

Assignment 1 - Group 8

Elia Di Gregorio and Robert Auerbach

2024-03-30

Contents

Exercise A	1
Exercise B	2
Exercise C	5
Different projections	5
Map and Dataset	6
Storing visualizations	7
Exercise D	7
Comparison of support for Komorowski and Duda	7
Postal voting envelopes anomalies	8
Turnout for each election round	9

Exercise A

The dependent variable `medv` shows the median value of owner-occupied homes in \$1000s. We chose the following covariates for our linear model: 1) `crim`: per capita crime rate by town. 2) `zn`: proportion of residential land zoned for lots over 25,000 sq.ft. 3) `indus`: proportion of non-retail business acres per town. 4) `chas`: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise). 5) `nox`: nitrogen oxides concentration (parts per 10 million).

Task: Create a function that takes your dependent variable and the covariates as inputs, and return a list with:
– OLS point estimates for the intercept, slope parameters, and the error variance. – Suitable test statistics with corresponding p-values for the relevant coefficients. – Intervals of the coefficients for a confidence level of 95%.

```
pp_pred <- function(dependent_variable, covariates) {  
  data <- data.frame(dependent_variable, covariates)  
  model <- lm(dependent_variable ~ ., data)  
  coefficients <- coef(model)  
  se <- summary(model)$coefficients[, "Std. Error"]  
  t_values <- coefficients / se  
  p_values <- 2 * pt(abs(t_values), df = df.residual(model), lower.tail = FALSE)  
  conf_int <- confint.default(model)  
  results <- data.frame(  
    "Coefficients" = coefficients,  
    "Standard Errors" = se,  
    "t-values" = t_values,  
    "p-values" = p_values,  
    "Confidence Intervals (95%)" = conf_int  
  )  
}
```

```
)

return(results)
}
```

Call the function and print list:

```
result <- property_price_prediction(dependent_variable, covariates)
print(result)
```

```
##              Coefficients Standard.Errors   t.values    p.values
## (Intercept)  29.48994059      2.22434765  13.257793  1.365091e-34
## crim        -0.21851904      0.04389177  -4.978588  8.831134e-07
## zn          0.05511047      0.01743801   3.160365  1.671221e-03
## indus       -0.38348055      0.07944258  -4.827141  1.843157e-06
## chas        7.02622266      1.33711744   5.254754  2.198513e-07
## nox        -5.42465902      4.69550757  -1.155287  2.485247e-01
##              Confidence.Intervals..95...2.5.. Confidence.Intervals..95...97.5..
## (Intercept)                25.13029931                33.84958187
## crim                  -0.30454533                -0.13249275
## zn                    0.02093261                 0.08928834
## indus                 -0.53918515                -0.22777594
## chas                   4.40552064                 9.64692469
## nox                  -14.62768474                 3.77836670
```

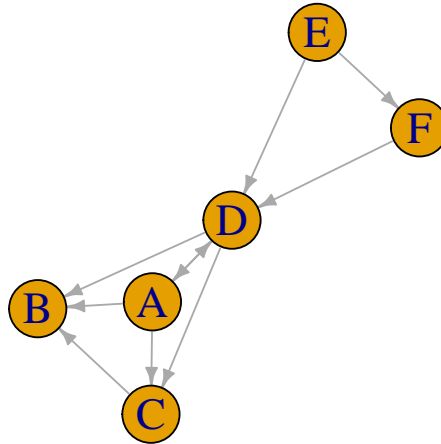
Exercise B

Task: Come up with some network of interest, vaguely related to some real-world example (describe very briefly), with at least six agents and ten edges between them.

Our real world example is one of a supply chain network where there is rarely any reciprocity: construction of airplanes. The main vertices is the firm constructing the airplane (B), where all manufactured parts end up (high in-degree, low out-degree). On the other hand, among manufactures of airplane parts little reciprocity might occur as they need specialized part to finish their own parts (D and A).

The adjacency matrix in R.can be drawn as follows:

```
plot(g, edge.arrow.size = 0.5, vertex.label.cex = 1.5, vertex.size = 30)
```



```
print(adj_matrix)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following objects are masked from 'package:tidyr':
##
##   expand, pack, unpack
## 5 x 5 sparse Matrix of class "dgCMatrix"
##   A B C D E
## A . 1 1 1 .
## B . . . . .
## C . 1 . . .
## D 1 1 1 . .
## E . . . 1 .
```

Who are the most and least central agents in the network? Name, explain, and try to quantify different notions of centrality.

