Code

Drivers of social equity in renewable energy at the municipal level: The case of local Japanese energy policy and preferences

Multicolinearity tests

1. Introduction

This document describes and records the code used in our paper, “Drivers of social equity in renewable energy at the municipal level: The case of local Japanese energy policy and preferences.” Below, we outline the questions we aim to answer, approach, and R code used to answer our questions, one by one.

1.1 Load Packages

Hide

```
# Load packages
require(tidyverse)
require(readxl)
require(Amelia)
require(distributions3)
require(gtools)
require(car)
require(scales)
require(lmtest)
set.seed(12345)
```

1.2 Import Data

First, we import data. The excel file *“population_data.xlsx”* contains data for all 1741 Japanese municipalities, while the excel file *“sampling_frame.xlsx”* contains information on each town in the sampling frame. It also contains responses for the actual towns which responded to our survey. Finally, *“regression_table.xlsx”* contains the data for the 47 final towns in our analysis, which we use to produce regression results.

Hide

```
# Import population data
pop = read_xlsx("population_data.xlsx")

# Import sampling frame
sample = read_xlsx("sampling_frame.xlsx")

# Import final set of cases
dat = read_xlsx("regression_table.xlsx")
```

2. Descriptive Statistics

Next, we calculate descriptive statistics for each key concept in our dataset. These are represented in Appendix A in our manuscript.

measure <chr>	**Median** <dbl>	**Average** <dbl>	**Std. Dev.** <dbl>	
Age_median_2010	4.930000e+01	4.877872e+01	5.059899e+00	3.83000
Biomass Capacity Factor	9.080000e-01	9.080000e-01	0.000000e+00	9.0800(
Biomass GWh	0.000000e+00	4.589166e+01	2.483771e+02	0.00000
Burden (100 scale)	5.511173e+01	5.364612e+01	2.623420e+00	4.47675
Burden (non-weighted)	5.515805e+01	5.332092e+01	4.431961e+00	2.78987
Climate_change_countermeasures	4.000000e+00	3.744681e+00	1.092824e+00	1.00000
CO2 Offset per capita (Total)	4.824837e+05	3.433297e+07	9.577138e+07	0.00000
Com_Tax_average	3.000000e+00	2.882979e+00	1.277790e+00	0.00000
Community_Development	3.000000e+00	2.893617e+00	1.237705e+00	0.00000
Community_Tax_Base	3.000000e+00	2.872340e+00	1.568952e+00	0.00000

1-10 of 35 rows | 1-5 of 7 columns Previous **1** 2 3 4 Next

3. Representativeness

Next, we would like to verify whether the towns which responded to our survey are representative of the population in terms of the traits that we care about. To do that, first, we must deal with missing data at the population level. We use multiple imputation via the **amelia** package to fill in data points for variables missing data for less than 5 percent of cases. We do this on the entire dataset, incorporating latent trends in this data to more accurately fill in missing data.

First, we calculate the logical bounds of our variables. These are the range of values that we allow missing data points to take on in the multiple imputation process. This helps avoid situations where we might fill in missing data for a scale from 0-100 with, say, 120, and impossible value.

3.1 Calculate Logical Bounds for Multiple Imputation

Hide

```
bounds = pop %>%
  # pivot to tidy, long form so we can calculate summary statistic easily
  pivot_longer(cols = -c(Code, Town_Japanese, Prefecture_Town_Japanese,
                         Prefecture_English, Prefecture_Town_English, Town_English),
               names_to = "measure",
               values_to = "value") %>%
  group_by(measure) %>%
  summarize(
    # Calculate percentage of missing data points in that column
    missing_percent = sum(if_else(is.na(value), 1, 0)) / n(),
    # Calculate minimum value for that column
    min = min(value, na.rm = TRUE),
    # calculate maximum value for that column
    max = max(value, na.rm = TRUE)) %>%
  # filter this list to just variables missing less than 5%
  filter(missing_percent < 0.05) %>%
  # create an id number for each measure
  mutate(column.number = 1:n()) %>% ungroup
```

3.2 Multiple Imputation with Amelia

Hide

```
# How much was missing from each?
summary(imp)
```

```


Amelia output with 5 imputed datasets.
Return code:  1
Message:  Normal EM convergence.

Chain Lengths:
--------------
Imputation 1:  8
Imputation 2:  4
Imputation 3:  6
Imputation 4:  6
Imputation 5:  5

Rows after Listwise Deletion:  1701
Rows after Imputation:  1741
Patterns of missingness in the data:  5

Fraction Missing for original variables:
-----------------------------------------
```

	Fraction Missing <dbl>
Code	0.000000000
Age_median_2010	0.006892590
Area_2010	0.000000000
Area_industrial_2010	0.000000000
Area_inhabitable_2010	0.001723148
Area_residential_2010	0.000000000
BiomasskW	0.000000000
BP	0.000000000
Crime_Rate_2008	0.000000000
Diversity	0.000000000


```
NA
```

3.3 Calculate Difference of Means across Imputations

Hide

```
# Let's use one-sample t-tests to compare whether
# our average sample average is statistically significantly different
# from the population average for key demographic and technological variables
# that we could collect at both levels.
# This allows us to tell how representative the survey sample data used later in the paper is.

# Calculate test statistic (difference of means) for each imputation
imp.stat = function(data){
  data %>%
    summarize(
      # Calculate test statistic
      SolarkW_statistic = t.test(
        dat$SolarkW,
        mu = mean(data$SolarkW))$statistic,
      WindkW_statistic = t.test(
        dat$WindkW,
        mu = mean(data$WindkW))$statistic,
      BiomasskW_statistic = t.test(
        dat$BiomasskW,
        mu = mean(data$BiomasskW))$statistic,
      Population_2010_statistic = t.test(
        dat$Population_2010,
        mu = mean(data$Population_2010))$statistic,
      Income_taxable_per_capita_2010_statistic = t.test(
        dat$Income_taxable_per_capita_2010,
        mu = mean(data$Income_taxable_per_capita_2010))$statistic,
      Unemployment_2010_statistic = t.test(
        dat$Unemployment_2010,
        mu = mean(data$Unemployment_2010))$statistic,
      Age_median_2010_statistic = t.test(
        dat$Age_median_2010,
        mu = mean(data$Age_median_2010))$statistic,
      Ratio_Revs_Exp_2010_statistic = t.test(
        dat$Ratio_Revs_Exp_2010,
        mu = mean(data$Ratio_Revs_Exp_2010))$statistic,
      Local_taxes_per_capita_2010_statistic = t.test(
        dat$Local_taxes_per_capita_2010,
        mu = mean(data$Local_taxes_per_capita_2010))$statistic,
      # Calculate standard error
      SolarkW_se = t.test(
        dat$SolarkW,
        mu = mean(data$SolarkW))$stderr,
      WindkW_se = t.test(
        dat$WindkW,
        mu = mean(data$WindkW))$stderr,
      BiomasskW_se = t.test(
        dat$BiomasskW,
        mu = mean(data$BiomasskW))$stderr,
      Population_2010_se = t.test(
        dat$Population_2010,
        mu = mean(data$Population_2010))$stderr,
      Income_taxable_per_capita_2010_se = t.test(
        dat$Income_taxable_per_capita_2010,
```

```
        mu = mean(data$Income_taxable_per_capita_2010))$stderr,
      Unemployment_2010_se = t.test(
        dat$Unemployment_2010,
        mu = mean(data$Unemployment_2010))$stderr,
      Age_median_2010_se = t.test(
        dat$Age_median_2010,
        mu = mean(data$Age_median_2010))$stderr,
      Ratio_Revs_Exp_2010_se = t.test(
        dat$Ratio_Revs_Exp_2010,
        mu = mean(data$Ratio_Revs_Exp_2010))$stderr,
      Local_taxes_per_capita_2010_se = t.test(
        dat$Local_taxes_per_capita_2010,
        mu = mean(data$Local_taxes_per_capita_2010))$stderr) %>%
      return()
}

# Collect difference of means and standard error for each imputed dataset
imp.result = imp$imputations %>%
  lapply(imp.stat) %>%
  bind_rows(.id = "imp") %>%
  # Pivot long ways
  pivot_longer(cols = -imp, names_to = "measure", values_to = "value") %>%
  # Classify each data point as a statistic or standard error
  mutate(type = str_extract(measure, "_se|_statistic") %>% str_remove("_"),
         measure = str_remove(measure, "_se|_statistic")) %>%
  # Pivot wider into a format mi.meld can receive to average results
  pivot_wider(id_cols = measure,
              names_from = c(type, imp),
              values_from = value)


# To prepare for calculating p.values from averaged test statistics,
# we are going to make a normal t-distribution
require(distributions3)
# calculate the distribution with 46 degrees of freedom
t46 = StudentsT(df = 46)

# Use Rubin's Rules to average the statistic,
# and merge the standard error within imputations and between imputations
mi.meld(q = imp.result %>%
          # grab means
          select(statistic_imp1, statistic_imp2,statistic_imp3,statistic_imp4, statistic_imp5),
        se = imp.result %>%
          # grab standard deviations
          select(se_imp1,se_imp2,se_imp3,se_imp4,se_imp5),
        # Each column is a set of results from an imputed dataframe
        byrow = FALSE) %>%
  # bind results together
  bind_rows() %>%
  # Add variable names back in
  mutate(measure = imp.result$measure)  %>%
  # reorder and rename columns
  select(measure, stat = q.mi, se = se.mi) %>%
  # Now calculate
```

```
  mutate(p.value = 1 - cdf(t46, abs(stat)) + cdf(t46, -abs(stat))) %>%
  # Add asterisks
  mutate(significance = stars.pval(p.value))
```

Hide

```
remove(imp.stat, imp.result, t46, order)
```

```
object 'order' not found
```

This suggests that our sample differs from the population in a statistically significant way for the installed capacity of Wind and Biomass, and the median age. Fortunately, wind and biomass make up a tiny portion of renewables in Japan overall, and most of cities' equity scores are determined by solar anyways, because it constitutes such a large share of Japanese renewables. With these limitations in mind, we proceed with our analysis.

