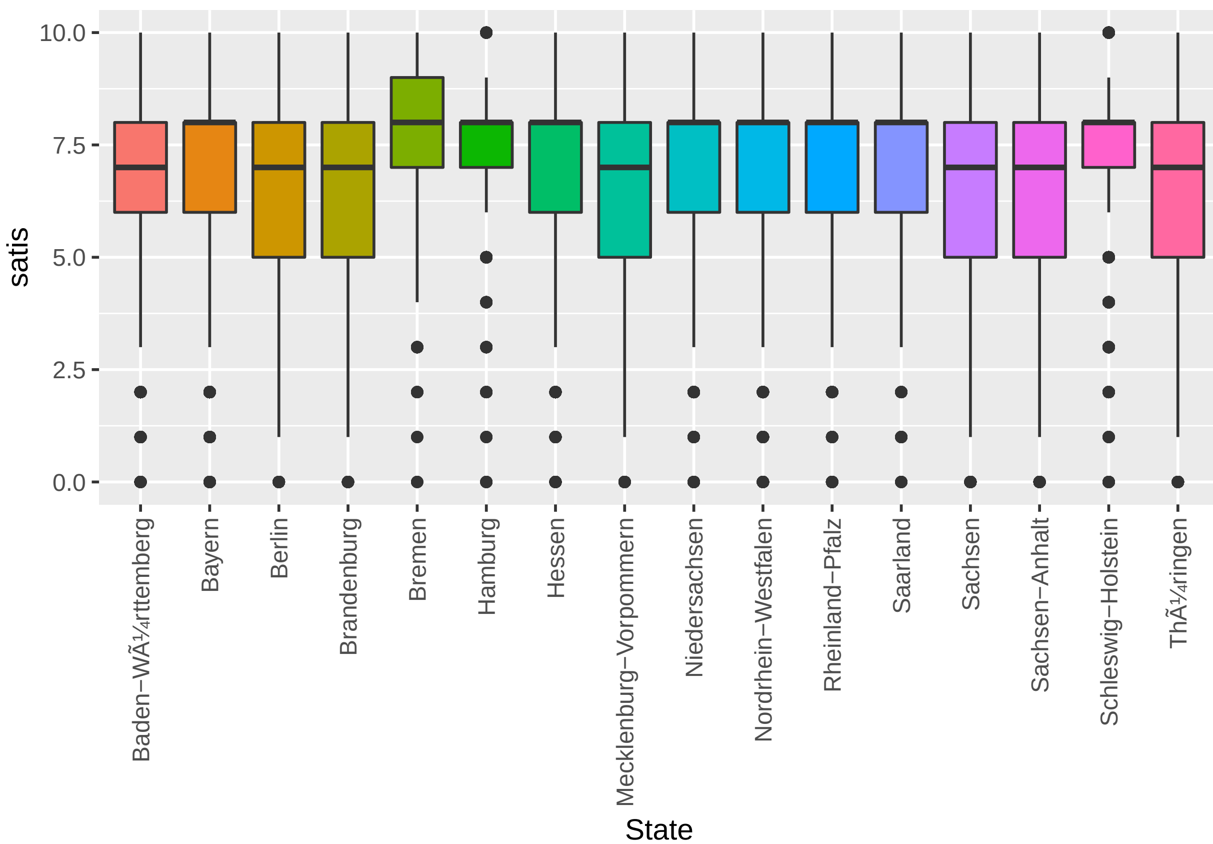
Analysis using Saarland, Nordrhein-Westfalen, and Hamburg may require some adjustment with the missing years in mind.

Because we are looking at emissions and happiness over a period of years, the values of emissions for each Bundesland also vary, which we can see over time (noting that Saarland is again missing data for some years).