We quantified the in- and out-degree of centrality. The in-degree shows how many links are directed towards a node, while the out-degree shows how many links are directed from a node (to another). A node from which a link goes out to another is considered a supplier, a node which is receiving links is considered a buyer. The `degree_table` shows how many links go in and out from each node. This way we can quantify the most central buyer and supplier in our network. According to the table B and D have the most links directed towards them while E has none directed towards it. Hence, B and C are the most central “buyers”, E the least. Also, A and D have the most links directed towards others while B has none directed towards others. Hence A and D are the most central “suppliers” and B the least.

```
degree_table
```

```
##      A B C D E F
## In  1 3 2 3 0 1
## Out 3 0 1 3 2 1
```

How would centrality change if you considered a row-normalized network instead?

We calculate the sums of each row of the adjacency matrix and then divide the adjacency matrix by the row sums. Again, we compute the centrality measures of in- and out-degree. As expected (based on the slides) the out-degree of the agents are equalized to 1. Also, the most central buyers are now B and D instead of B and C. Clearly the row-normalization leads to a distortion.

```
degree_table_norm
```

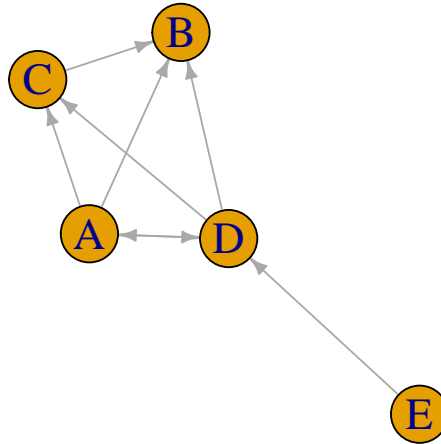
```
##      A      B      C      D E  F
## In  0.3333333 1.666667 0.6666667 1.833333 0 0.5
## Out 1.0000000 0.000000 1.0000000 1.000000 1 1.0
```

How would the network change if you removed or added a specific agent? We removed agent “F” and repeated the steps from above.

The most central buyer is now B alone, E still is the least central buyer. The most central supplier is still A and D, the least is still B.

We also compared the reciprocity and transitivity of both networks. The reciprocity is higher with the agent F removed, which is due to the fact that the network has a higher share of reciprocated links relative to the amount of possible reciprocated links. Also, the so called clustering coefficient (Transitivity) is higher in the Network with fewer agents as there is a 10% higher probability that vertices are connected. This is especially true for a network with low reciprocity such as ours.

```
plot(g1, edge.arrow.size = 0.5, vertex.label.cex = 1.5, vertex.size = 30)
```



```
degree_table_g1
```

```
##      A B C D E
## In  1 3 2 2 0
## Out 3 0 1 3 1
```

```
rectrans_table
```

```
##           Network Complete Network Removed
## Reciprocity      0.2000000      0.25
## Transitivity     0.7142857      0.80
```

Exercise C

In this exercise, we will work with spatial projections of Türkiye's NUTS-2 regions. Despite not being a member of the European Union, it has adopted the Nomenclature of Territorial Units for Statistics (NUTS) since 2002.

Different projections

Türkiye counts 26 subregions, and we accessed its spatial data directly from the GISCO source via `get_eurostat_spatial()` function. By default this function loads the EPSG-4326 projection of the map, corresponding to the World Geodetic System 1984 ensemble (WGS84). It is based on a geocentric datum, meaning it defines the Earth's shape as an ellipsoid (a flattened sphere) rather than a perfect sphere.

```
st_crs(Europe)
```

```
## Coordinate Reference System:
##   User input: EPSG:4326
##   wkt:
```

```

## GEOGCRS["WGS 84",
##     ENSEMBLE["World Geodetic System 1984 ensemble",
##         MEMBER["World Geodetic System 1984 (Transit)"],
##         MEMBER["World Geodetic System 1984 (G730)"],
##         MEMBER["World Geodetic System 1984 (G873)"],
##         MEMBER["World Geodetic System 1984 (G1150)"],
##         MEMBER["World Geodetic System 1984 (G1674)"],
##         MEMBER["World Geodetic System 1984 (G1762)"],
##         MEMBER["World Geodetic System 1984 (G2139)"],
##         ELLIPSOID["WGS 84",6378137,298.257223563,
##             LENGTHUNIT["metre",1]],
##         ENSEMBLEACCURACY[2.0]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433]],
##     CS[ellipsoidal,2],
##         AXIS["geodetic latitude (Lat)",north,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433]],
##         AXIS["geodetic longitude (Lon)",east,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433]],
##     USAGE[
##         SCOPE["Horizontal component of 3D system."],
##         AREA["World."],
##         BBOX[-90,-180,90,180]],
##     ID["EPSG",4326]]

