

1. Organizing and Cleaning Data

a. Load Libraries

```
# Load the pacman package for efficient package management
library(pacman)

# Use pacman to load the required packages
p_load(
  haven, # For reading and writing data
  tidyverse, # A collection of packages for data manipulation,
visualization,
  # and more
  prodest, # For productivity and efficiency estimation
  estprod, # For estimating production functions
  plm, # For panel data econometrics
  huxtable, # For creating and formatting tables
  stargazer, # For creating LaTeX, HTML, and ASCII tables
  vtable, # For creating variable tables
  lmtest, # For conducting statistical tests
  hrbrthemes, # Additional themes for ggplot2
  viridis, # Color palettes for visualization
  modelsummary,
  vtable
)
```

b. Load data set

The dataset is in .dta format, which I received from a professor. One of the reasons I kept the data in this format is because .dta files preserve metadata (variable labels, value labels, and formats). This was particularly useful since the original data has 400 variables, and I will only be using a subset in my analysis. Keeping the data in .dta format makes it easier to sort and filter the relevant variables. Additionally, I have the corresponding documentation for this dataset.

```
# load 2009-2013 Enterprise Survey data
original_data <- read_dta("C:\\Users\\Aayush\\Documents\\files prior to
1-7-2024\\Nepal_2009_2013.dta")
```

c. Clean

The *original_data* has 968 observations. I filtered out firms according to the needs of my analysis. The data is filled with variables, but the variables do not really make sense, so I used the documentation and column labels to rename them. There are also columns with values less than 0, indicating that the firms either refused to answer or that the question does not apply to them. Some of the variables, such as *Informal*, are dummies, but they are coded as 1 and 2 instead of 1 and 0, so I had to recode them. The variable *yofop*, which is the total number of years the firms have existed, is not present in the data, so I created it by calculating the difference between the survey year and the firm's establishment year. I also took the log of sales, capital, labor, and intermediate inputs because the *Levinsohn and Petrin* method of calculating productivity requires these variables to be in log form. I adjusted any monetary values for inflation using the inflation rate for Nepal from 2009 to 2013 from official World Bank data. Lastly, I filtered out only manufacturing firms and retained those that have data for both years, thus creating a balanced panel dataset. *manufacturing_firms_panel_data* is left with 158 firms. I will also explain some key variables being used in the analysis.

- *sales*, In Last Fiscal Year, What Were This Establishment'S Total Annual Sales?
- *capital*, Cost For Establishment To Re-Purchase All Of Its Machinery
- *labor*, Total Labor Cost (Incl. Wages, Salaries, Bonuses, Etc) In Last Fiscal Year
- *interm*, Cost Of Raw Materials And Intermediate Goods Used In Prod. In Last Fiscal Year
- *Informal*, Does This Establishment Compete Against Unregistered Or Informal Firms?
- *Experience*, How Many Years Of Experience Working In This Sector Does The Top Manager Have?
- *export*, percentage of sales as exports
- *Credit*, Establishment has A Line Of Credit Or Loan From A Financial Institution?
- *One_Product*, First Product/Service Percent of Total Annual Sales
- *Foreign_tech*, Do You Use Technology Licensed From A Foreign-Owned Company?
- *Tax_burden*, How Much Of An Obstacle is Tax Rates?(scale 0-4)

```

manufacturing_firms_panel_data <- original_data %>%
  # Select necessary columns for data analysis
  select(
    year, a0, id2009, d2, n7a, n2a, n2e, e11, b7, k8, a6b, j30c, j30a, l1,
    b5, l4a,
    b7, d3c, e6, b2b, c30a, e1, d3b, d3c, d1a3
  ) %>%
  # Filter rows with values greater than or equal to 0
  filter(if_all(
    c(
      d2, n7a, n2a, n2e, e11, b7, k8, a6b, j30c, j30a, l1, b5, l4a, b7, d3c,
      e6,
      b2b, c30a, e1
    ),
    ~ . >= 0
  )) %>%
  # Rename columns for easier reference
  rename(
    sales = d2, capital = n7a, labor = n2a, interm = n2e, ID = id2009,
    Informal = e11, Experience = b7, Credit = k8, Size = l1, Foreigntech =
    e6,
    Bussiness_permit = j30c, Tax_burden = j30a, local = e1,
    one_product = d1a3
  ) %>%
  # Convert Informal variable to a dummy variable
  mutate(Informal = ifelse(Informal == 2, 0, 1)) %>%
  # Add a column for the number of years of operation
  mutate(yofop = ifelse(
    year == 2009, 2009 - b5,
    ifelse(year == 2013, 2013 - b5, NA)
  )) %>%
  # Determine if the firm sells only one product
  mutate(one_product = ifelse(one_product < 100, 0, 1)) %>%
  # Calculate total export percentage as of sales
  mutate(export = d3b + d3c) %>%
  # Take the natural log of certain columns
  mutate(across(c(sales, capital, labor, interm), ~ log(.))) %>%
  # Adjust monetary values for inflation
  mutate(across(
    c(sales, capital, labor, interm),
    ~ ifelse(year == 2013, (. / 142.52) * 100, .)
  )) %>%
  # Convert other variables to dummy variables
  mutate(local = ifelse(local == 1, 1, 0)) %>%

