

Replication of Guiso, Sapienza, and Zingales (2016)

```
packages = c("dplyr", "readr", "haven", "estimatr",  
             "conleyreg", "lmtest", "sandwich", "tibble",  
             "quantreg")  
sapply(packages, library, character.only = TRUE)
```

Load in the data:

```
city_data = read_dta("../datasets/3-guiso-sapienza-zingales/ltp1F.dta")  
istat_codes = read_delim("../datasets/3-guiso-sapienza-zingales/Codici-statistici-e-denominazioni-istat",  
                        delim = ";",  
                        locale = locale(encoding = "ISO-8859-1"))
```

Rows: 7903 Columns: 23

-- Column specification -----

Delimiter: ";"

chr (17): Codice Regione, Codice dell'Unità territoriale sovracomunale
(val...

dbl (6): Codice Ripartizione Geografica, Tipologia di Unità territoriale so...

i Use `spec()` to retrieve the full column specification for this data.

i Specify the column types or set `show_col_types = FALSE` to quiet this message.

```
coords = read_csv("../datasets/3-guiso-sapienza-zingales/places.csv")
```

Rows: 14666 Columns: 6

-- Column specification -----

Delimiter: ","

chr (2): name, type

dbl (4): X, Y, osm_id, population

i Use ``spec()`` to retrieve the full column specification for this data.

i Specify the column types or set ``show_col_types = FALSE`` to quiet this message.

Perform necessary merges:

```
istat_codes = istat_codes |>
  select(`Codice Comune formato numerico`, `Denominazione in italiano`)
coords = coords |>
  select(X, Y, name)

city_merged = city_data |>
  left_join(istat_codes, by = c("istcom" = "Codice Comune formato numerico")) #|>
# left_join(coords, by = c("Denominazione in italiano" = "name"))

city_final = city_merged |>
  filter(dummyroma == 0,
         regione < 20 | is.na(regione),
         totassoc_p > 0) |>
  janitor::clean_names()
```

Heteroscedasticity?

```
mod1 = lm(totassoc_p ~ libero_comune_allnord + altitudine + escursione +
          costal + nearsea + population + pop2 + gini_income + gini_land,
          data = city_final,
          weights = population)
coeftest(mod1, vcov = vcovHC(mod1, type = "HC3"))
```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.70894	1.34017	-0.5290	0.5968317
libero_comune_allnord	1.84100	0.43534	4.2289	2.387e-05 ***
altitudine	1.96676	0.57607	3.4141	0.0006447 ***
escursione	1.38597	0.28237	4.9084	9.457e-07 ***
costal	0.24669	0.40958	0.6023	0.5470004
nearsea	1.07509	0.64395	1.6695	0.0950742 .
population	-4.21584	6.12787	-0.6880	0.4914964

```

pop2                1.41368    7.08761  0.1995 0.8419124
gini_income         10.02068    3.23643  3.0962 0.0019703 **
gini_land           0.64744    0.74096  0.8738 0.3822736
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Removing outliers:

```

mod2 = lm(totassoc_p ~ libero_comune_allnord + altitudine + escursione +
          costal + nearsea + population + pop2,
          data = city_final |> filter(totassoc_p < 75),
          weights = population)
coeftest(mod2, vcov = vcovHC(mod2, type = "HC1"))

```

t test of coefficients:

```

              Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.54899    0.11671 30.4086 < 2.2e-16 ***
libero_comune_allnord 2.06447    0.33467  6.1687 7.391e-10 ***
altitudine     2.22274    0.37737  5.8900 4.097e-09 ***
escursione     1.26338    0.18882  6.6910 2.443e-11 ***
costal         0.40435    0.31073  1.3013  0.1932
nearsea        0.41561    0.26706  1.5563  0.1197
population    -3.10611    2.71257 -1.1451  0.2522
pop2           1.10589    2.03365  0.5438  0.5866
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

med = rq(totassoc_p ~ libero_comune_allnord + altitudine + escursione +
          costal + nearsea + population + pop2,
          tau = c(0.1, 0.5, 0.9),
          data = city_final,
          weights = population)
summary(med, se = "boot", R = 1000)

```

```

Call: rq(formula = totassoc_p ~ libero_comune_allnord + altitudine +
          escursione + costal + nearsea + population + pop2, tau = c(0.1,
          0.5, 0.9), data = city_final, weights = population)

```

tau: [1] 0.1

Coefficients:

	Value	Std. Error	t value	Pr(> t)
(Intercept)	2.24176	0.18648	12.02143	0.00000
libero_comune_allnord	2.77975	0.39995	6.95017	0.00000
altitudine	-0.20675	0.42988	-0.48095	0.63057
escursione	0.52992	0.16232	3.26468	0.00110
costal	-0.61811	0.43588	-1.41806	0.15623
nearsea	0.26045	0.23835	1.09276	0.27455
population	-2.59393	4.75395	-0.54564	0.58534
pop2	1.52445	7.04066	0.21652	0.82859

Call: rq(formula = totassoc_p ~ libero_comune_allnord + altitudine +
escursione + costal + nearsea + population + pop2, tau = c(0.1,
0.5, 0.9), data = city_final, weights = population)

tau: [1] 0.5

Coefficients:

	Value	Std. Error	t value	Pr(> t)
(Intercept)	3.52713	0.18561	19.00283	0.00000
libero_comune_allnord	2.67512	0.63495	4.21314	0.00003
altitudine	0.77385	0.61995	1.24825	0.21199
escursione	1.26837	0.26078	4.86378	0.00000
costal	0.46696	0.36223	1.28913	0.19741
nearsea	0.19818	0.32520	0.60942	0.54227
population	-6.90078	7.11075	-0.97047	0.33185
pop2	4.10781	10.67336	0.38487	0.70035

Call: rq(formula = totassoc_p ~ libero_comune_allnord + altitudine +
escursione + costal + nearsea + population + pop2, tau = c(0.1,
0.5, 0.9), data = city_final, weights = population)

tau: [1] 0.9

Coefficients:

	Value	Std. Error	t value	Pr(> t)
(Intercept)	5.34664	0.21079	25.36456	0.00000
libero_comune_allnord	1.16673	0.66007	1.76758	0.07719
altitudine	3.67252	0.85839	4.27838	0.00002
escursione	1.84271	0.41906	4.39727	0.00001
costal	0.45201	0.59497	0.75972	0.44746

nearsea	-0.06251	0.47944	-0.13039	0.89627
population	0.55976	6.64987	0.08418	0.93292
pop2	-2.26874	14.75011	-0.15381	0.87776