```

To use another projection and/or CRS we employed the `st_transform()` function and recurred to the Equal Earth Projection and to the Lambert Azimuthal Equal Area Projection. Both preserve the relative sizes of areas on the Earth's surface. This means that areas on the map are represented accurately in relation to each other in terms of size, making it suitable for thematic mapping and spatial analysis. However the former is classified as a pseudo-cylindrical projection because it projects the Earth's surface onto a cylinder and then unwraps the cylinder to form a rectangular map, whereas the latter projects the Earth's surface onto a plane tangent to a specific point (the center of the projection). The difference in projections is highlighted by the following map:

Map and Dataset

The dataset used was `tgs00111`: Nights spent at tourist accommodation establishments by NUTS 2 regions (here the Metadata for consultation). The dataset covers internal tourism, in other words tourism flows within the country (domestic tourism) or from abroad to destinations in the country (inbound tourism) for the year 2022. For the sake of the exercise we further enhanced the dataset with two additional columns with the share of domestic and foreign tourist overnights (continuous scale), and one column for a factor variable based on the condition that if the domestic tourism share of overnight stay is greater than 50%, the tourist is labeled as "Domestic Tourist"; otherwise, they are labeled as "Foreign Tourist".

By plotting the latter variable, we were able to group together regions based on whether the majority of overnight stay in hotels were by domestic tourists or foreign tourists.

Not surprisingly, regions with famous "instagrammable" destinations like Istanbul or Antalya and Burdur had a majority of foreign tourists staying overnight in hotels, while the rest of the country registered a higher domestic tourists inflow, suggesting a limited international touristic network in Türkiye.

The plot for the continuous scale variable focused more on the domestic dimension of tourism and represented the share of domestic tourist overnight stays in different Turkish regions.

Besides stressing the point made by the previous map, this time we get more insights on the distribution of

Domestic tourists in Türkiye. To further enrich the plot, we also decided to highlight the names of those 3 regions with beyond 95% of domestic tourists overnight stays.

Storing visualizations

There are two conceptually different ways to store visualizations: raster-based and vector-based formats.

Raster-based formats like PNG and JPEG store images as grids of pixels, making them suitable for complex color gradients and detailed images. However, they are resolution-dependent, which means they can lose quality when scaled up.

On the other hand, vector-based formats such as SVG and PDF store image data using mathematical formulas to define shapes, lines, and colors. This makes them ideal for visualizations with geometric shapes, charts, maps, and illustrations where scalability and high-quality printing are crucial. Unlike raster formats, vector graphics are resolution-independent and can be scaled without loss of quality.