```

```
mutate(Credit = ifelse(Credit == 1, 1, 0)) %>%
mutate(Foreigntech = ifelse(Foreigntech == 1, 1, 0)) %>%
# Select only manufacturing firms
filter(a0 == 1) %>%
# Select only rows valid for a balanced panel
group_by(ID) %>%
filter(all(c(2009, 2013) %in% year)) %>%
ungroup()

# Display the first few rows of the modified dataset
head(manufacturing_firms_panel_data)
```

	Experience	Credit	a6b	Bussiness_permit	Tax_burden	Size	b5	l4a	d3c	Fore
0	10	1	3	0	2	200	1995	100	0	
0	15	1	2	2	2	74	1995	17	10	
0	20	1	3	0	2	450	1994	15	100	
1	15	1	3	0	2	215	1994	100	60	
0	3	1	3	0	2	250	1959	225	0	
0	20	1	3	0	1	250	1959	20	40	

2. Calculating Total Factor Productivity (TFP)

In order to calculate Total Factor Productivity, I employ two methods: the Levinsohn-Petrin model and the Olley-Pakes model. The Levinsohn-Petrin method uses intermediate inputs (like materials) to control for unobserved productivity shocks, while the Olley-Pakes method uses investment as a proxy for these shocks in estimating production functions. The Levinsohn-Petrin method is advantageous when investment data is noisy or sparse, as it relies on intermediate inputs, which are often more consistently reported. On the other hand, the Olley-Pakes method is better suited when investment is a reliable and responsive proxy for productivity shocks, providing a clearer estimation of production functions. Since I have data for the cost of intermediate goods, I have decided to use the Levinsohn-Petrin method for my thesis.

```
# Estimating productivity using the Levinsohn-Petrin model
levinsohn_model <- levinsohn_petrin(
  data = manufacturing_firms_panel_data,
  formula = sales ~ labor | capital | interm,
```

```

    id = "ID",
    time = "year",
    bootstrap = TRUE
)

# Estimating productivity using the Olley-Pakes model
olleypakes_model <- olley_pakes(
  data = manufacturing_firms_panel_data,
  formula = sales ~ labor | capital | interm,
  id = "ID",
  time = "year",
  bootstrap = TRUE
)

# Summarize the results of the Levinsohn-Petrin model
summary(levinsohn_model)

## Call
## levinsohn_petrin(data = manufacturing_firms_panel_data, formula = sales ~
##   labor | capital | interm, id = "ID", time = "year", bootstrap = TRUE)
##
## Coefficients
##           Estimate Std. Error z value Pr(>|z|)
## labor      0.4624765  0.0306328 15.0974 < 2.2e-16 ***
## capital -0.0070391  0.0021649 -3.2514  0.001148 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## #Bootstrapped standard errors.

# Summarize the results of the olleypakes model
summary(olleypakes_model)

## Olley-Pakes Production Function Estimator
##
## Call
## olley_pakes(data = manufacturing_firms_panel_data, formula = sales ~
##   labor | capital | interm, id = "ID", time = "year", bootstrap = TRUE)
##
## Coefficients
##           Estimate Std. Error z value Pr(>|z|)
## labor      0.462477  0.070739  6.5378 6.244e-11 ***
## capital 0.028244  0.087703  0.3220  0.7474

```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## #Bootstrapped standard errors.

