Spatial Economics – Assignment 1

Max Heinze (h11742049@s.wu.ac.at) Kevin Kain (h12232066@s.wu.ac.at)
Jaime Miravet (h12235992@s.wu.ac.at)

April 2, 2024

Contents

The executable code that was used in compiling the assignment is available on GitHub at https://github.com/maxmheinze/spatial.

Task A

Preliminaries

First, we load the `MASS` package and check what variables there are in the `Boston` dataset.

```
# Header ----------------------------------------------------------------

rm(list = ls())
gc()

pacman::p_load(MASS)

# Check Column Names ----------------------------------------------------

colnames(Boston)
```

Creating the Function

Next, we create the desired function.

```
# Create the function ---------------------------------------------------

boston_quick_ols <- function(dependent, ...) {

    # Create a formula string from the inputs
    independents <- paste(c(...), collapse = " + ")
    formula_string <- paste(dependent, "~", independents)

    # Fit the model
    fitted_model <- lm(as.formula(formula_string), data = Boston)

    # Get the summary
    fitted_model_summary <- summary(fitted_model)

    # Get point estimates and confidence intervals
    list_coef <- fitted_model_summary$coefficients
    list_conf <- confint(fitted_model, level = 0.95)
    list_ervr <- fitted_model_summary$sigma^2

    # Output a list
    return(list(coefficients = list_coef[, 1], error_variance = list_ervr, test_statistic_t
    = list_coef[,
        3], test_statistic_p = list_coef[, 4], confidence_intervals = list_conf))

}
```

A Simple Linear Model

Next, we apply the function, using a collection of four independent variables.

```
boston_quick_ols("medv", "rm", "age", "dis", "nox")
```

```
## $coefficients
##  (Intercept)           rm          age          dis          nox
##  -6.61135440   8.00051949  -0.06932587  -1.08526888 -22.10858455
##
## $error_variance
## [1] 37.35166
##
## $test_statistic_t
```

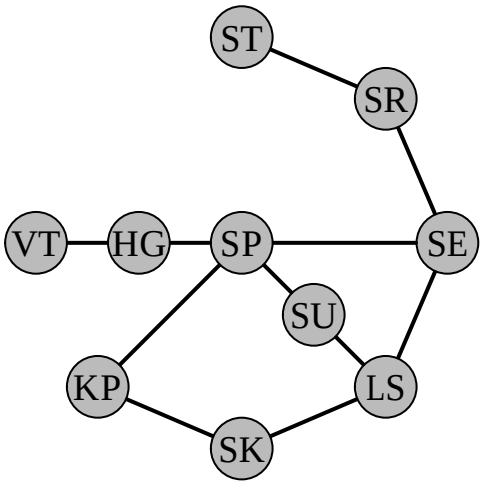
```
## (Intercept)          rm         age         dis         nox
##   -1.590287   19.654992   -4.422259   -4.850184   -5.486524
##
## $test_statistic_p
##  (Intercept)           rm          age          dis          nox
## 1.124008e-01 3.520952e-64 1.198751e-05 1.649044e-06 6.516890e-08
##
## $confidence_intervals
##                    2.5 %       97.5 %
## (Intercept) -14.7793094   1.55660058
## rm            7.2007886   8.80025034
## age          -0.1001258  -0.03852595
## dis          -1.5248891  -0.64564866
## nox         -30.0256124 -14.19155674
```

Task B

Creating a Graph and an Adjacency Matrix

We chose the network of all first-district Vienna subway stations. The graph and the adjacency matrix can be found below. Nodes represent individual stations, and edges represent direct subway connections between two stations, without passing another station or changing to another line. We abstract from the existence of different subway lines. The two-character node labels are to be read as follows: ST is Schottentor, SR is Schottenring, SE is Schwedenplatz, LS is Landstraße, SK is Stadtpark, KP is Karlsplatz, SU is Stubentor, SP is Stephansplatz, HG is Herrengasse, and VT is Volkstheater.

```r
# Header ----------------------------------------------------------------
pacman::p_load(igraph, extrafont)


# Create Matrix ---------------------------------------------------------

# Create the adjacency matrix
adj_matrix <- matrix(c(0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
    0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1,
    0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
    1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
    1, 0), nrow = 10, byrow = TRUE)

# Define node names
node_names <- c("ST", "SR", "SE", "LS", "SK", "KP", "SU", "SP", "HG", "VT")
dimnames(adj_matrix) <- list(node_names, node_names)

coords_matrix <- matrix(c(0, 0, 3.5, -1.5, 5, -5, 3.5, -8.5, 0, -10, -3.5, -8.5,
    1.75, -6.75, 0, -5, -2.5, -5, -5, -5), ncol = 2, byrow = TRUE)


# Create graph ----------------------------------------------------------

graph_1 <- graph_from_adjacency_matrix(adj_matrix, mode = "undirected")


# Output ----------------------------------------------------------------

plot(graph_1, layout = coords_matrix, vertex.size = 30, vertex.color = "#BBBBBB",
    vertex.label.cex = 1.2, vertex.label.font = 2, vertex.label.family = "Lato",
    vertex.label.color = "black", edge.color = "black", edge.width = 2)
```




```
knitr::kable(adj_matrix)
```

	ST	SR	SE	LS	SK	KP	SU	SP	HG	VT
ST	0	1	0	0	0	0	0	0	0	0
SR	1	0	1	0	0	0	0	0	0	0
SE	0	1	0	1	0	0	0	1	0	0
LS	0	0	1	0	1	0	1	0	0	0
SK	0	0	0	1	0	1	0	0	0	0
KP	0	0	0	0	1	0	0	1	0	0
SU	0	0	0	1	0	0	0	1	0	0
SP	0	0	1	0	0	1	1	0	1	0
HG	0	0	0	0	0	0	0	1	0	1
VT	0	0	0	0	0	0	0	0	1	0