4. Sample Survey Descriptive Statistics

Next, we calculate descriptive statistics on results from our survey sample.

measure <chr>	**mean** <dbl>	**sd** <dbl>
Climate_change_countermeasures	3.744681	1.0928241
Community_Development	2.893617	1.2377055
Community_Tax_Base	2.872340	1.5689518
Disaster_Resilience	2.617021	1.4378908
Electricity_Prices	2.744681	1.2591938
Employment	2.638298	1.3092569
Environmental_conservation	4.468085	0.8035536
Fair_Labor_Conditions	2.191489	1.2961196
Pollution_countermeasures	3.468085	1.5301472
Social_Equity	2.638298	1.1689001

1-10 of 10 rows

5. Modeling

Next, we model our variables. For each model, we display the summary table, and then run diagnostics. We display multicolinearity tests, namely results from the variance inflation factor (VIF) test. The gold standard for VIF is below 2.5, while extremely problematic values tend to be over 10. None of our models have VIFs over 3; most are at or under 2.5. As a result, we proceed with confidence that multicolinearity is not a problem in our models.

5.1 Model 1: Effect of Technology on Equity

Hide

```
summary(equity)
```

```
Call:
lm(formula = `Equity (y)` ~ `Solar GWh` + `Wind GWh` + `Biomass GWh` +
    `PV_output_2018 (kWh/kWp/annum)` + Windspeed_2018, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3609 -0.4935 -0.1035  0.2239  1.9789

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                       1.159e-16  1.238e-01   0.000 1.000000
`Solar GWh`                       8.399e-01  2.120e-01   3.961 0.000291 ***
`Wind GWh`                       -3.843e-01  2.112e-01  -1.819 0.076145 .
`Biomass GWh`                    -1.346e-01  1.314e-01  -1.025 0.311510
`PV_output_2018 (kWh/kWp/annum)` -4.346e-02  1.364e-01  -0.319 0.751719
Windspeed_2018                   -4.064e-02  1.365e-01  -0.298 0.767410
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.849 on 41 degrees of freedom
Multiple R-squared:  0.3575,    Adjusted R-squared:  0.2792
F-statistic: 4.563 on 5 and 41 DF,  p-value: 0.002123
```

Hide

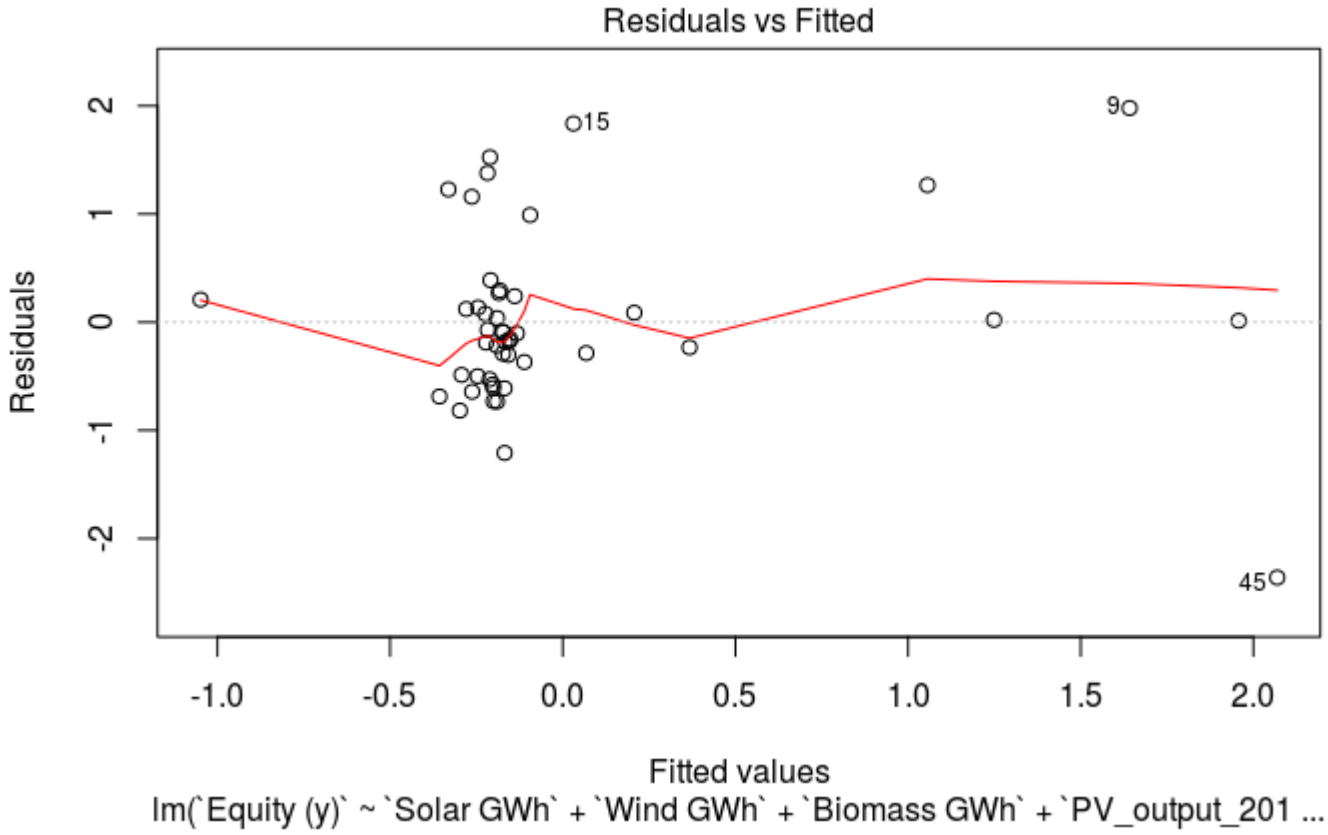
```
# What is the average VIF for this model?
mean(vif(equity))
```