# Filter and compute total factor productivity (TFP) using the Olley-Pakes
model
manufacturing_firms_panel_data2 <- manufacturing_firms_panel_data %>%
  # Calculate log TFP based on the Olley-Pakes model coefficients
  mutate(logtfp = sales -
    (levinsohn_model$t0[1] * labor) -
    (levinsohn_model$t0[2] * capital) -
    interm) %>%
  # Scale the log TFP values (standardization)
  mutate(avetfp = scale(logtfp))
```

3. Calculate summary statistics

There are many ways to represent summary tables, or tables in general, in R. However, my personal favorite is using the *datasummary* package. It integrates well with the tidyverse and supports output in various formats such as LaTeX, HTML, and huxtable. I also create a correlation table using the *datasummary_correlation* function in the *datasummary* package.

```
# Remove the labels (if they are stored as attributes)
attr(manufacturing_firms_panel_data2$Size, "label") <- NULL
attr(manufacturing_firms_panel_data2$Experience, "label") <- NULL

# Select specific columns from the dataset for the summary table
selected_data <- manufacturing_firms_panel_data2 %>%
  select(
    Informal, Size, Experience, export, Credit, one_product,
    Foreigntech, Tax_burden
  )

# Convert all columns in selected_data to numeric
selected_data[] <- lapply(selected_data, function(x)
  as.numeric(as.character(x)))

# Generate summary statistics for the selected columns
datasummary_skim(selected_data,
  output = "huxtable",
```

```
title = " Summary Statistics"
)
```

	Unique	Missing Pct.	Mean	SD	Min	Median	Max
Informal	2	0	0.3	0.5	0.0	0.0	1.0
Size	59	0	52.4	76.8	4.0	22.0	450.0
Experience	37	0	17.9	10.0	1.0	16.0	43.0
export	15	0	10.0	23.3	0.0	0.0	100.0
Credit	2	0	0.5	0.5	0.0	1.0	1.0
one_product	2	0	0.5	0.5	0.0	1.0	1.0
Foreigntech	2	0	0.1	0.2	0.0	0.0	1.0
Tax_burden	5	0	1.3	1.3	0.0	1.0	4.0

```
# Calculate the correlation matrix for specified variables
datasummary_correlation(selected_data,
  output = "huxtable",
  title = "Correlation matrix"
)
```

Table 1: Correlation matrix

	Informal	Size	Experience	export	Credit	one_product	Foreigntech	Tax_burden
Informal	1
Size	-.13	1
Experience	.02	-.16	1
export	-.10	.32	.08	1
Credit	.03	.19	-.10	.03	1	.	.	.
one_product	-.25	.08	.05	.01	-.06	1	.	.
Foreigntech	-.12	.13	.02	.00	-.15	.13	1	.
Tax_burden	-.03	.16	-.07	-.01	-.03	-.10	.13	1

4. Data Visualization

Let's check relationship between my independent and dependent variables. I plot a scatter plot and a linear line between all the controls and the dependent variable in order to see the relationships and correlations between them, assess the strength and direction of these associations, and identify any potential outliers or trends that could influence the results of the analysis. I also plot histograms of all the variables to visualize their distributions, identify skewness or kurtosis, and detect any potential anomalies or outliers that may need to be addressed in the analysis. Lastly, I plot heterogeneity across time to examine how the relationships between variables may vary over different time periods, identify any temporal patterns or trends, and assess the stability or changes in the effects of the variables over time.