The other primary variable of interest is life satisfaction. These plots show, again, variation within each state and the variation in life satisfaction across years, mapped as the mean of observations from each state.




```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

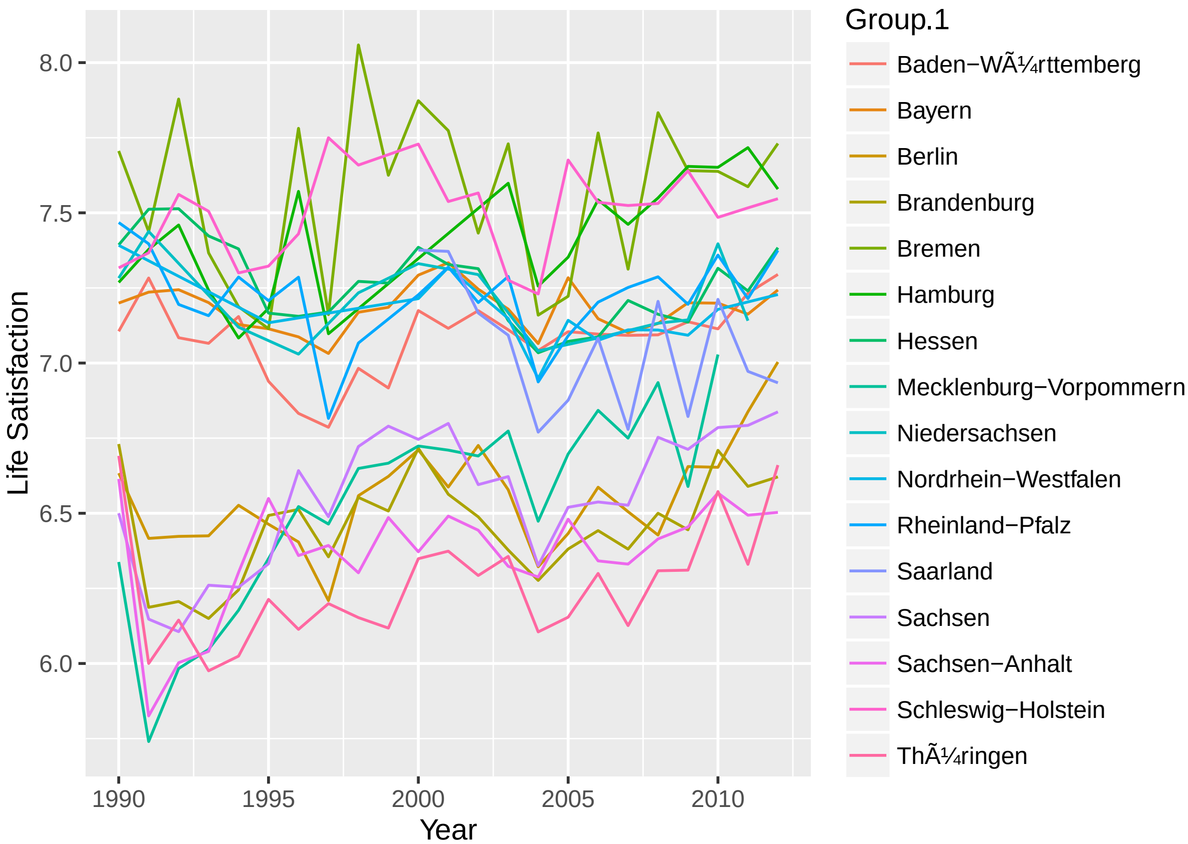
```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