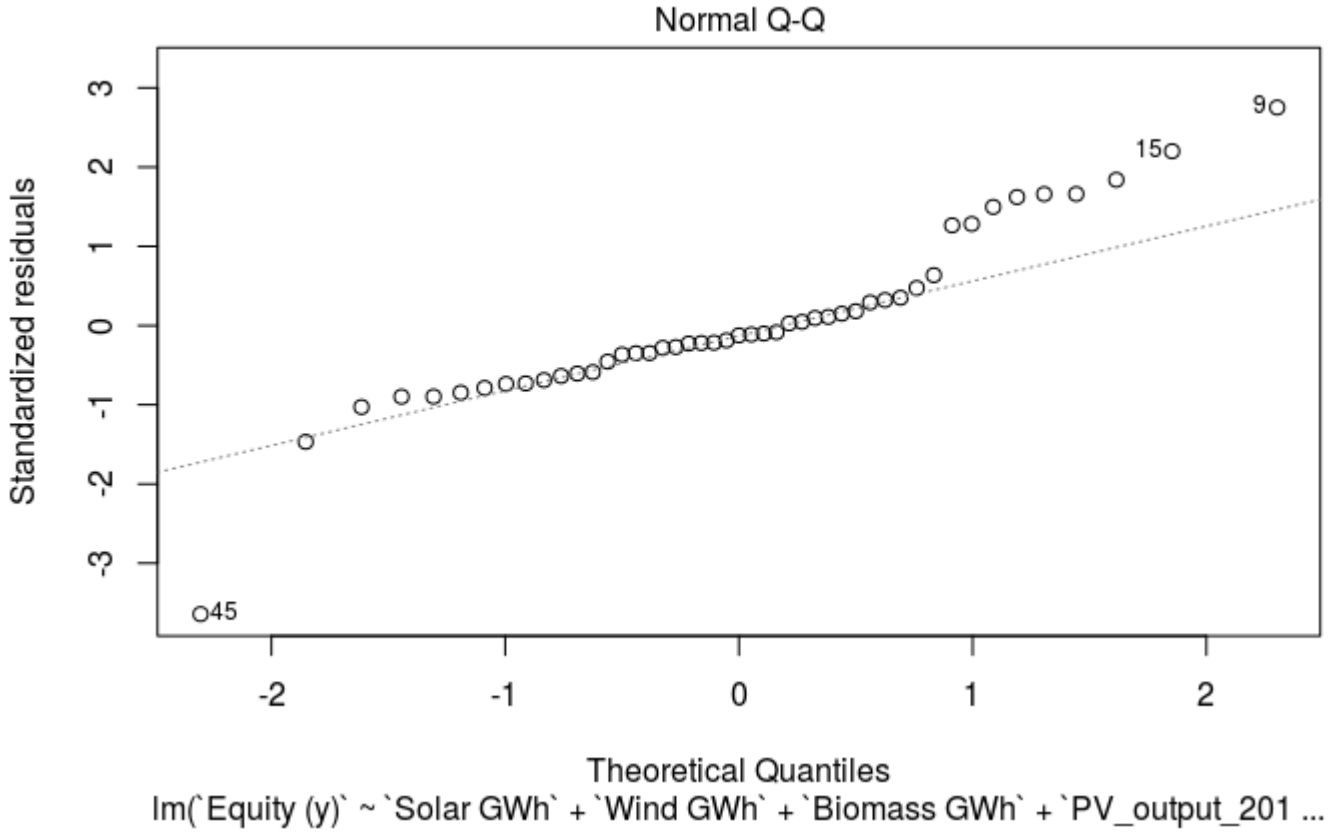
Hide

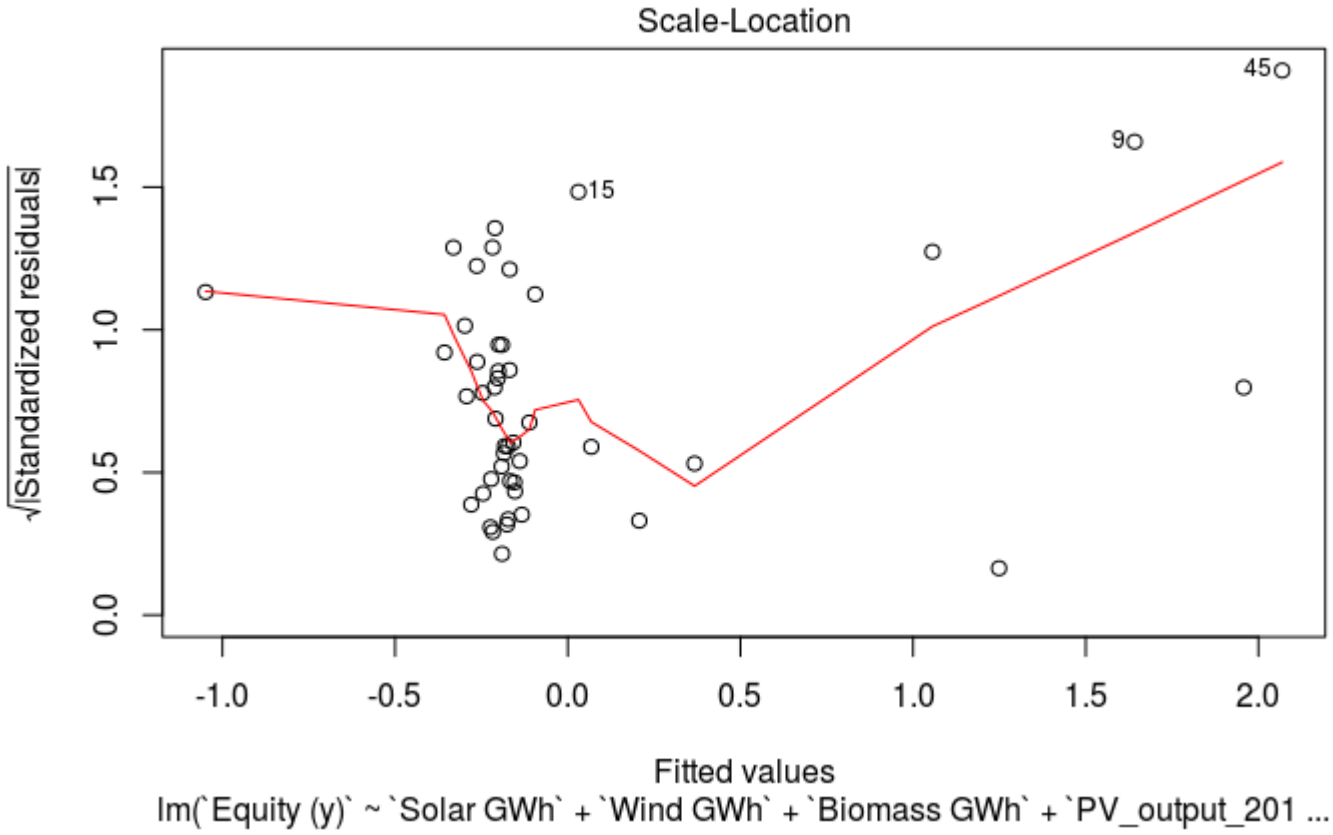
```
# Report individual VIF results
vif(equity) # variance inflation factors
```

```
                    `Solar GWh`                      `Wind GWh`                     `Biomass GW
h`
                       2.868426                        2.847479                          1.1013
03
`PV_output_2018 (kWh/kWp/annum)`                  Windspeed_2018
                       1.187860                        1.188807
```

Next, we evaluate heteroskedasticity. The plots below indicate some heteroskedasticity, but multiple regression tends to be quite robuts against it. As a result, we also plot residuals to evaluate whether they are normally distributed or are experiencing autocorrelation of some sort. We see little evidence of this.


Residuals vs Fitted
Residuals
Fitted values
lm(`Equity (y)` ~ `Solar GWh` + `Wind GWh` + `Biomass GWh` + `PV_output_201 ...
15
9
45
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0
2
1
0
-1
-2

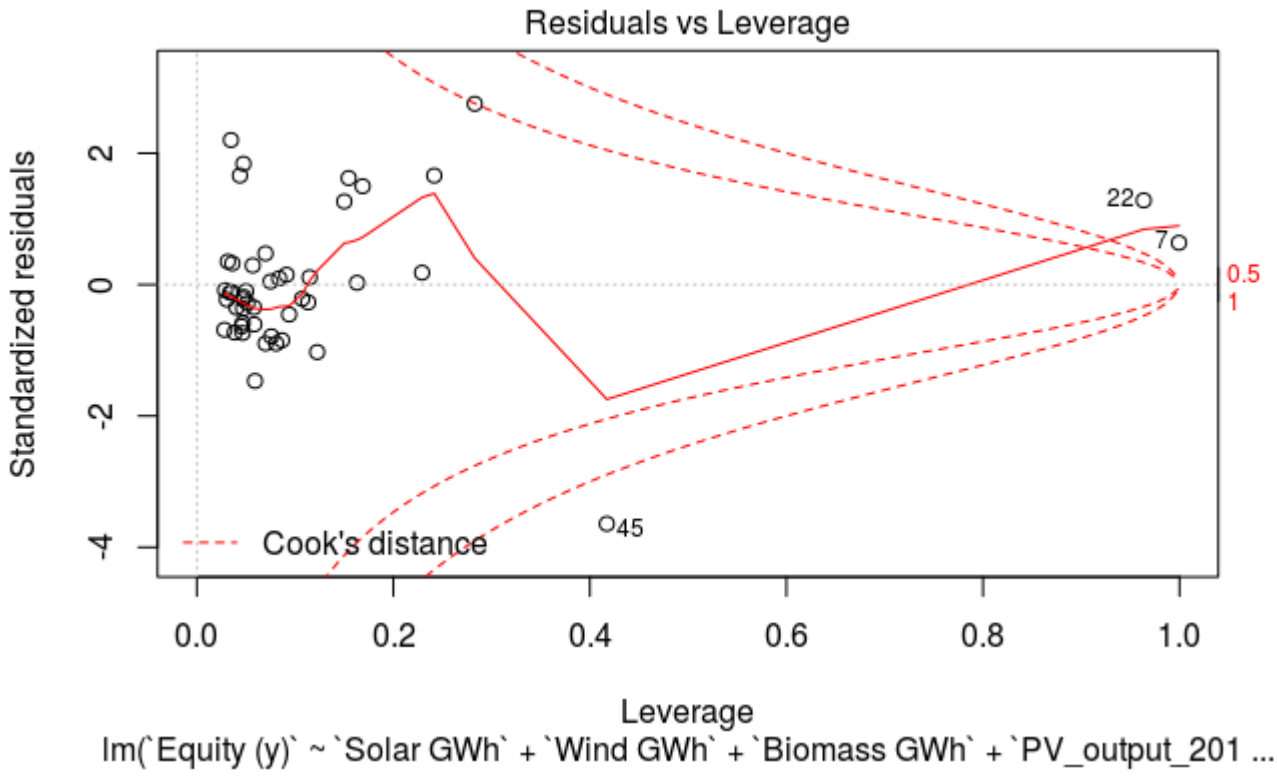

Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(`Equity (y)` ~ `Solar GWh` + `Wind GWh` + `Biomass GWh` + `PV_output_201 ...
9
15
45
-2
-1
0
1
2
3
2
1
0
-1
-2
-3

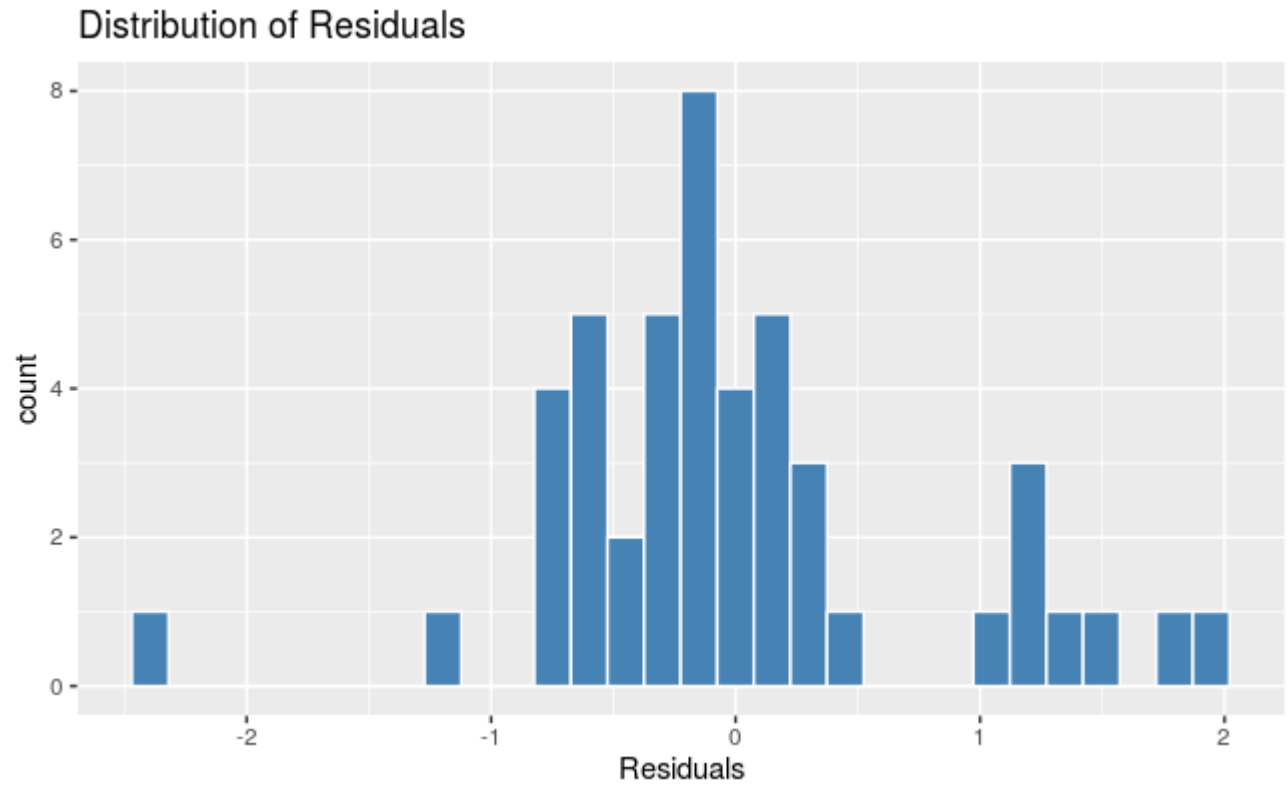


Hide

```
# Diagnostic Plots
plot(equity)
```

```
NaNs producedNaNs produced
```





