Analysing GeoSocial-linked data in R

Matthew Curry & Tomasz Zając

Institute for Social Science Research

The University of Queensland

ARDC Summer School 2024 Day 3

Social Science Stream

Acknowledgment of **Country**

The University of Queensland (UQ) acknowledges the Traditional Owners and their custodianship of the lands on which we meet.

We pay our respects to their Ancestors and their descendants, who continue cultural and spiritual connections to Country.

We recognise their valuable contributions to Australian and global society.

Day 3 Part II

- Use simulated survey data with linked geosocial variables
- Use data manipulation methods from Day 1 to prepare data for analysis
- Regression analyses, predictions, and graphs in R

(Tentative) Schedule

Session 1 (~11:45 – 12:30)

- Outline of our analysis
- Read data into R
- Inspect data

Lunch (12:30 – 1:30)

Session 2 (1:30 – 3:00)

- Manipulate data set to create a data set for analysis
- Descriptive statistics
- Analyse data using regression techniques

Outline of Methodology

- Data
- Research Question
- Methods

Data

- Simulated survey data with 6,000 observations
- Longitudinal with 2 survey waves (2009 & 2019) linked to 2011 Census
- Respondents were aged 15 in 2009 and 25 in 2019
- Variables contain information on demographic and socioeconomic background from adolescence and educational and labour market outcomes in early adulthood
- Linked to geosocial indicators using the GeoSocial tool at the SA3 level

Research Question

- What are the effects of place-based geosocial indicators, socio-demographic background, and education on individual earnings?

- Outcomes:
  - Earnings
  - Poverty

- Covariates:
  - Geosocial variables
  - Demographic background
  - Socioeconomic background
  - Education

Methods

Preparation

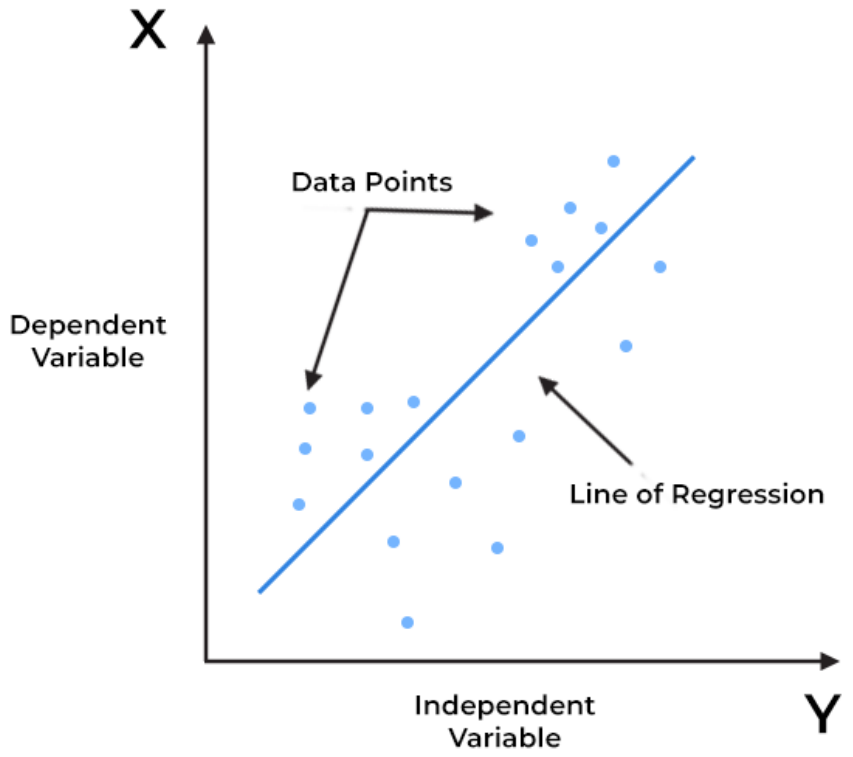
- Reading in a .csv data file — `readr::read_csv()`
- Exploring variables — `table(); summary(); hist()`
- Recoding variables — `dplyr::mutate()`
- Selecting sample and variables for analysis — `select(); filter()`
- Computing descriptive statistics — `psych::describe(); summarytools::freq()`