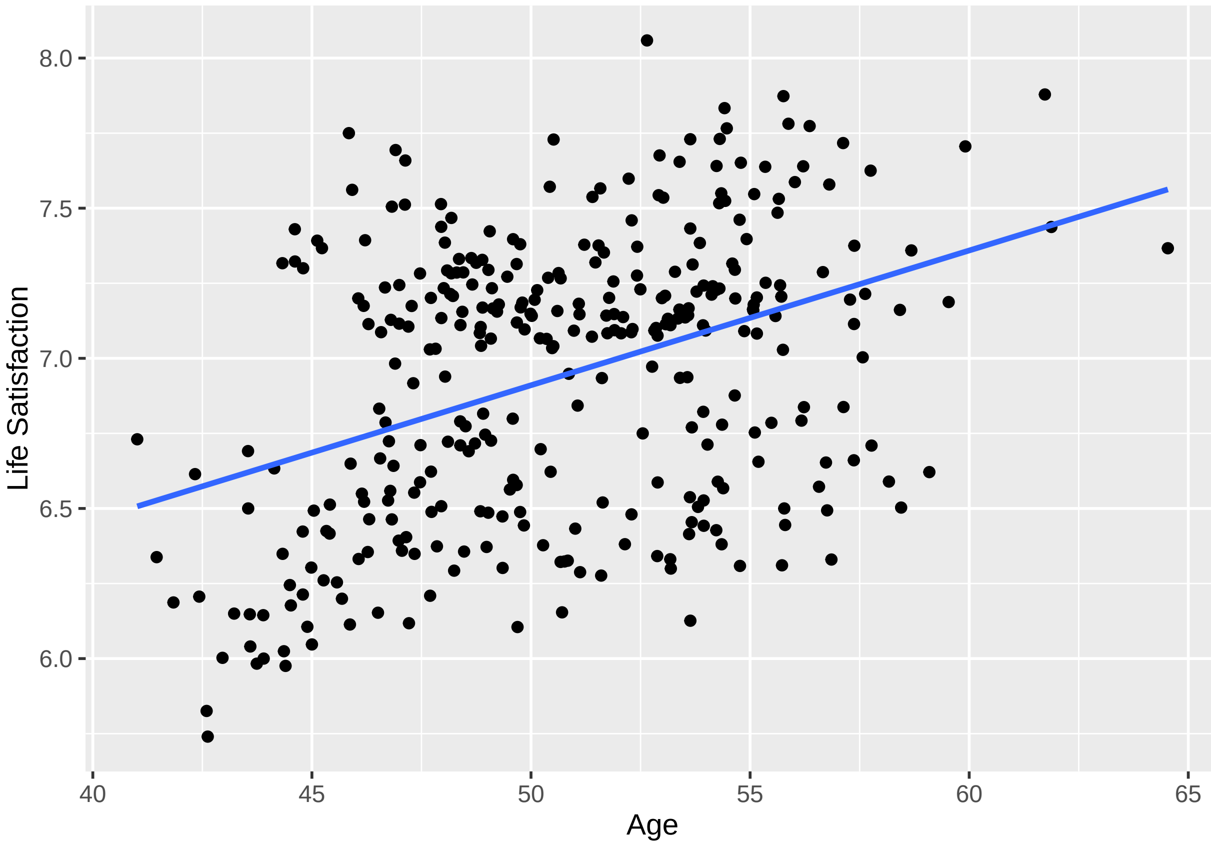
```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

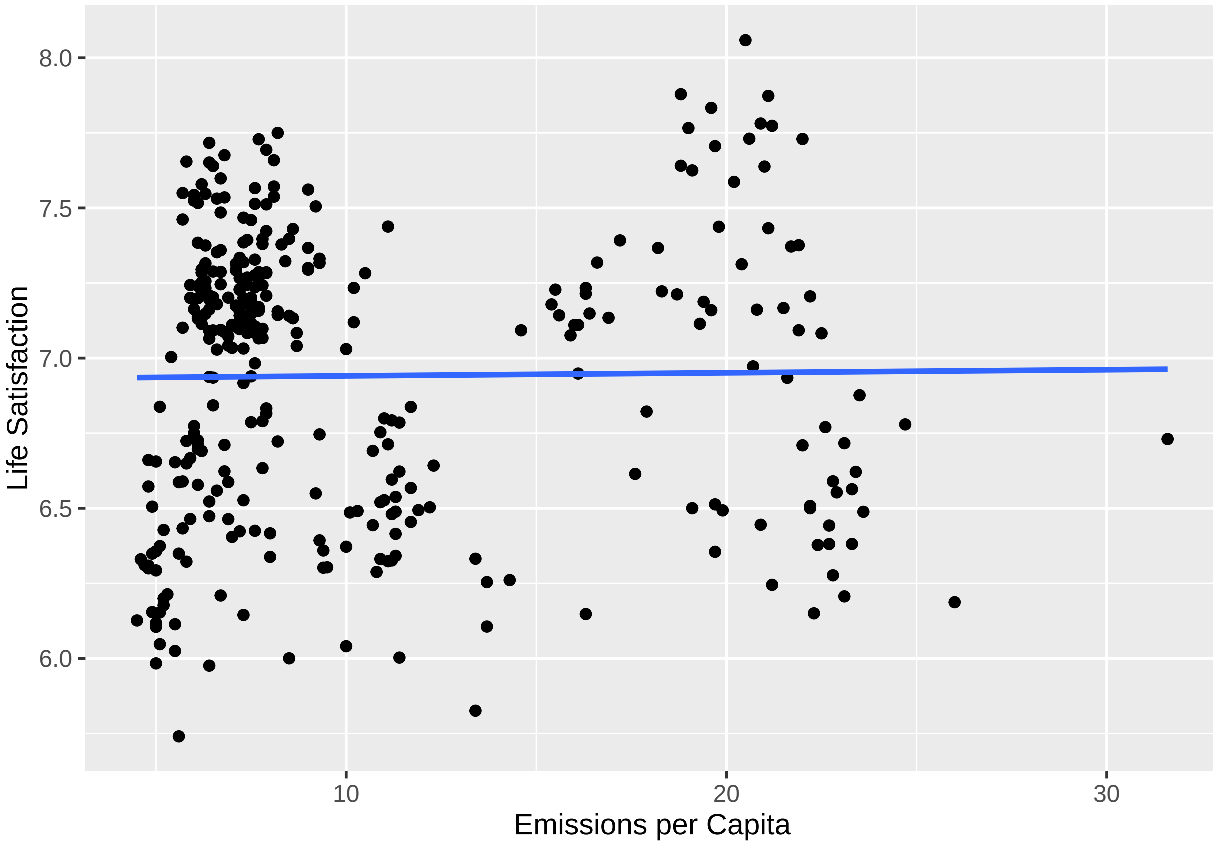
```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