5.2 Models 2-3: Interaction Effects on Equity

Hide

```
summary(equity2)
```

```
Call:
lm(formula = `Equity (y)` ~ `Solar GWh` + `Wind GWh` + `Biomass GWh` +
    `PV_output_2018 (kWh/kWp/annum)` + Windspeed_2018 + `Solar GWh` *
    `PV_output_2018 (kWh/kWp/annum)`, data = .)

Residuals:
     Min       1Q   Median       3Q      Max
-1.15750 -0.51469 -0.08444  0.30243  1.67208

Coefficients:
                                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                                 0.18908    0.13506   1.400   0.1692
`Solar GWh`                                 1.59932    0.34435   4.644 3.65e-05 ***
`Wind GWh`                                 -0.55212    0.20641  -2.675   0.0108 *
`Biomass GWh`                              -0.11057    0.12270  -0.901   0.3729
`PV_output_2018 (kWh/kWp/annum)`           -0.61667    0.24796  -2.487   0.0172 *
Windspeed_2018                              0.02393    0.12939   0.185   0.8542
`Solar GWh`:`PV_output_2018 (kWh/kWp/annum)` -1.91238    0.71032  -2.692   0.0103 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7909 on 40 degrees of freedom
Multiple R-squared:  0.4561,    Adjusted R-squared:  0.3745
F-statistic:  5.59 on 6 and 40 DF,  p-value: 0.0002765
```

Hide

```
# Calculate average variance inflation factor
mean(vif(equity2))
```

```
[1] 4.423633
```

Hide

```
summary(equity3)
```

```
Call:
lm(formula = `Equity (y)` ~ `Solar GWh` + `Wind GWh` + `Biomass GWh` +
    `PV_output_2018 (kWh/kWp/annum)` + Windspeed_2018 + `Wind GWh` *
    Windspeed_2018, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3506 -0.4673 -0.1563  0.2198  1.9780

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -0.58597    1.72080  -0.341 0.735243
`Solar GWh`                      0.84738    0.21546   3.933 0.000325 ***
`Wind GWh`                      -4.24071   11.29677  -0.375 0.709353
`Biomass GWh`                   -0.12994    0.13351  -0.973 0.336294
`PV_output_2018 (kWh/kWp/annum)` -0.05398    0.14133  -0.382 0.704548
Windspeed_2018                   1.61296    4.84513   0.333 0.740945
`Wind GWh`:Windspeed_2018       11.03446   32.31839   0.341 0.734567
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8583 on 40 degrees of freedom
Multiple R-squared:  0.3594,    Adjusted R-squared:  0.2633
F-statistic:  3.74 on 6 and 40 DF,  p-value: 0.004786
```

Hide

```
mean(vif(equity3))
```

```
[1] 3080.811
```

5.3 Transformations of Model 1

Hide

```
# These transformations reduces each reduce heteroskedasticity.
# Reciprocal transformation appears to improve it.
dat %>%
  mutate(
    original = `Equity (y)`,
    reciprocal = (1 / `Equity (y)`),
    sqrt = `Equity (y)` %>% sqrt,
    log = `Equity (y)` %>% log) %>%
  select(original, reciprocal, sqrt, log)
```

```
NaNs producedNaNs produced
```

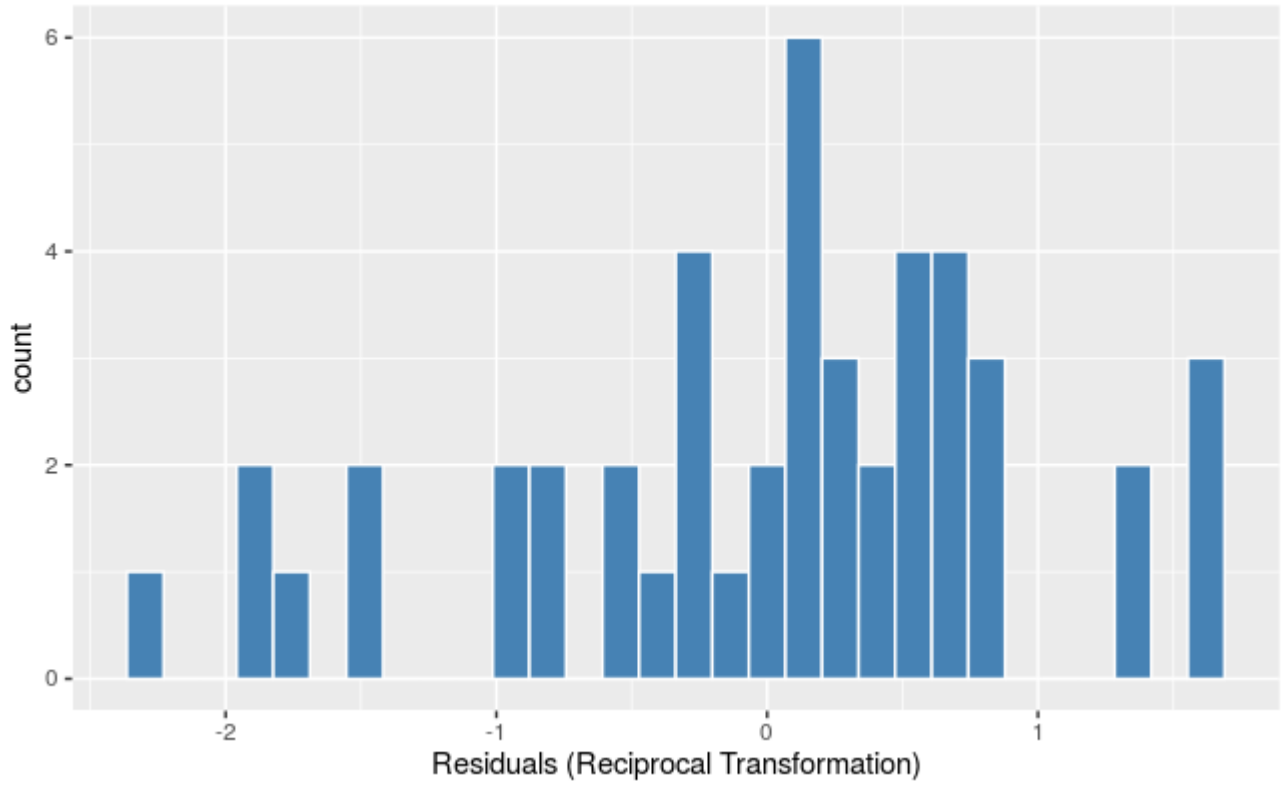
original <dbl>	reciprocal <dbl>	sqrt <dbl>	log <dbl>
2.4856989	0.40230135	1.5766099	0.91055386
-0.7758951	-1.28883402	NaN	NaN
-0.9652083	-1.03604579	NaN	NaN
-1.1266596	-0.88757954	NaN	NaN
-0.9628918	-1.03853825	NaN	NaN
1.6496939	0.60617306	1.2844041	0.50058976
14.2962502	0.06994841	3.7810382	2.65999728
10.4108675	0.09605348	3.2265876	2.34285021
23.4554843	0.04263395	4.8430862	3.15510434
-1.7682590	-0.56552802	NaN	NaN

1-10 of 47 rows Previous **1** 2 3 4 5 Next

Hide

```
summary(equity_transformed)
```

```
Call:
lm(formula = `Equity (y)` ~ `Solar GWh` + `Wind GWh` + `Biomass GWh` + 
    `PV_output_2018 (kWh/kWp/annum)` + Windspeed_2018, data = .)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.2319 -0.4691  0.1182  0.5734  1.6860 

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     3.334e-16  1.458e-01   0.000    1.000
`Solar GWh`                     1.618e-01  2.495e-01   0.648    0.520
`Wind GWh`                     -1.270e-01  2.486e-01  -0.511    0.612
`Biomass GWh`                  -1.545e-01  1.546e-01  -0.999    0.324
`PV_output_2018 (kWh/kWp/annum)` -2.470e-01  1.606e-01  -1.538    0.132
Windspeed_2018                 -1.446e-01  1.606e-01  -0.900    0.373

Residual standard error: 0.9992 on 41 degrees of freedom
Multiple R-squared:   0.11, Adjusted R-squared:  0.001516 
F-statistic: 1.014 on 5 and 41 DF,  p-value: 0.4219
```



5.4.1 Effect of Community Preferences on Equity Outcomes

Hide

```
summary(equity_preference)
```

```
Call:
lm(formula = `Equity (y)` ~ Env_Clim_average + Pollution_countermeasures +
    Electricity_Prices + Com_Tax_average + Employment, data = .)

Residuals:
     Min       1Q   Median       3Q      Max
-1.36246 -0.64535 -0.09826  0.29244  3.14181

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)               -3.576e-17  1.376e-01   0.000   1.0000
Env_Clim_average           8.199e-02  1.469e-01   0.558   0.5799
Pollution_countermeasures  9.389e-02  1.567e-01   0.599   0.5524
Electricity_Prices         2.657e-01  1.509e-01   1.761   0.0856 .
Com_Tax_average            5.129e-01  1.942e-01   2.640   0.0117 *
Employment                -4.227e-01  2.098e-01  -2.015   0.0505 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9432 on 41 degrees of freedom
Multiple R-squared:  0.2071,    Adjusted R-squared:  0.1104
F-statistic: 2.142 on 5 and 41 DF,  p-value: 0.07959
```