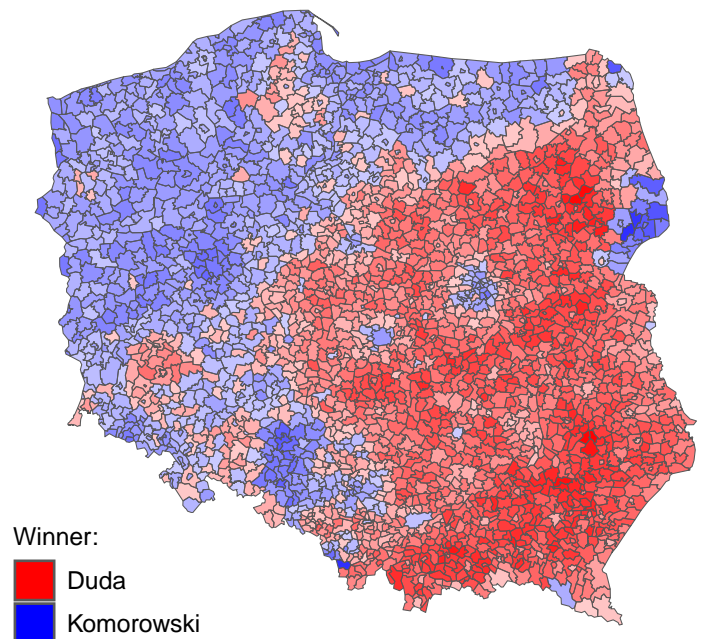
For visualizations created using R and ggplot2, which inherently produce vector graphics, it is recommended to save them in vector-based formats like SVG or PDF. These formats maintain sharpness and clarity when scaled to any size, making them suitable for presentations, printing, and high-resolution displays. SVG is particularly useful for web-based graphics and interactive visualizations, while PDF is excellent for high-quality printing and cross-platform compatibility.

Exercise D

Comparison of support for Komorowski and Duda

```
plot(PL_plot_1)
```

2015 Polish Presidential Election: Duda vs. Komorowski
Poland – Municipality Level



Source: PKW

The provided R code generates a thematic map representing the outcomes of the second round of the 2015 Polish presidential elections at the municipality level. The data is grouped based on whether the winning candidate in each municipality was Duda or Komorowski. The plot uses different shades of red or blue (representing Duda or Komorowski, respectively) to indicate varying levels of majority support for the respective candidate.

The transparency of the colors also varies to reflect the degree of majority/support for the winning candidate in each municipality. Darker shades indicate a higher level of majority/support, while lighter shades represent a lower level of majority/support. This additional dimension helps visually emphasize areas where the winning candidate had a more significant lead in terms of voter share.

Postal voting envelopes anomalies

Three types of anomalies are identified:

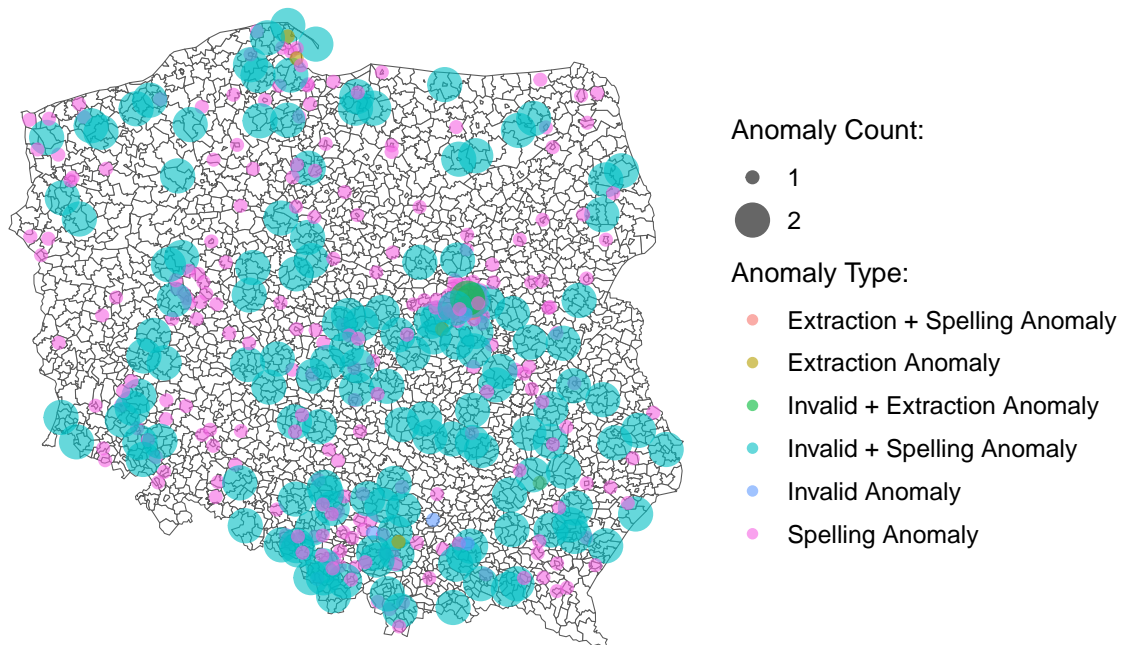
- **anomaly_invalid:** Checks if there are more invalid voting papers than postal voting envelopes received in either round of voting.
- **anomaly_spelling:** Identifies anomalies related to discrepancies in the number of invalid voting papers and specific errors on postal voting envelopes, such as missing declarations, signatures, voting envelopes, or signs of envelope opening.
- **anomaly_count:** Detects anomalies in the count of voting envelopes placed in the ballot box versus the number of voting papers taken from envelopes in either round of voting.