```
# Reshape dataframe to long format for plotting
data4_long <- tidyr::pivot_longer(
  manufacturing_firms_panel_data2,
  cols = c(
    "Informal", "Experience", "Credit", "Size",
    "Foreigntech", "Tax_burden", "Bussiness_permit", "local"
  )
)

# Scatterplot of avetfp against each independent variable
ggplot(data4_long, aes(x = value, y = avetfp)) +

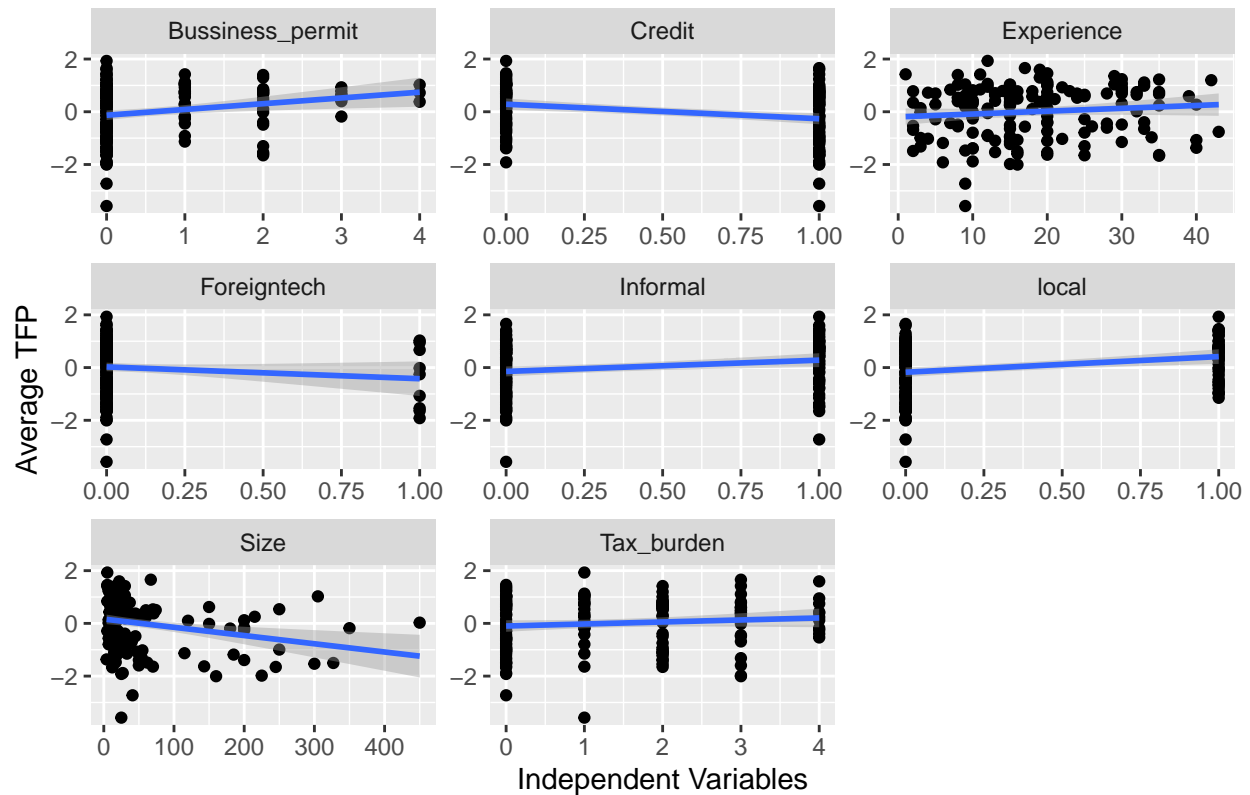
  # Add scatter points
  geom_point() +