Hide

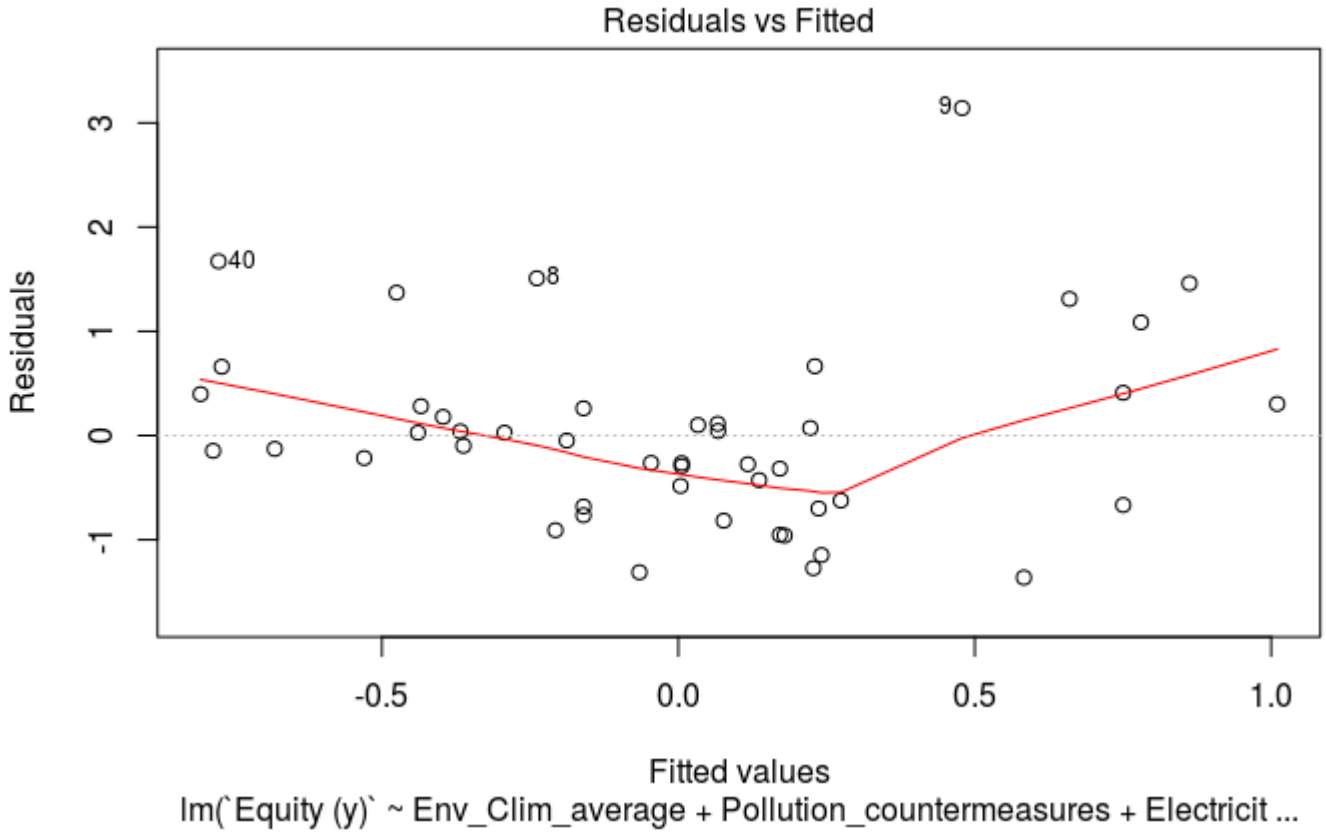
```
# Great VIF results - not multicolinearity
mean(vif(equity_preference))
```