##
## % Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvar
```

```
## % Date and time: Tue, Apr 19, 2016 - 6:55:26 AM
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}}lccccc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
## Statistic & \multicolumn{1}{c}{N} & \multicolumn{1}{c}{Mean} & \multicolumn{1}{c}{St. Dev.} & \multic
## \hline \\[-1.8ex]
## Group.2 & 332 & 2,001.280 & 6.641 & 1,990 & 2,012 \\
## Year & 332 & 2,001.280 & 6.641 & 1,990 & 2,012 \\
## Emissions & 332 & 10.502 & 5.793 & 4.500 & 31.600 \\
## Use & 332 & 169.748 & 52.282 & 79.300 & 337.000 \\
## satis & 332 & 6.941 & 0.472 & 5.740 & 8.059 \\
## environ & 332 & 1.775 & 0.170 & 1.212 & 2.281 \\
## age & 332 & 50.689 & 4.177 & 41.013 & 64.533 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
##
## % Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard
## % Date and time: Tue, Apr 19, 2016 - 6:55:26 AM
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}} c}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
## text \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

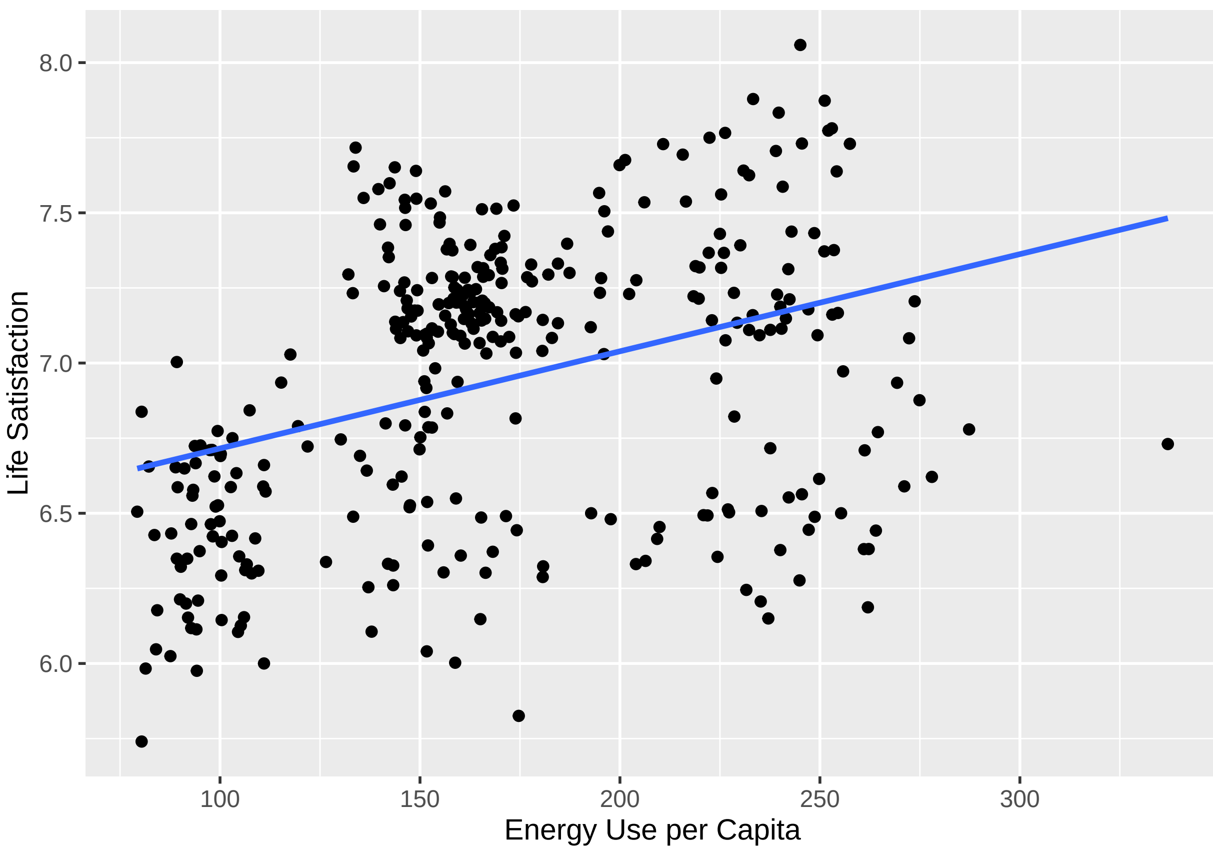


Again using the means of each state, we can create some scatterplots showing correlation between variables. Life satisfaction appears to be positively correlated with age, energy use, and concern about the environment, but does not show a strong correlation with emissions.