  # Add linear fit line
  geom_smooth(method = lm) +

  # Facet plot by variable
  facet_wrap(~name, scales = "free") +

  # Add title and axis labels
  labs(
    title = "Scatterplots of avetfp against Independent Variables",
    x = "Independent Variables", y = "Average TFP"
  )
```


Scatterplots of avetfp against Independent Variables



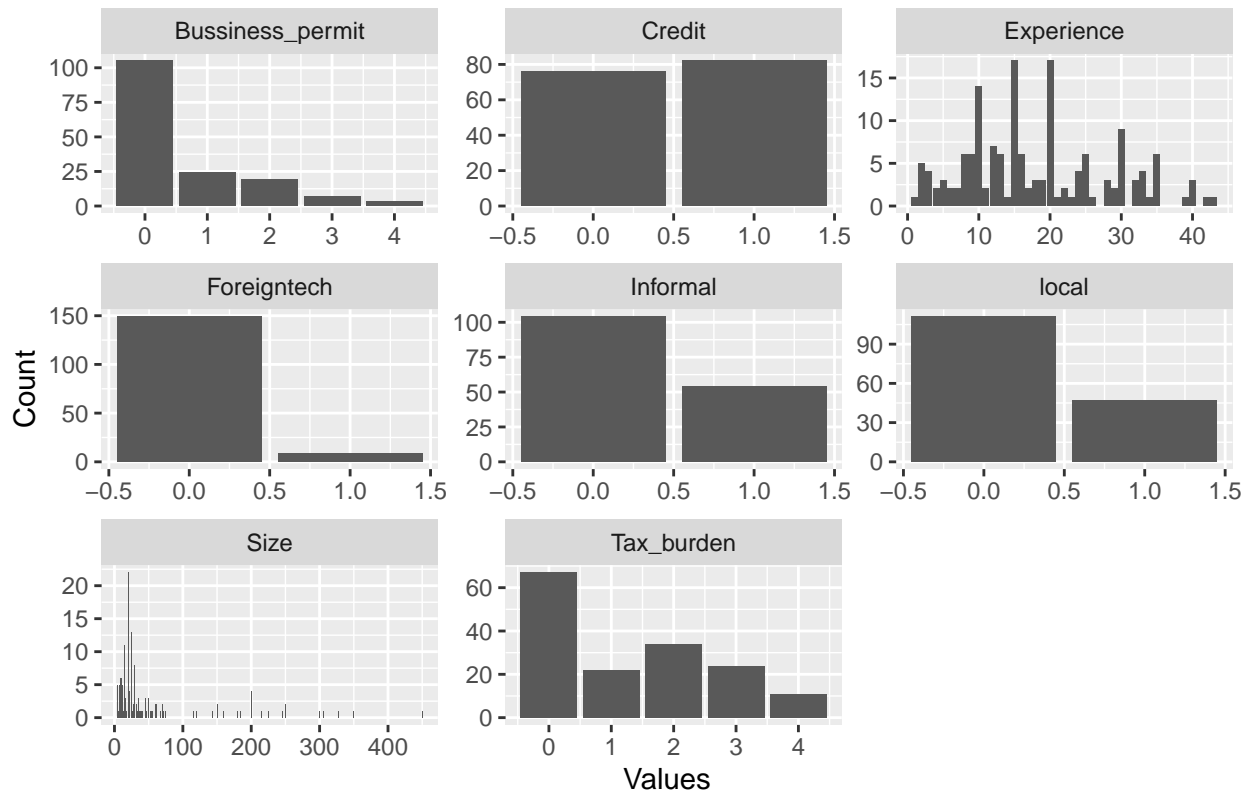
```
# Histogram of values for each independent variable
ggplot(data4_long, aes(x = value)) +
```

