ECON0128 Assignment 2

Fayyaz Baig

2024-10-23

```
knitr::opts_chunk$set(echo = FALSE)
# Clear Environments and Load necessary libraries
rm(list=ls())
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(texreg)
```

```
## Version:  1.39.4
## Date:     2024-07-23
## Author:   Philip Leifeld (University of Manchester)
##
## Consider submitting praise using the praise or praise_interactive functions.
## Please cite the JSS article in your publications -- see citation("texreg").
##
## Attaching package: 'texreg'
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(DBI)
library(RPostgres)
```

Frisch Waugh Lovell Theorem

Data Loading

```
## 'data.frame':    3044 obs. of  12 variables:
##  $ county_fips: int  1001 1003 1005 1007 1009 1011 1013 1015 1017 1019 ...
##  $ state      : chr  "ALABAMA" "ALABAMA" "ALABAMA" "ALABAMA" ...
##  $ state_po   : chr  "AL" "AL" "AL" "AL" ...
##  $ county_name: chr  "AUTAUGA" "BALDWIN" "BARBOUR" "BIBB" ...
##  $ pop        : int  58805 231767 25223 22293 59134 10357 19051 116441 34772 24971 ...
##  $ un_rate    : num  2.9 2.9 4 3.2 2.8 3.9 3.9 3.7 3.1 3 ...
##  $ y_hh_med   : int  67565 71135 38866 50907 55203 33124 42268 50259 39318 50388 ...
##  $ south      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ white_share: num  0.738 0.832 0.455 0.744 0.868 ...
##  $ total_votes: int  27770 109679 10518 9595 27588 4613 9488 50983 15284 12301 ...
##  $ rep_votes  : int  19838 83544 5622 7525 24711 1146 5458 35101 8753 10583 ...
##  $ rep_share  : num  0.714 0.762 0.535 0.784 0.896 ...
```

```
##   county_fips   state state_po county_name    pop un_rate y_hh_med south
## 1        1001 ALABAMA       AL     AUTAUGA  58805     2.9    67565     1
## 2        1003 ALABAMA       AL     BALDWIN 231767     2.9    71135     1
## 3        1005 ALABAMA       AL     BARBOUR  25223     4.0    38866     1
## 4        1007 ALABAMA       AL        BIBB  22293     3.2    50907     1
## 5        1009 ALABAMA       AL      BLOUNT  59134     2.8    55203     1
## 6        1011 ALABAMA       AL     BULLOCK  10357     3.9    33124     1
##   white_share total_votes rep_votes rep_share
## 1   0.7377079       27770     19838 0.7143680
## 2   0.8320731      109679     83544 0.7617137
## 3   0.4551163       10518      5622 0.5345123
## 4   0.7440832        9595      7525 0.7842626
## 5   0.8677066       27588     24711 0.8957155
## 6   0.2099792        4613      1146 0.2484284
```

```
##  [1] "county_fips" "state"       "state_po"    "county_name" "pop"
##  [6] "un_rate"     "y_hh_med"    "south"       "white_share" "total_votes"
## [11] "rep_votes"   "rep_share"
```

Data Processing

We transform the variable y_hh_med into its logarithmic form.