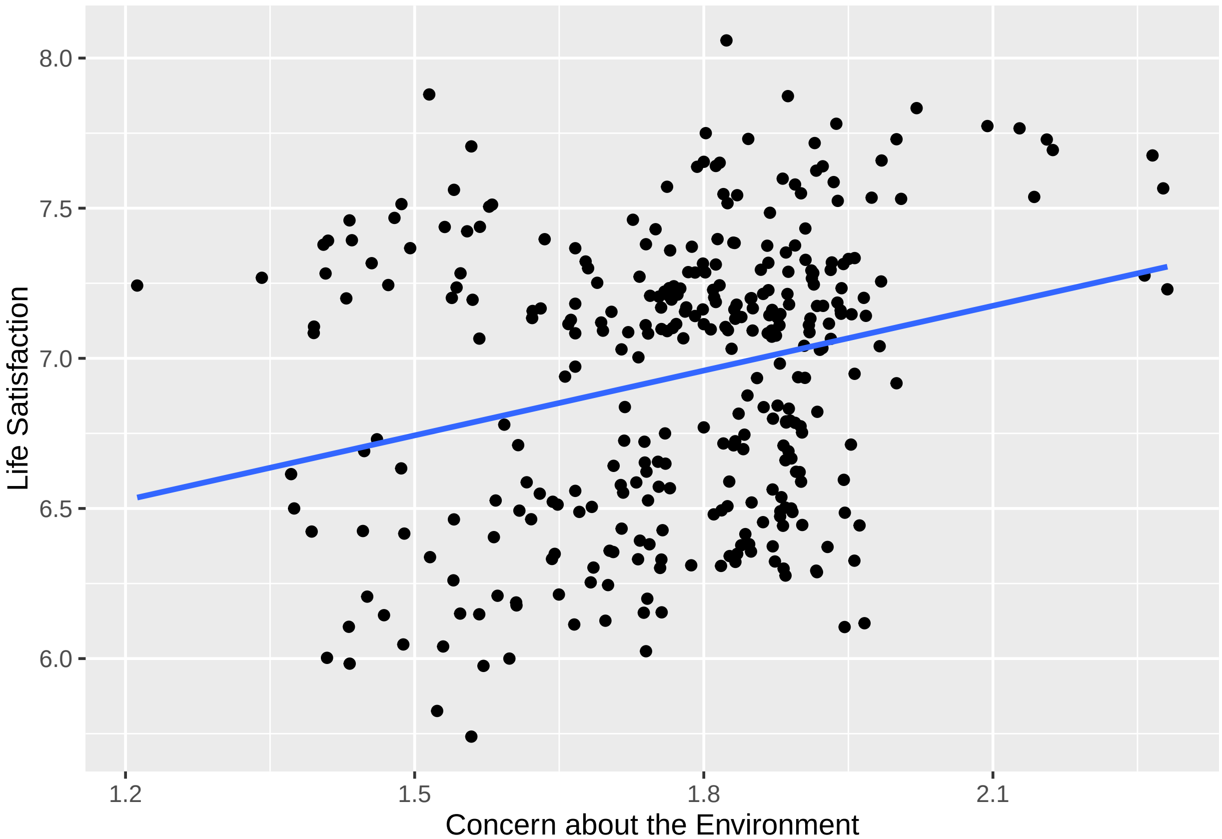

8.0
7.5
7.0
6.5
6.0
Life Satisfaction
40
45
50
55
60
65
Age



8.0
7.5
7.0
6.5
6.0
Life Satisfaction
10
20
30
Emissions per Capita



8.0
7.5
7.0
6.5
6.0
Life Satisfaction
100
150
200
250
300
Energy Use per Capita



Inferential Statistics

To run inferential statistics, we first create “pdataind” as panel data.

```
## serie LandCode  is NA and has been removed
```

```
##    Emissions          Use           environ          gender
##  Min.   : 4.50   Min.   : 79.3   Min.   :1.000   Min.   :1.000
##  1st Qu.: 6.50   1st Qu.:148.7   1st Qu.:1.000   1st Qu.:1.000
##  Median : 7.60   Median :164.0   Median :2.000   Median :2.000
##  Mean   :10.19   Mean   :172.3   Mean   :1.805   Mean   :1.522
##  3rd Qu.:13.70   3rd Qu.:210.8   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :31.60   Max.   :337.0   Max.   :3.000   Max.   :2.000
##       age
##  Min.   : 17.00
##  1st Qu.: 38.00
##  Median : 51.00
##  Mean   : 50.41
##  3rd Qu.: 64.00
##  Max.   :102.00
```

```
##      satis
##  Min.   : 0.000
##  1st Qu.: 6.000
```

```
##  Median : 7.000
##  Mean   : 6.948
##  3rd Qu.: 8.000
##  Max.   :10.000
```

Y1 represents life satisfaction, measured on a scale ranging from 0 to 10. X1 represents other variables of interest: Emissions, Energy Use, Concern for the Environment, Gender, and Age. We know that the data set is unbalanced, as we noticed earlier that we don't have information for all years from all the states.