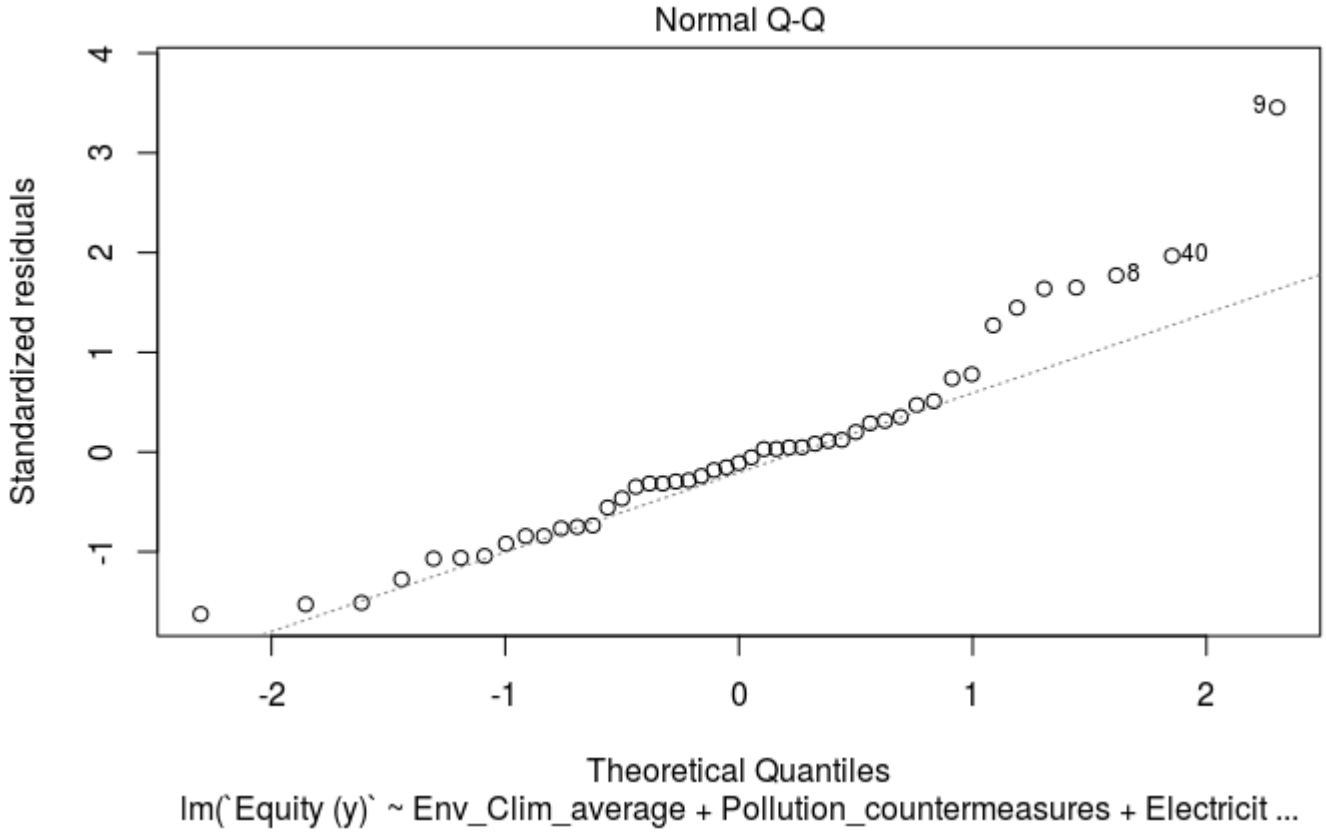
```
[1] 1.55808
```

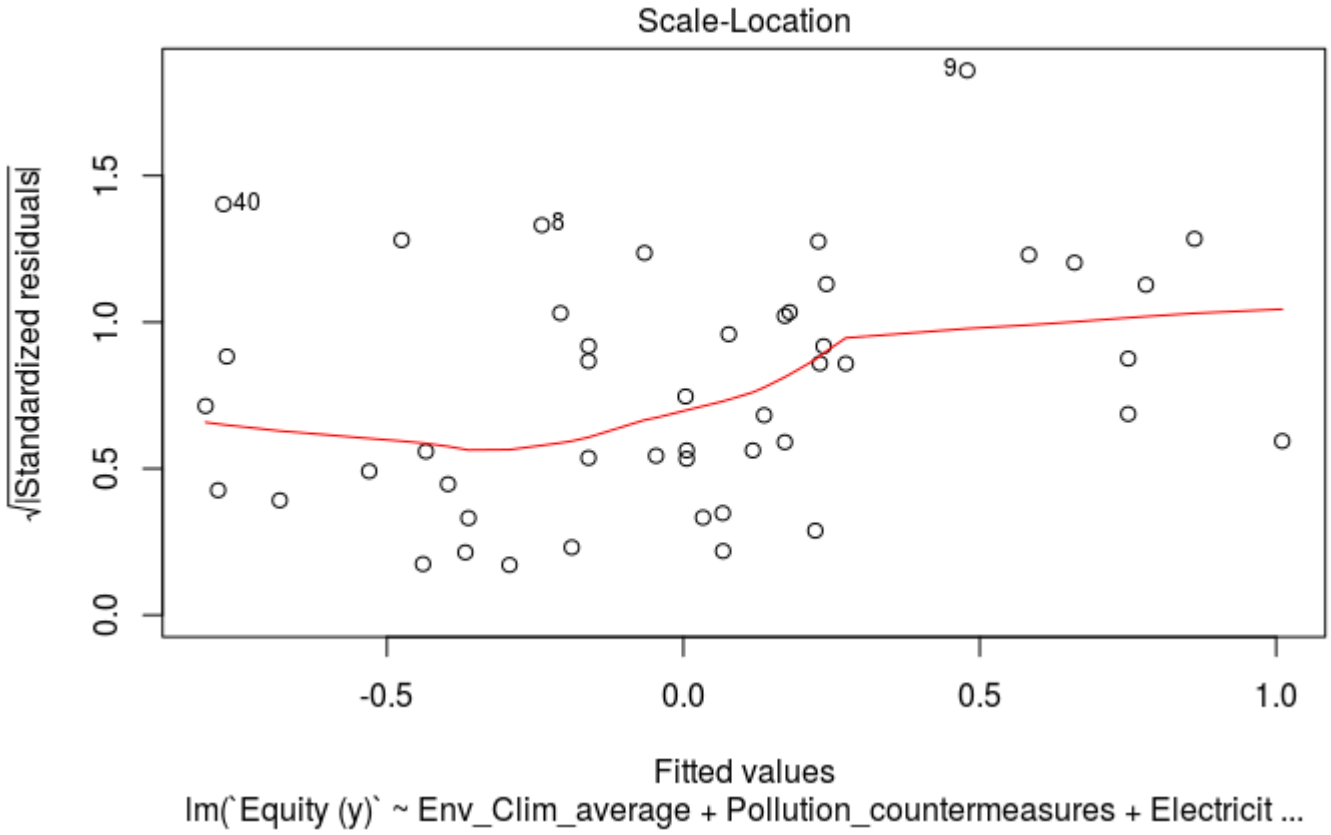
Hide

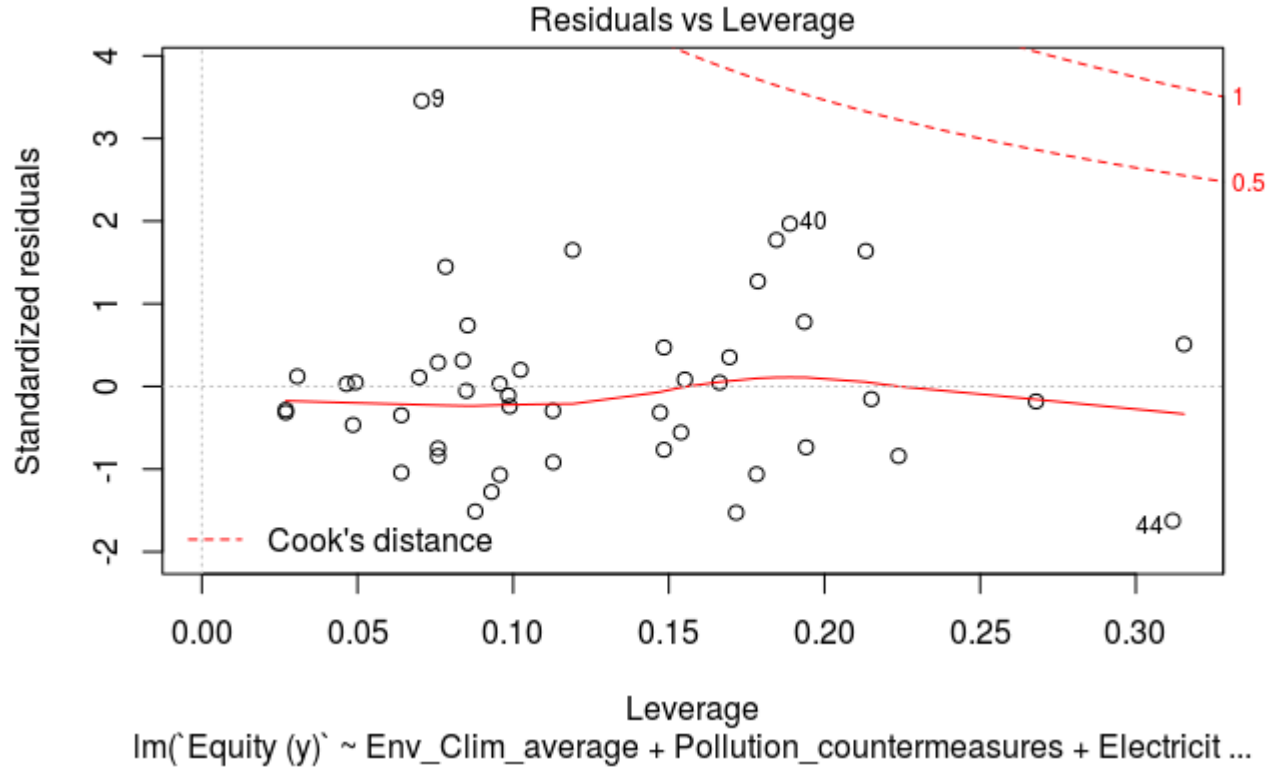
```
vif(equity_preference)
```

```
        Env_Clim_average Pollution_countermeasures        Electricity_Prices
                1.116380                  1.270208                  1.176986
         Com_Tax_average                Employment
                1.950854                  2.275971
```


Residuals vs Fitted
Residuals
Fitted values
lm(`Equity (y)` ~ Env_Clim_average + Pollution_countermeasures + Electricit ...
9
40
8
-1
0
1
2
3
-0.5
0.0
0.5
1.0


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(`Equity (y)` ~ Env_Clim_average + Pollution_countermeasures + Electricit ...
9
40
8
-1
0
1
2
3
4
-2
-1
0
1
2





5.4.2 Effect of Community Preferences on unweighted Equity Outcomes

Hide

```
summary(equity_preference)
```

```
Call:
lm(formula = `Equity (y)` ~ Env_Clim_average + Pollution_countermeasures +
    Electricity_Prices + Com_Tax_average + Employment, data = .)

Residuals:
     Min       1Q   Median       3Q      Max
-1.36246 -0.64535 -0.09826  0.29244  3.14181

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)               -3.576e-17  1.376e-01   0.000   1.0000
Env_Clim_average           8.199e-02  1.469e-01   0.558   0.5799
Pollution_countermeasures  9.389e-02  1.567e-01   0.599   0.5524
Electricity_Prices         2.657e-01  1.509e-01   1.761   0.0856 .
Com_Tax_average            5.129e-01  1.942e-01   2.640   0.0117 *
Employment                -4.227e-01  2.098e-01  -2.015   0.0505 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9432 on 41 degrees of freedom
Multiple R-squared:  0.2071,    Adjusted R-squared:  0.1104
F-statistic: 2.142 on 5 and 41 DF,  p-value: 0.07959
```

5.5 Effect of Community Preferences on Change in Equity Outcomes

Hide

```
summary(change_equity_preference)
```

```
Call:
lm(formula = Equity_change ~ Env_Clim_average + Pollution_countermeasures +
    Electricity_Prices + Com_Tax_average + Employment, data = .)

Residuals:
     Min       1Q   Median       3Q      Max
-3.09130 -0.42256  0.05912  0.60179  1.67093

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                -2.937e-17  1.388e-01   0.000   1.0000
Env_Clim_average           -4.546e-02  1.482e-01  -0.307   0.7606
Pollution_countermeasures -1.901e-01  1.581e-01  -1.202   0.2361
Electricity_Prices         -2.758e-01  1.522e-01  -1.813   0.0772 .
Com_Tax_average            -4.049e-01  1.959e-01  -2.067   0.0451 *
Employment                  3.592e-01  2.116e-01   1.697   0.0972 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9512 on 41 degrees of freedom
Multiple R-squared:  0.1935,    Adjusted R-squared:  0.09514
F-statistic: 1.967 on 5 and 41 DF,  p-value: 0.104
```