Analysis

- Linear regression (Ordinary Least Squares) — `lm()`
- Logistic regression — `glm(, family=binomial(link = “logit”))`

Ordinary Least Squares Regression

- OLS a method of summarising a relationship between 1 or more independent variables and a continuous outcome
- Residuals: distance between each data point and the regression line
- OLS draws a line that minimizes the sum of squares of the residuals
- $Y = a + bX_i + e_i$



OLS Regression in R

- Linear model in R: `lm()`
- `lm(outcome ~ xvar1 + xvar2 + xvar3, data = yourdata)`
- Save to an object: `myresults <- lm(outcome ~ xvar1 + xvar2 + xvar3, data = yourdata)`
- To view results:
  - `summary(myresults)`
  - Stargazer package: `stargazer(myresults, type="text")`
    - Can produce outputs in text, html, LaTeX

Logistic Regression

- For dichotomous outcomes, OLS regression can produce estimates that are impossible
  - e.g., predicted probabilities above 100% or below 0%.

- Logistic regression (aka: logit) models the log-odds of a dichotomous outcome, which bounds predicted outcomes to a range of 0 – 1

- Coefficient interpretation: A one-unit change in X is associated with a B increase in the log odds of Y.

- More informative to transform into odds or predicted probabilities for meaningful interpretation

Logistic Regression in R

- To run a logit in R, we use the `glm()` function while specifying the family and link

- ```
  myresults <- glm(outcome ~ xvar1 + xvar2 + xvar3,
                   family = binomial(link = "logit"), data=mydata)
  ```

- `summary(myresults)`
  - This will provide logit coefficients and standard errors in log-odds
  - Hard to make sense of the magnitude of effects (though can tell direction and statistical significance)

- `stargazer()` package can also return coefficients and standard errors
  ```

Odds Ratios

- Odds: pr(event) / pr(non-event)
- Transform into odds ratios by exponentiating logit coefficients
- Odds ratios are the multiplicative effect of a one-unit change in X on the odds of Y
- Odds ratios range from 0 to ∞, where 1 = no effect (“even odds”)
- O.R. < 1 are negative effects; O.R. > 1 are positive effects, expressed as a percentage change after subtracting 1
  - O.R. = 0.80: “A 20% decrease in the odds of Y”
  - O.R. = 1.65: “A 65% increase in the odds of Y”

Odds ratios in R

- Odds ratios after logit: `exp(cbind(OR = coef(myresults)))`
- `stargazer()` can return odds ratios and the test statistics to tell statistical significance
- `stargazer(myresults, apply.coef = exp, t.auto=F, p.auto=F, report = "vct*", type="text")`
  - `apply.coef = exp` exponentiated coefficients
  - `t.auto=F` and `p.auto=F` suppress the default standard errors and p-values
  - `report = "vct*"` reports the variables, coefficients, t-stats, and significance stars

Predicted probabilities

- Predicted probabilities are easiest to interpret conceptually
  - Can only be evaluated at specific values of the covariates
  - This is because changes are not additive as they are in OLS: a 1-unit change in X does not add a b-unit change in Y linearly
  - The effect of a 1-unit change in X on the predicted probability of Y depends on the values of X and other covariates

- Predicted probabilities can be calculated in R using `marginaleffects::predictions()`

Marginal Predictions

- After computing OLS or logistic regression coefficients, we can calculate predicted outcomes by evaluating the regression equations

- For OLS, these predictions are in the units of the outcome variable, e.g., $/week

- For logistic regression, these predictions are in probabilities of Y==1

Marginal predictions in R

- `marginaleffects` package, using the `predictions()` function
- There are several ways to estimate adjusted predictions. We will select values of independent variables of interest, then take the average predictions over the entire data set
- The `predictions()` function does this by copying the data for each selected value of the variable(s) we specify, then averaging the estimated outcomes/predicted probabilities

```
p_mod <- predictions(mod,
                     type = "response",
                     by = c("var1", "var2"),
                     newdata = datagrid(var1 = unique(data$var1),
                                        var2 = unique(data$var2),
                     grid_type = "counterfactual"))
```

R exercises (according to Dall-E image generator)