Below, we tested various types of panel data models.

```
## Oneway (individual) effect Pooling Model
##
## Call:
## plm(formula = Y1 ~ X1, data = pdataind, model = "pooling")
##
## Unbalanced Panel: n=22285, T=1-23, N=140830
##
## Residuals :
##    Min. 1st Qu.  Median 3rd Qu.    Max.
##  -7.700  -1.110   0.342   1.110   4.020
##
## Coefficients :
##               Estimate  Std. Error  t-value Pr(>|t|)
## (Intercept)  6.16834228  0.03114077 198.0793   <2e-16 ***
## X1Emissions -0.09340835  0.00167109 -55.8968   <2e-16 ***
## X1Use        0.01104070  0.00018611  59.3239   <2e-16 ***
## X1environ    0.06628195  0.00769927   8.6089   <2e-16 ***
## X1gender    -0.01355593  0.00958703  -1.4140   0.1574
## X1age       -0.00534866  0.00027374 -19.5394   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    465260
## Residual Sum of Squares: 452070
## R-Squared:      0.028347
## Adj. R-Squared: 0.028346
## F-statistic: 821.692 on 5 and 140824 DF, p-value: < 2.22e-16
```

```
## Oneway (individual) effect Between Model
##
## Call:
## plm(formula = Y1 ~ X1, data = pdataind, model = "between")
##
## Unbalanced Panel: n=22285, T=1-23, N=140830
##
## Residuals :
##    Min. 1st Qu.  Median 3rd Qu.    Max.
## -7.3400 -0.5390  0.0669  0.7110  3.8700
##
## Coefficients :
##               Estimate  Std. Error  t-value  Pr(>|t|)
## (Intercept)  6.10499761  0.06783485  89.9980 < 2.2e-16 ***
## X1Emissions -0.09277665  0.00282401 -32.8528 < 2.2e-16 ***
```

```
## X1Use        0.01029657  0.00030811  33.4187 < 2.2e-16 ***
## X1environ    0.12037222  0.01892220   6.3614 2.038e-10 ***
## X1gender     0.02326979  0.02441911   0.9529    0.3406
## X1age       -0.00426204  0.00066807  -6.3796 1.810e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    31990
## Residual Sum of Squares: 30206
## R-Squared:      0.05576
## Adj. R-Squared: 0.055745
## F-statistic: 263.128 on 5 and 22279 DF, p-value: < 2.22e-16
```

```
## Oneway (individual) effect First-Difference Model
##
## Call:
## plm(formula = Y1 ~ X1, data = pdataind, model = "fd")
##
## Unbalanced Panel: n=22285, T=1-23, N=140830
##
## Residuals :
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -10.30000  -1.77000   0.00251   1.75000  10.20000
##
## Coefficients :
##                Estimate  Std. Error  t-value  Pr(>|t|)
## (intercept) -0.00558269  0.00728448  -0.7664   0.44345
## X1Emissions -0.07466170  0.03434678  -2.1738   0.02973 *
## X1Use        0.01404905  0.00324545   4.3288 1.500e-05 ***
## X1environ    0.05554390  0.00851167   6.5256 6.799e-11 ***
## X1gender    -0.02587608  0.01032822  -2.5054   0.01223 *
## X1age       -0.00586130  0.00030099 -19.4736 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    748530
## Residual Sum of Squares: 745660
## R-Squared:      0.0038401
## Adj. R-Squared: 0.0038399
## F-statistic: 91.391 on 5 and 118539 DF, p-value: < 2.22e-16
```