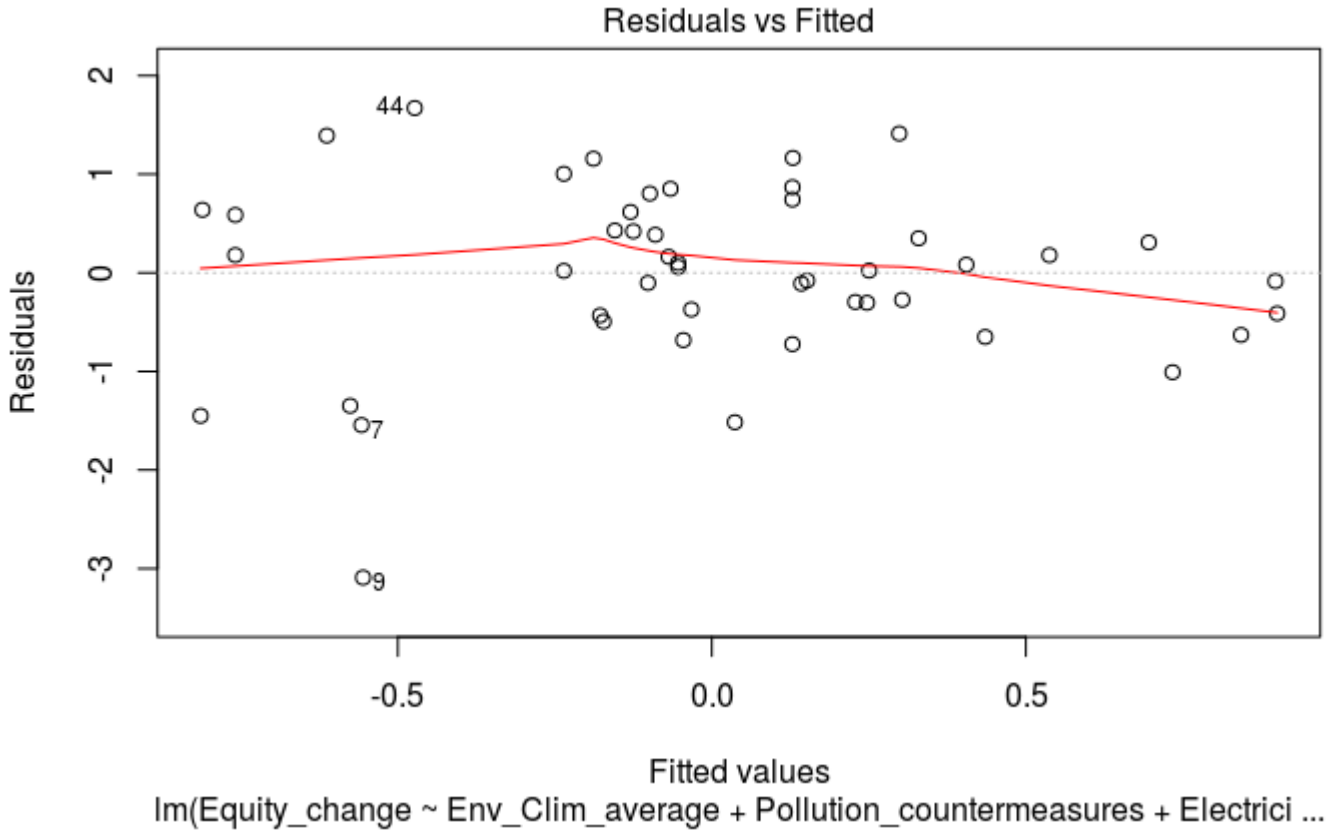
Multicolinearity diagnostics

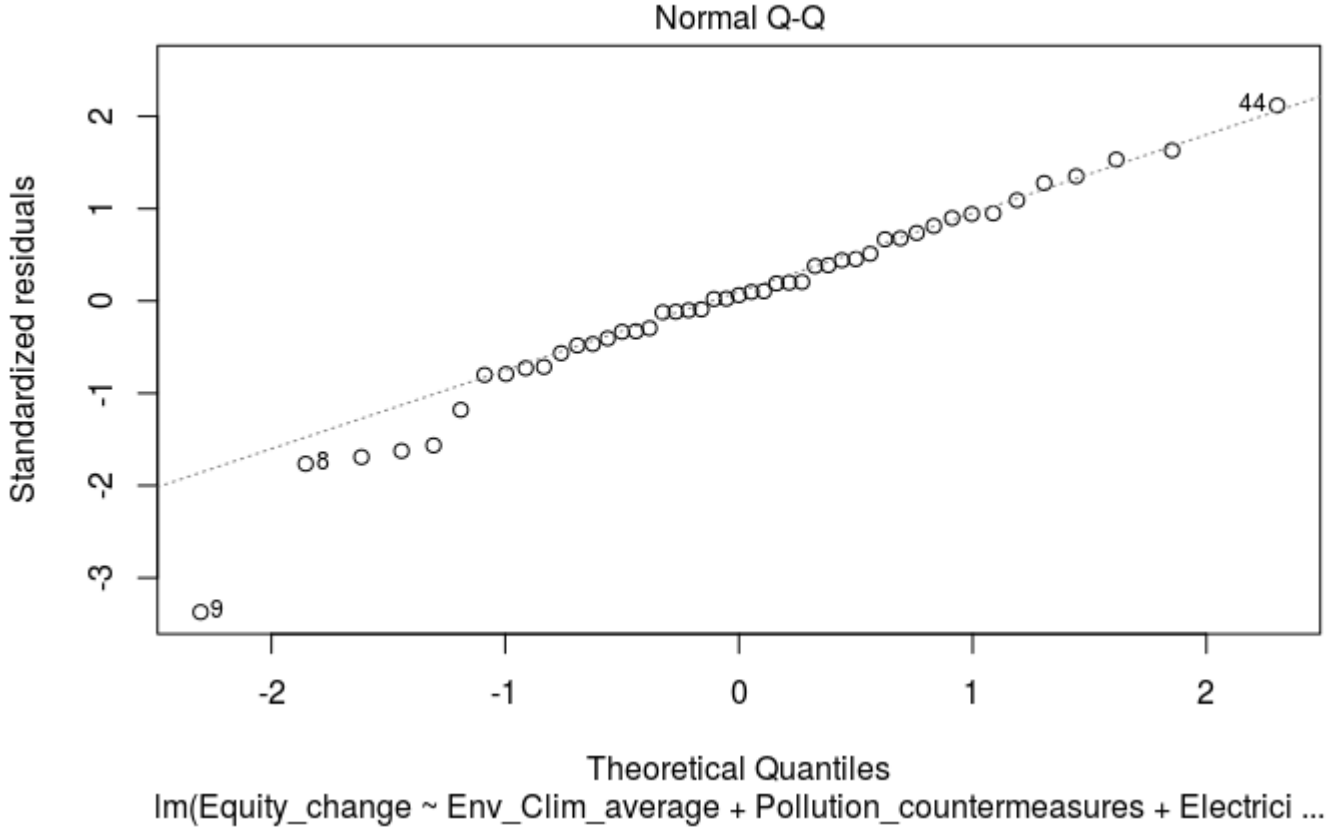
Hide

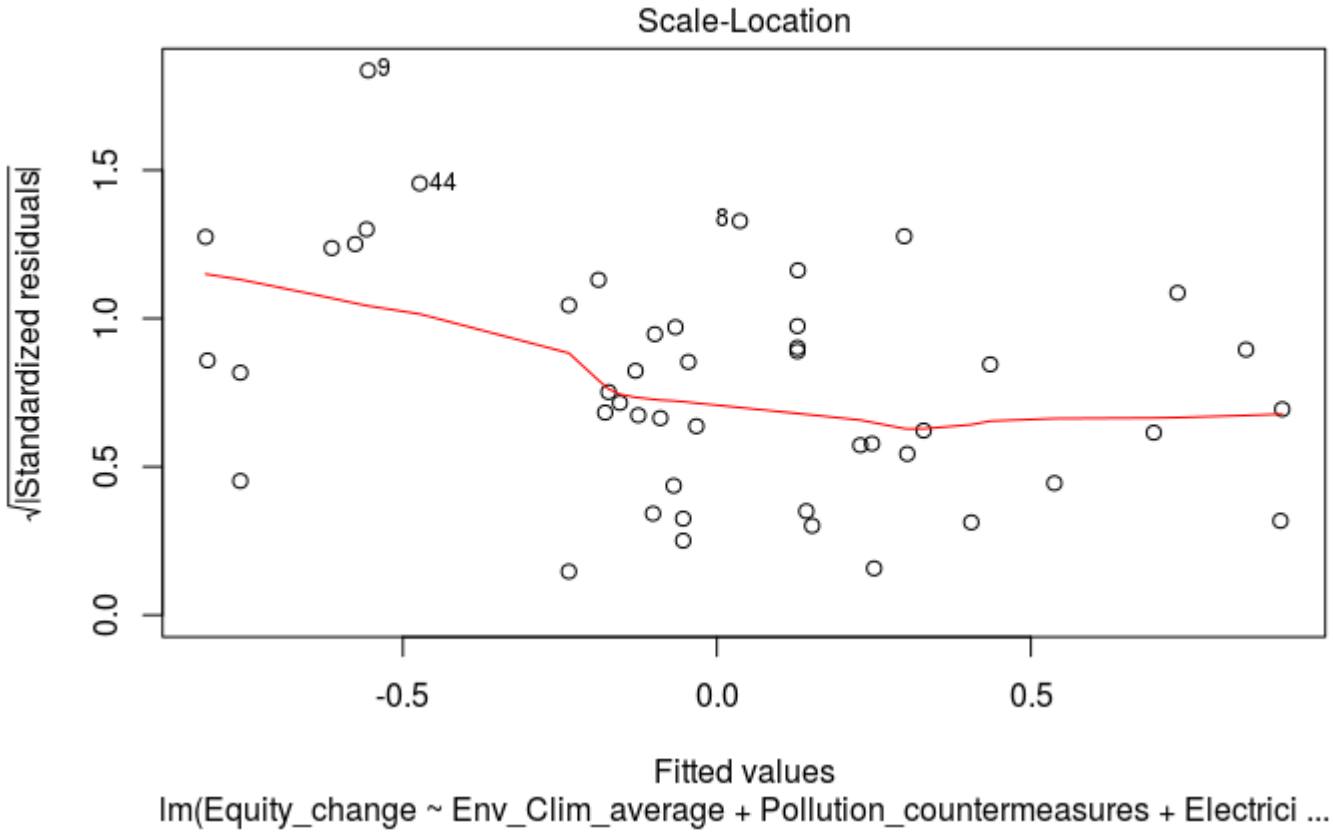
```
vif(change_equity_preference)
```

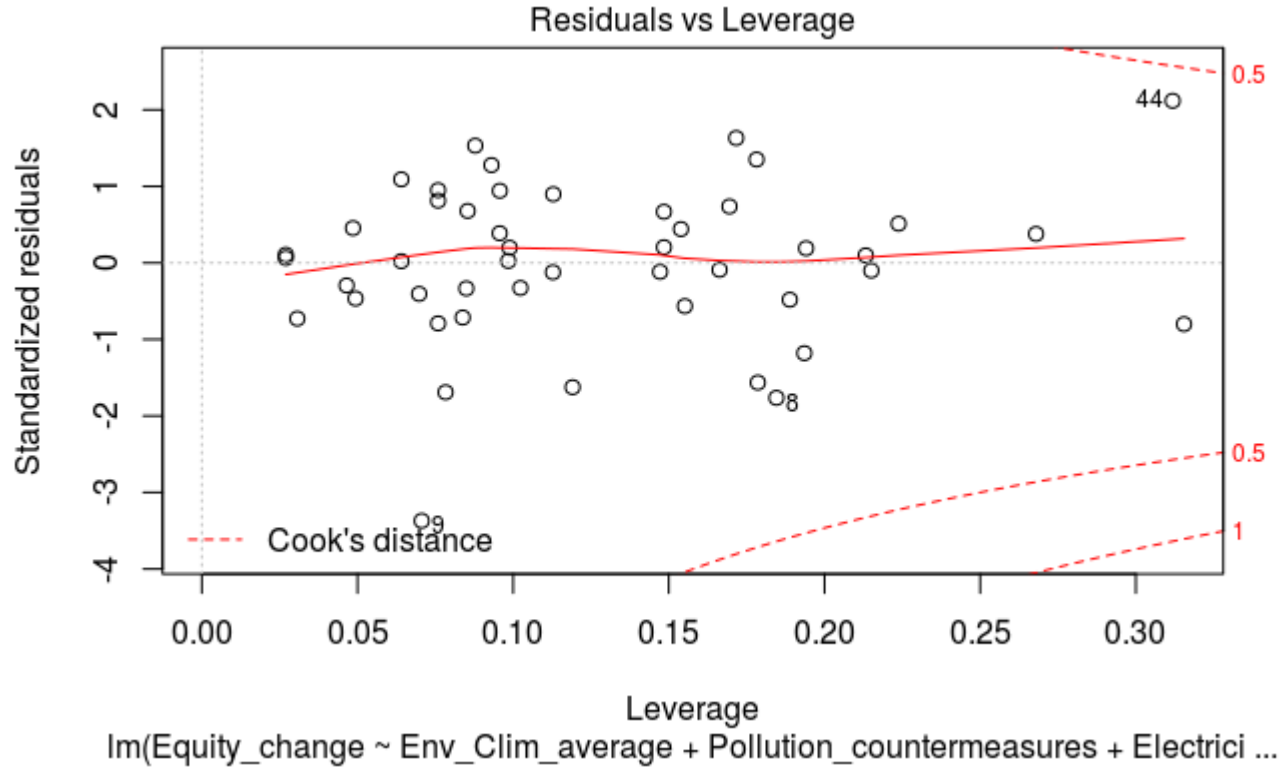
```
        Env_Clim_average Pollution_countermeasures        Electricity_Prices
                1.116380                  1.270208                  1.176986
         Com_Tax_average                Employment
                1.950854                  2.275971
```

Heteroskedasticity diagnostics


Residuals vs Fitted
Residuals
Fitted values
lm(Equity_change ~ Env_Clim_average + Pollution_countermeasures + Electrici ...
44
7
9
2
1
0
-1
-2
-3
-0.5
0.0
0.5


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Equity_change ~ Env_Clim_average + Pollution_countermeasures + Electrici ...
44
8
9
2
1
0
-1
-2
-3