PL_plot_2

Warning: Using size for a discrete variable is not advised.

2015 Polish Presidential Election: anomalies in PVE

Poland – Municipality Level



Data source: PKW

Anomalies are represented as points (`geom_point()`) on the map, with each point's color indicating the type of anomaly and size representing the how many anomalies types were overall detected. In this way, one can

highlight areas where specific types of anomalies or their combinations were observed more frequently.

By visually exploring anomalies in the handling of voting materials, it appears that this was a widespread problem, involving several municipalities, although no clear pattern can be detected.

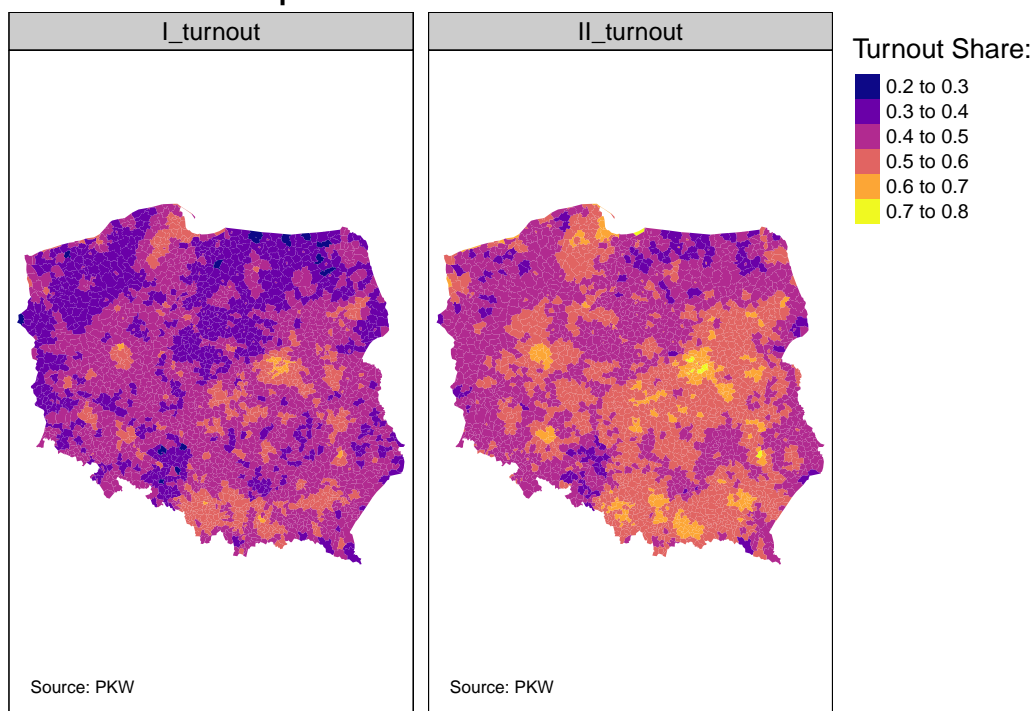
Notice that out of the three anomalies, the one about spelling is the “weak” one. Despite postal voting envelopes were return with a declaration, and a sealed voting envelope, the count of invalid PVE is still higher. It is not clear though, whether this is due to some further unrecorded issues (e.g. the misspelling of the candidate’s name, thus the label for this variable). Further material, information or context should be validate the theory.

Turnout for each election round

For the final visualization, employing `tm_shape()` function, Turnout share in the first and second round of the presidential election was plotted. We made use of the `tm_facets` specification to combine in a single plot both heatmaps.

PL_plot_3

Turnout Comparison between I and II round



Overall, it appears that in the second round there was a higher public involvement. In rural areas this was the less stronger than in the large urban areas, where participation peaked as far as reaching almost 80%.