

Replication of Guiso, Sapienza, and Zingales (2016)

```
packages = c("dplyr", "readr", "haven", "estimatr",  
             "conleyreg", "lmtest", "sandwich", "tibble",  
             "quantreg", "stringr", "spatInfer", "modelsummary",  
             "ggplot2", "tinytable")  
sapply(packages, library, character.only = TRUE)
```

Load in the data:

```
city_data = read_dta("../datasets/3-guiso-sapienza-zingales/ltp1F.dta")  
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.

Load in comprehensive ISTAT code data:

```
codici = read_delim("../datasets/3-guiso-sapienza-zingales/codici-istat.csv",  
                    delim = ";",  
                    locale = locale(encoding = "ISO-8859-1")) |>  
  janitor::clean_names()
```

Rows: 7896 Columns: 26

-- Column specification -----

Delimiter: ";"

chr (19): Codice Regione, Codice dell'Unità territoriale sovracomunale
(val...

dbl (7): 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.

```
preced = read_delim("../datasets/3-guiso-sapienza-zingales/comuni-precedenti.csv",  
                    delim = ";",  
                    locale = locale(encoding = "ISO-8859-1")) |>  
  janitor::clean_names()
```

Rows: 2670 Columns: 6

-- Column specification -----

Delimiter: ";"

chr (6): Codice Provincia, Codice di denominazione, Denominazione precedente...

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.

```
soppr = read_delim("../datasets/3-guiso-sapienza-zingales/comuni-soppressi.csv",  
                   delim = ";",  
                   locale = locale(encoding = "ISO-8859-1")) |>  
  janitor::clean_names()
```

New names:

Rows: 2634 Columns: 16

-- Column specification

----- Delimiter: ";" chr

(9): Sigla Automobilistica, Codice Unità territoriale sovracomunale (Uts... dbl

(2): Anno, Comune soppresso per scorporo lgl (5): ...12, ...13, ...14, ...15,
...16

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.

* `` -> `...12`

* `` -> `...13`

* `` -> `...14`

* `` -> `...15`

* `` -> `...16`

```

codici1 = codici |>
  select(denominazione_in_italiano,
         codice_comune_numerico_con_103_province_dal_1995_al_2005) |>
  rename(codice_comune = codice_comune_numerico_con_103_province_dal_1995_al_2005)

soppr1 = soppr |>
  select(codice_comune, denominazione_comune) |>
  mutate(codice_comune = str_replace(codice_comune, "^0+", ""),
         codice_comune = as.double(codice_comune))

preced1 = preced |>
  select(denominazione_precedente, codice_comune) |>
  mutate(codice_comune = str_replace(codice_comune, "^0+", ""),
         codice_comune = as.double(codice_comune))

istat = codici1 |>
  full_join(soppr1, by = c("codice_comune" = "codice_comune",
                          "denominazione_in_italiano" = "denominazione_comune")) |>
  full_join(preced1, by = c("codice_comune" = "codice_comune",
                          "denominazione_in_italiano" = "denominazione_precedente")) |>
  group_by(codice_comune) |>
  slice(1) |>
  ungroup()

```

Perform necessary merges:

```

#standardize_name = function(name) {
#  name |>
#    str_to_lower() |>
#    str_replace_all("[:punct:]", "") |>
#    str_squish()
#}

coords = coords |>
  group_by(name) |>
  filter(n() == 1 | population > 0) |>
  slice(1) |>
  ungroup() |>
  select(X, Y, name) #|>
#  mutate(name = standardize_name(name))

city_merged = city_data |>

```

```

left_join(istat, by = c("istcom" = "codice_comune")) |>
# mutate(name = standardize_name(denominazione_in_italiano)) |>
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() |>
  rename(X = x, Y = y)

```

Estimate Column V of Table 3:

```

mod1 = lm(totassoc_p ~ libero_comune_allnord + altitudine + escursione +
          costal + nearsea + population + pop2 + gini_income + gini_land +
          income_p,
          data = city_final |> filter(!is.na(X)),
          weights = population)
coeftest(mod1, vcov = vcovHC(mod1, type = "HC1"))

```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-1.501537	0.905406	-1.6584	0.0972955	.
libero_comune_allnord	1.550765	0.272854	5.6835	1.393e-08	***
altitudine	1.715161	0.470266	3.6472	0.0002677	***
escursione	1.238985	0.233244	5.3120	1.130e-07	***
costal	0.593419	0.236659	2.5075	0.0121903	*
nearsea	1.504856	0.649697	2.3162	0.0205845	*
population	-7.031120	1.784723	-3.9396	8.270e-05	***
pop2	3.264858	1.320322	2.4728	0.0134392	*
gini_income	-2.344066	2.226061	-1.0530	0.2923859	
gini_land	1.474642	0.485590	3.0368	0.0024030	**
income_p	0.373586	0.037885	9.8611	< 2.2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Implementing Conley and Kelly (2024) using the `spatInfer` package:

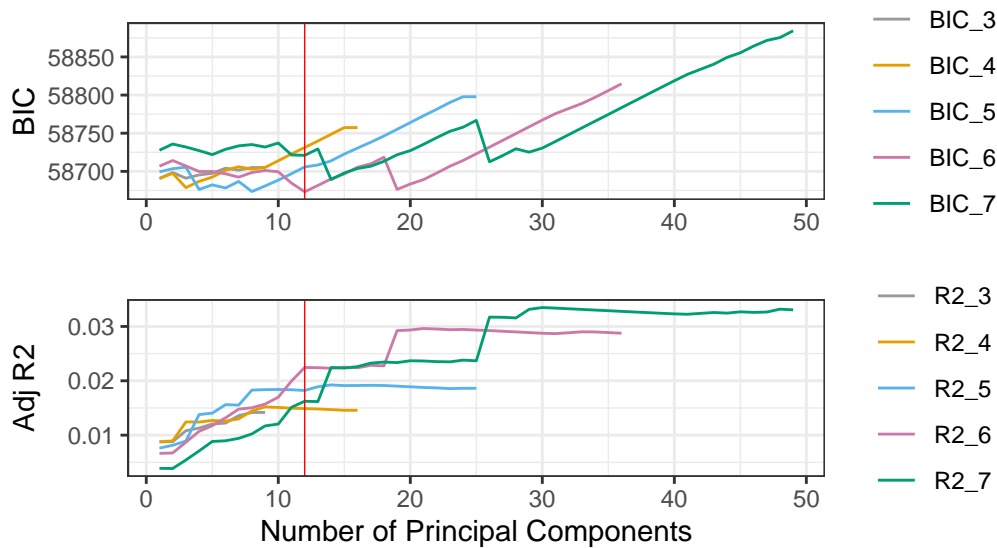
```

filtered = city_final |> filter(!is.na(X))

optimal_basis(fm = totassoc_p ~ libero_comune_allnord + altitudine + escursione +
  costal + nearsea + population + pop2 + gini_income + gini_land +
  income_p,
  df = filtered,
  max_splines = 7,
  Description = "Social Capital and Association Density")

```

Social Capital and Association Density BIC and R2 of 3x3 to 7x7 te
 BIC is minimized by a 6x6 Linear Tensor with 12 PCs.

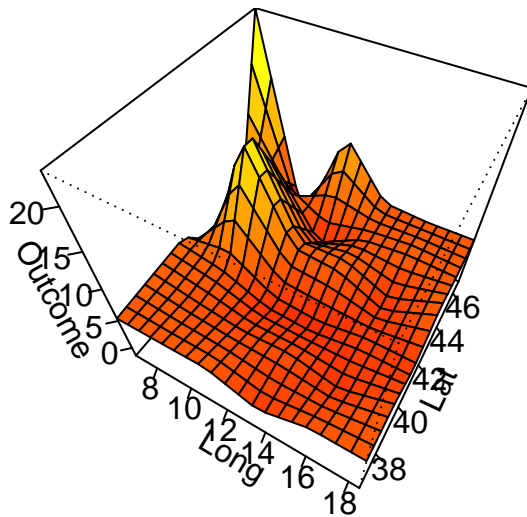


```

plot_basis(fm = totassoc_p ~ libero_comune_allnord + altitudine + escursione +
  costal + nearsea + population + pop2 + gini_income + gini_land +
  income_p,
  df = filtered,
  splines = 6,
  Title="6x6 Tensor Surface of Social Capital")

```

6x6 Tensor Surface of Social Capital



Run placebo:

```
#plbo = placebo(fm = totassoc_p ~ libero_comune_allnord + altitudine + escursione +  
#             costal + nearsea + population + pop2 + gini_income + gini_land +  
#             income_p,  
#             df = filtered,  
#             splines = 6,  
#             pc_num = 12,  
#             nSim = 200)  
  
#placebo_table(plbo)
```

Removing outliers:

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