5.6 Effect of Equity factors on Equity & Equity per GWh

Hide

```
vif(epg.tech) # can't use because of multicolinearity
```

```
       `CO2 Offset (weighted)`          `PM Offset (weighted)` `Electricity Prices (weighted)`
                    1542.366468                        9.493532                         1.174966
    `Community Tax (weighted)`        `Employment (weighted)`       `Satisfaction (weighted)`
                    2341.275841                      582.790339                         1.024218
```

Hide

```
# can't use because of multicolinearity
vif(epg.tech)
```

```
       `CO2 Offset (weighted)`          `PM Offset (weighted)` `Electricity Prices (weighted)`
                    1542.366468                        9.493532                         1.174966
    `Community Tax (weighted)`        `Employment (weighted)`       `Satisfaction (weighted)`
                    2341.275841                      582.790339                         1.024218
```

6. Effect of Demographics on Equity per GWh

Measure <chr>	**Correlation: Equity** <dbl>	**Correlation: Equity per GWh** <dbl>
Age_median_2010	0.124	-0.200
Income_taxable_per_capita_2010	-0.078	0.150
Percent_University Educated_2010	-0.075	0.509
PercentTotal_Migration_2010	0.203	0.362
Population_2010	-0.120	0.056
Ratio_Revs_Exp_2010	-0.077	0.064
Unemployment_2010	0.007	-0.035

7 rows

Hide

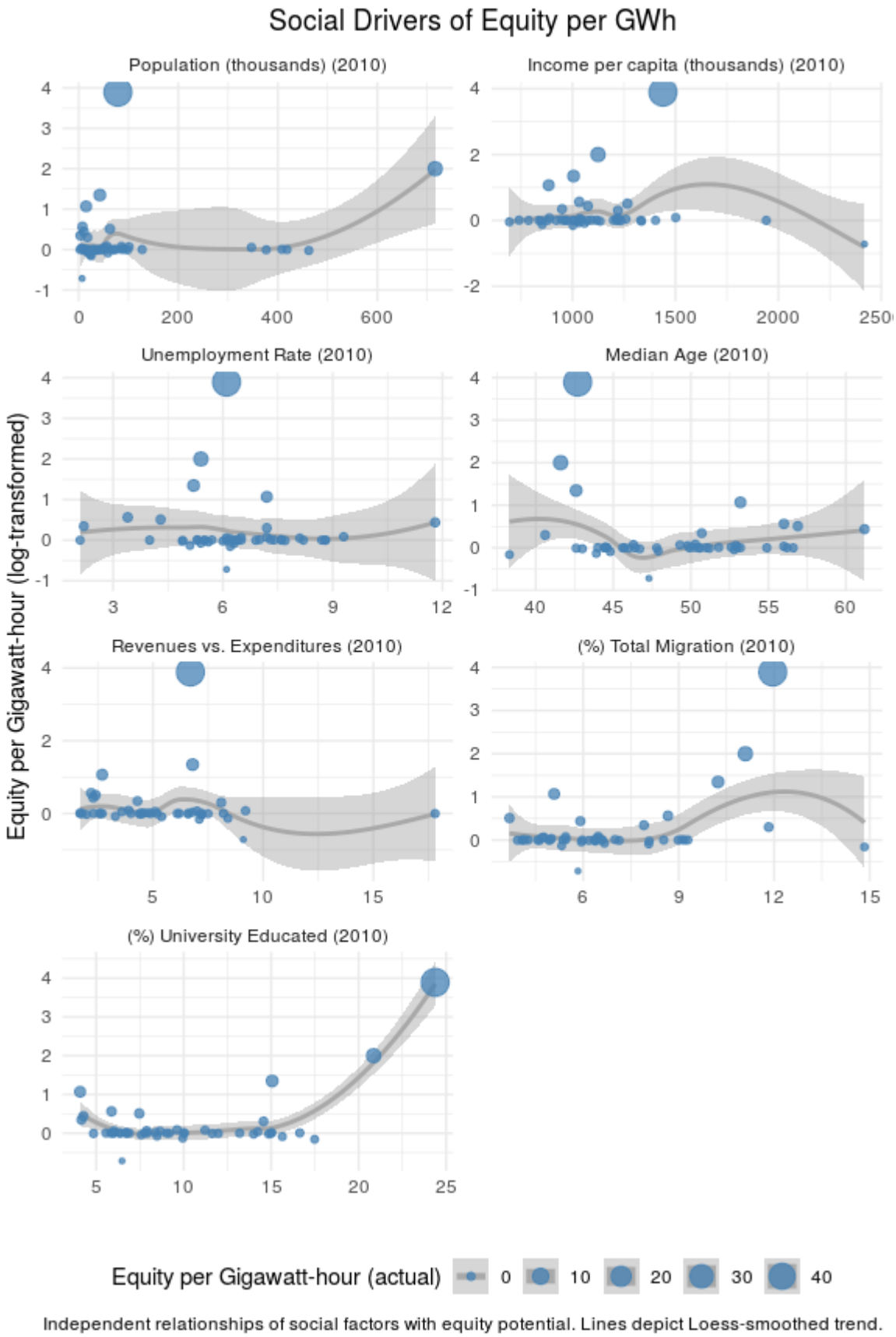
```
summary(epg.dem)
```

```
Call:
lm(formula = formula, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4468 -0.3531 -0.0002  0.2219  3.5987

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                       -0.04308    0.12526  -0.344  0.73279
Population_2010                   -0.16663    0.20208  -0.825  0.41475
Income_taxable_per_capita_2010    -0.02446    0.12806  -0.191  0.84953
Unemployment_2010                 -0.03780    0.13170  -0.287  0.77565
Age_median_2010                    0.31872    0.18225   1.749  0.08841 .
Ratio_Revs_Exp_2010               -0.16926    0.13907  -1.217  0.23109
PercentTotal_Migration_2010        0.19916    0.18105   1.100  0.27825
`Percent_University Educated_2010`  0.59029    0.19664   3.002  0.00472 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.841 on 38 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.3114,    Adjusted R-squared:  0.1845
F-statistic: 2.454 on 7 and 38 DF,  p-value: 0.03522
```

Hide

```
exp(0.590)
```

```
[1] 1.803988
```



However, this is a very small sample. The fact that we garnered statistically significant results is great, but we would feel better if we could use another method that retains validity despite small sample sizes. For this reason, we turn to permutation tests.

Permutation tests create 1000 different versions of the dataset, where everything remains the same, except the outcome variable gets randomly shuffled. This allows us to see what statistics we would get if the observed outcomes were assigned by chance. The more permutations, the higher detail we can use to justify whether our results really are statistically significant.

Hide

```
# First, we can gather 1000 permutations of the dataset

perms <- more %>%
  # Transform outcome to decrease heteroskedasticity
  mutate(`Equity per GWh` = log(`Equity per GWh` + 1)) %>%
  # Select variables for model
  select(`Equity per GWh`,   Income_taxable_per_capita_2010, Unemployment_2010,
         Ratio_Revs_Exp_2010, Population_2010, Age_median_2010,
         `Percent_University Educated_2010`,
         PercentTotal_Migration_2010) %>%
  # rescale variables
  scale() %>% as.data.frame() %>%
  # Permute 1000 times, shuffling the Equity per GWh variable
  permute(n = 1000, `Equity per GWh`)
```

Hide

```
# Second, we run models on these 1000 permuted datasets
models <- map(perms$perm, ~ lm(formula = formula, data = .))
remove(perms)
```

Hide

```
# Third, we collect model-level statistics from each model using the glance function
null.model <- map_df(models, broom::glance, .id = "id")

# Fourth, we collect the coefficient tables for each model using the tidy function
null.coef <- map_df(models, broom::tidy, .id = "id")
```

Hide

```
# Was the observed model statistic (F-statistic) more extreme than the statistics generated by c
hance?
# Evaluate if it was greater than or not (TRUE/FALSE == 1/0)
# Add together, and divide by total
mean(null.model$statistic > broom::glance(epg.dem)$statistic)
```

Hide

```
null.coef %>%
  # Grab the estimate
  select(id, term, estimate) %>%
  # Join in the observed estimate and std. error
  left_join(by = "term",
            y = epg.dem %>%
              tidy() %>%
              select(term,
                     estimate.obs = estimate,
                     std.error.obs = std.error)) %>%
  # For each term,
  group_by(term) %>%
  summarize(
    # Get the original observed estimate
    estimate.perm = mean(estimate.obs),
    # and calculate the percentile
    # how many times out of the total that the observed was more extreme than the null
    percentile = mean(estimate > mean(estimate.obs))
  ) %>%
  # Now convert that into a p.value by...
  # Getting how far it is from the middle
  mutate(
    estimate.perm = estimate.perm %>% round(3),
    p.value = if_else(
    percentile > 0.50,
    1 - percentile,
    percentile))
```

Good news; pretty solid verification of results via permutation tests as well, which tend to be more robust to small-sample sizes.

Hide

```
remove(null.coef,
       null.model,
       models,
       more, more_longer)
```

7. Bivariate correlations to find pearson product moment correlation coefficients

Because we couldn't assess the effect of equity components on the equity scores we created due to multicolinearity, instead, we use pearson's r, the correlation coefficient, to describe this.

Hide

```
dat %>%
  # Select equity outcomes and equity components (raw measures used to build equity outcomes)
  select(`Town`, `Equity (y)`, `Equity per GWh`,
         `CO2 Offset per capita (Total)`,`PM Offset per capita (Total)`,
         `Savings per kWh (weighted average)`,`FAT Revenue per capita (Total)`,
         `Net Jobs per capita (Total)`,`RE kW unwanted per capita`) %>%
  # Now pivot longer into a tidy format
  pivot_longer(
    # Holding Town ID and the equity outcomes as indexes
    cols = -c(`Town`, `Equity (y)`, `Equity per GWh`),
    names_to = "measure",
    values_to = "value") %>%
  # Now for each equity component
  group_by(`Measure` = measure) %>%
  summarize(
    # Calculate Pearson's r for its correlation with Equity
    `Correlation: Equity` = cor(
      `Equity (y)`, value,
      use = "complete.obs", method = "pearson") %>%
      # rounded to 3 digits
      round(3),
    # Calculate Pearson's r for its correlation with Equity per GWh
    `Correlation: Equity per GWh` = cor(
      `Equity per GWh`, value,
      use = "complete.obs", method = "pearson") %>%
      # rounded to 3 digits
      round(3))
```