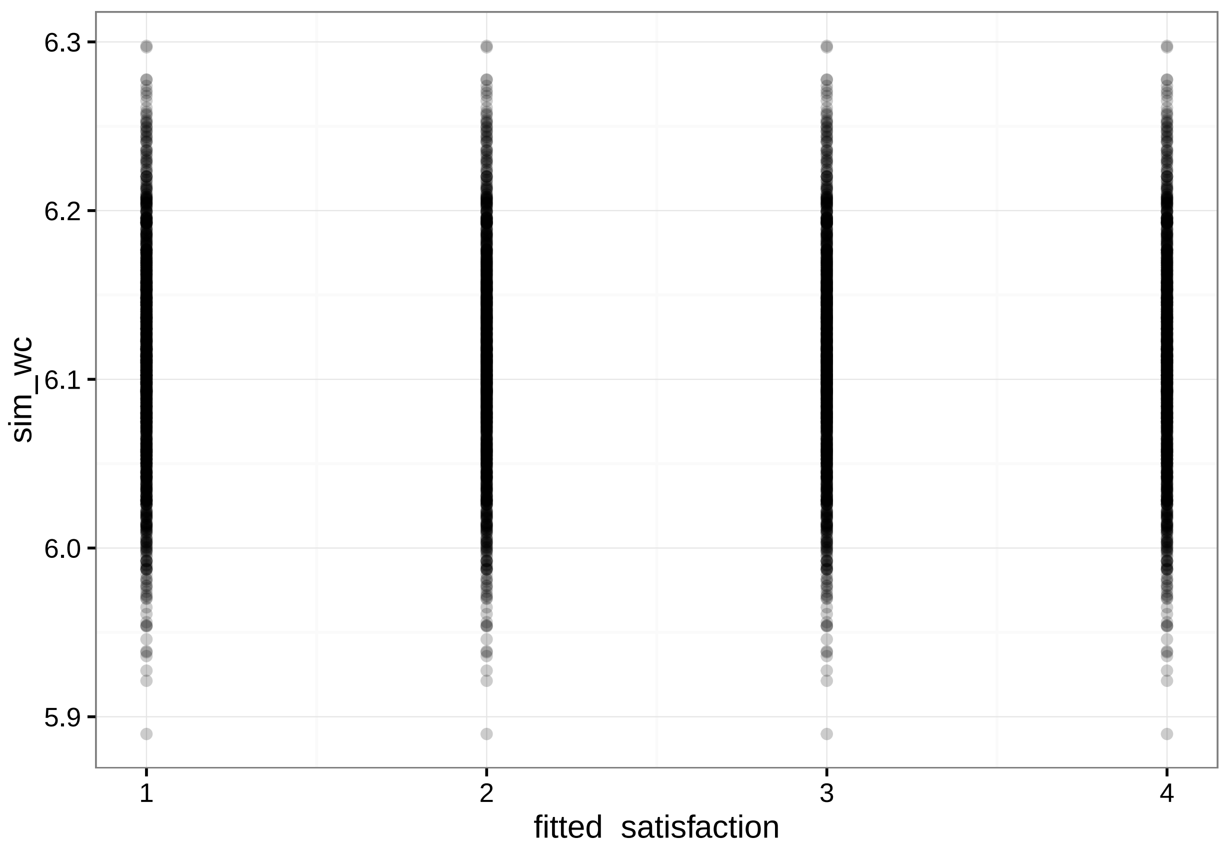
```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = Y1 ~ X1, data = pdataind, model = "within")
##
## Unbalanced Panel: n=22285, T=1-23, N=140830
##
## Residuals :
##    Min. 1st Qu.  Median 3rd Qu.    Max.
##  -7.880  -0.903   0.128   1.080   5.170
##
## Coefficients :
##                Estimate  Std. Error  t-value  Pr(>|t|)
```

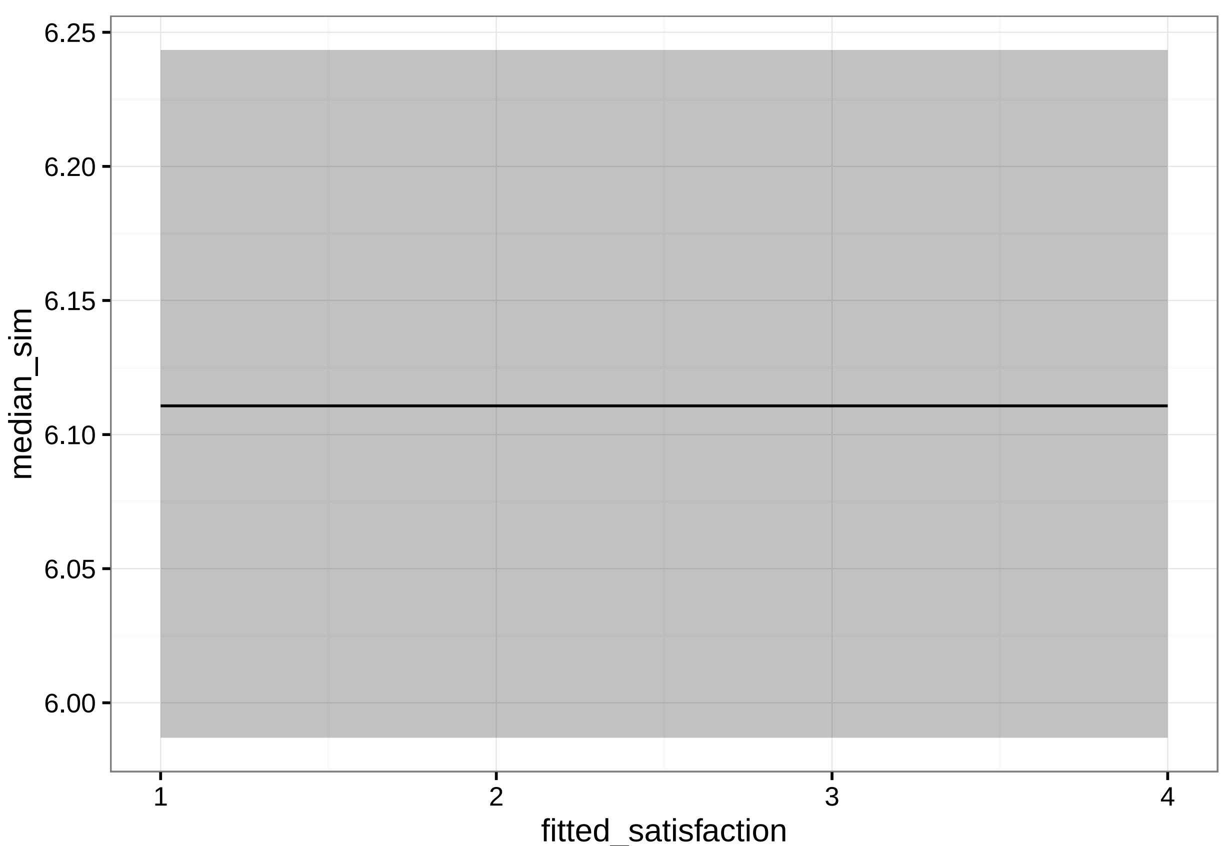
```
## X1Emissions -0.08841148  0.01626967  -5.4341 5.517e-08 ***
## X1Use        0.01276929  0.00152923   8.3501 < 2.2e-16 ***
## X1environ    0.05866414  0.00851204   6.8919 5.533e-12 ***
## X1gender    -0.01951174  0.01032560  -1.8896   0.05881 .
## X1age       -0.00587612  0.00030079 -19.5359 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    374420
## Residual Sum of Squares: 372810
## R-Squared:      0.0043051
## Adj. R-Squared: 0.0036237
## F-statistic: 102.507 on 5 and 118540 DF, p-value: < 2.22e-16
```

```
##
##  F test for individual effects
##
## data:  Y1 ~ X1
## F = 2.6248, df1 = 118540, df2 = 22279, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