Multiple Regression We perform a multiple regression of rep_share on log_y_hh_med and white_share.

```
##
## ========================
##               Model 1
## ------------------------
## (Intercept)      2.69 ***
##                 (0.11)
## log_y_hh_med    -0.22 ***
##                 (0.01)
## white_share      0.44 ***
##                 (0.01)
## ------------------------
## R^2              0.35
## Adj. R^2         0.35
## Num. obs.     3044
```

```
## ========================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

Application of Frisch Waugh Lovell (FWL) Theorem

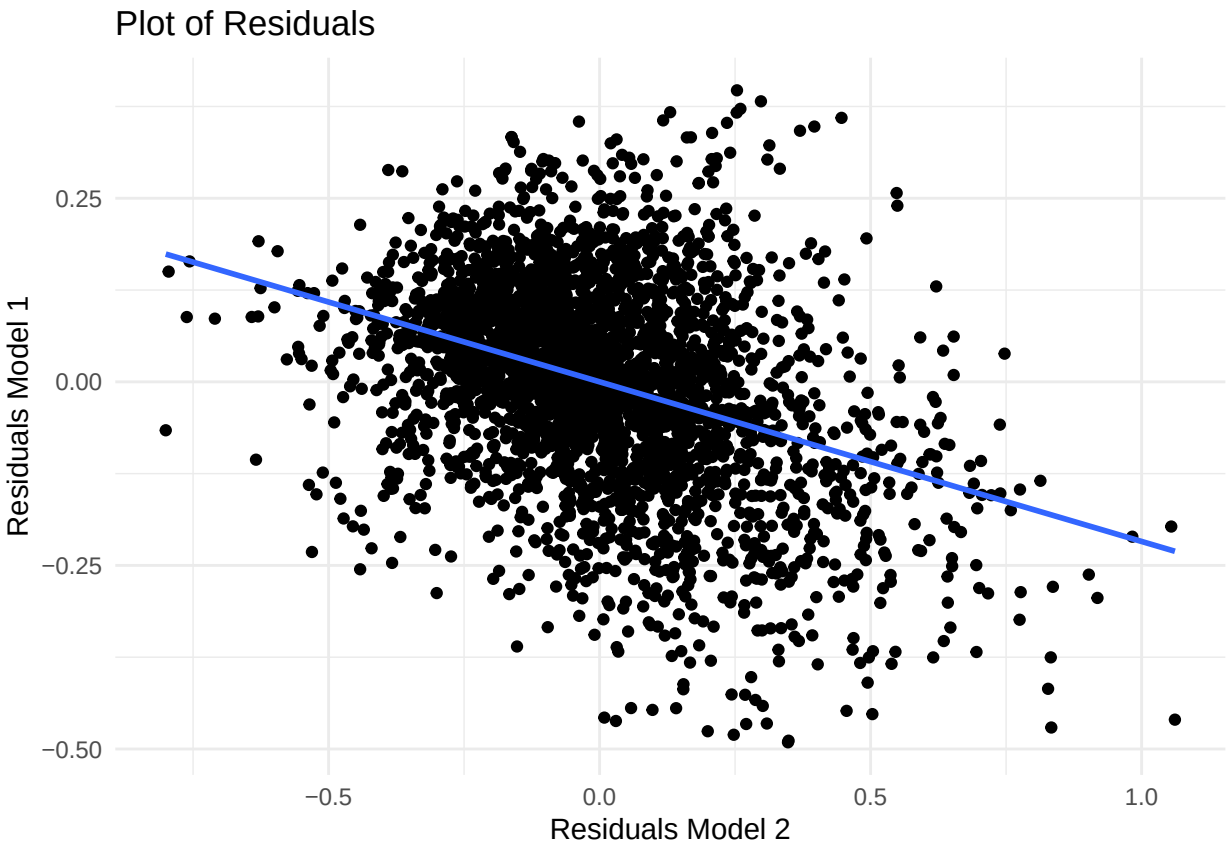
We demonstrate the FWL theorem by conducting residual regressions.

```
##
## ====================================
##               Model 1      Model 2
## ------------------------------------
## (Intercept)      2.69 ***    -0.00
##                 (0.11)       (0.00)
## log_y_hh_med    -0.22 ***
##                 (0.01)
## white_share      0.44 ***
##                 (0.01)
## res2                         -0.22 ***
##                              (0.01)
## ------------------------------------
## R^2              0.35         0.13
## Adj. R^2         0.35         0.13
## Num. obs.     3044         3044
## ====================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

Visualizing the FWL Theorem

We create a scatter plot of the residuals from the two regressions using ggplot2.

```
## `geom_smooth()` using formula = 'y ~ x'
```



The plot demonstrates the FWL theorem by showing the relationship between the two residuals. The slope should be equal to the coefficient of the log_y_hh_med variable in the original model.

Direct Regression of rep_share on res2

We regress rep_share on res2 to further confirm the FWL theorem.

```
##
## ================================================
##                Model 1      Model 2      Model 3
## ------------------------------------------------
## (Intercept)      2.69 ***    -0.00         0.65 ***
##                 (0.11)       (0.00)       (0.00)
## log_y_hh_med    -0.22 ***
##                 (0.01)
## white_share      0.44 ***
##                 (0.01)
## res2                         -0.22 ***    -0.22 ***
##                              (0.01)       (0.01)
## ------------------------------------------------
## R^2              0.35         0.13         0.10
## Adj. R^2         0.35         0.13         0.10
## Num. obs.     3044         3044         3044
## ================================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

Task 2: Database Interaction

Part 1: Setting Up and Connecting to Database First, we install and load the necessary packages to connect to a PostgreSQL database.

```
## [1] "movies"  "ratings" "tags"
```

```
## [1] "movie_id" "title"    "genres"
```

Part 2: Querying the Database

Method 1: SQL Queries

We retrieve all records from the ratings table where the rating is 5.

Method 2: Queries with dbplyr

We filter and arrange the ratings table using dbplyr.

Part 3: Working with Multiple Tables We rank the top 10 movies according to the number of ratings 4.5 and above in 2007.