```
##
## <table style="text-align:center"><caption><strong>Regression Estimates of Life Satisfaction</strong>
## <tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td></td><td colspan="4" style="border-bottom: 1px solid black"></td></tr>
## <tr><td style="text-align:left"></td><td colspan="4">Y1</td></tr>
## <tr><td style="text-align:left"></td><td>(1)</td><td>(2)</td><td>(3)</td><td>(4)</td></tr>
## <tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td style="text-align:left"></td><td>(0.03)</td><td>(0.07)</td><td></td><td></td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">X1Emissions</td><td>-0.09<sup>***</sup></td><td>-0.09<sup>***</sup><
## <tr><td style="text-align:left"></td><td>(0.002)</td><td>(0.003)</td><td>(0.03)</td><td>(0.02)</td><
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">X1Use</td><td>0.01<sup>***</sup></td><td>0.01<sup>***</sup></td><td>
## <tr><td style="text-align:left"></td><td>(0.0002)</td><td>(0.0003)</td><td>(0.003)</td><td>(0.002)</
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">X1environ</td><td>0.07<sup>***</sup></td><td>0.12<sup>***</sup></td>
## <tr><td style="text-align:left"></td><td>(0.01)</td><td>(0.02)</td><td>(0.01)</td><td>(0.01)</td></t
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">X1gender</td><td>-0.01</td><td>0.02</td><td>-0.03<sup>**</sup></td><
## <tr><td style="text-align:left"></td><td>(0.01)</td><td>(0.02)</td><td>(0.01)</td><td>(0.01)</td></t
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">X1age</td><td>-0.01<sup>***</sup></td><td>-0.004<sup>***</sup></td><
## <tr><td style="text-align:left"></td><td>(0.0003)</td><td>(0.001)</td><td>(0.0003)</td><td>(0.0003)<
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">Constant</td><td>6.17<sup>***</sup></td><td>6.10<sup>***</sup></td><
## <tr><td style="text-align:left"></td><td>(0.03)</td><td>(0.07)</td><td></td><td></td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td style="text-align:left">R<sup>2</sup></td><td>0.03</td><td>0.06</td><td>0.004</td><td>0.004<
## <tr><td style="text-align:left">Adjusted R<sup>2</sup></td><td>0.03</td><td>0.06</td><td>0.004</td><
## <tr><td style="text-align:left">F Statistic</td><td>821.69<sup>***</sup> (df = 5; 140824)</td><td>26
## <tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## </table>
```

```
## [1]  0 10

##   X.Intercept. X1Emissions      X1Use X1environ    X1gender        X1age
## 1     6.018885 -0.09586723 0.01087314 0.1192140  0.02375484 -0.003872026
## 2     6.151989 -0.09753577 0.01072065 0.1077767  0.01648174 -0.004888117
## 3     6.090931 -0.09324284 0.01053594 0.1247382 -0.01360983 -0.003928525

## [1] 1000

## [1] 4000

## [1] "X.Intercept."        "X1Emissions"         "X1Use"
## [4] "X1environ"           "X1gender"            "X1age"
## [7] "fitted_satisfaction"

## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.
```


6.3
6.2
6.1
6.0
5.9
sim_wc
1
2
3
4
fitted_satisfaction


6.25
6.20
6.15
6.10
6.05
6.00
median_sim
1
2
3
4
fitted_satisfaction

Multilevel analysis

First step of the MRC