```
# Add bar plot
geom_bar() +
```

```
# Facet plot by variable
facet_wrap(~name, scales = "free") +
```

```
# Add title and axis labels
labs(title = "Histograms of Independent Variables", x = "Values", y =
"Count")
```

Histograms of Independent Variables

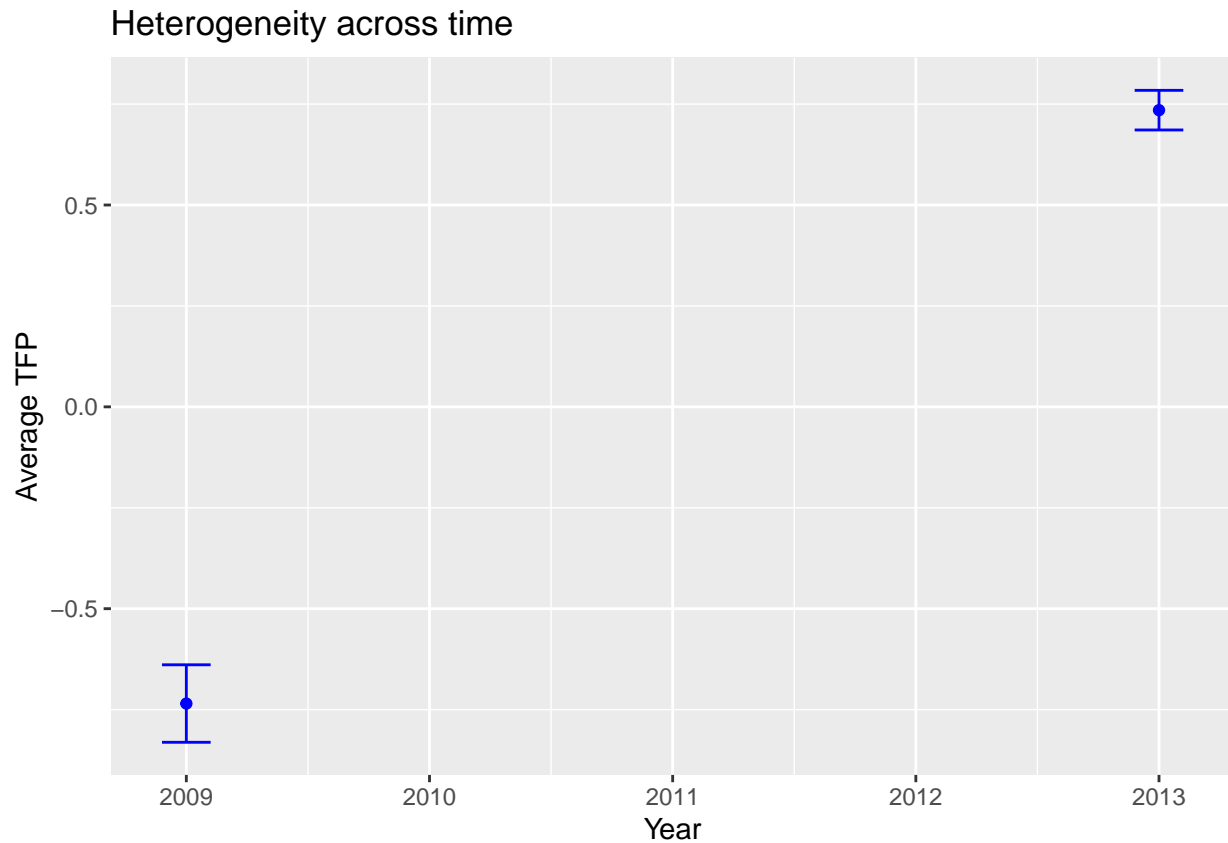


```
# Plot to observe heterogeneity of TFP over time
ggplot(manufacturing_firms_panel_data2, aes(x = year, y = avetfp)) +

  # Plot the mean TFP for each year
  stat_summary(fun = mean, geom = "point", color = "blue") +

  # Add error bars representing standard error around the mean
  geom_errorbar(
    stat = "summary", fun.data = "mean_se",
    color = "blue", width = 0.2
  ) +

  # Add title and axis labels
  labs(title = "Heterogeneity across time", x = "Year", y = "Average TFP")
```



5. Regression Models

I am going to run a few different regression models for my panel data because this approach allows me to compare results across different specifications, account for various potential biases, and ensure the robustness of my findings by evaluating how different models handle the data.

a. OLS Model

$$\ln tfp_{it} = \beta_0 + \beta_1 Informal_{it} + \gamma X_{it} + \varepsilon_{it} \quad (1)$$

Where:

$\ln tfp_{it}$ is the dependent variable for firm i at year t ,

$Informal_{it}$ is the main independent variable for firm i at year t ,

X_{it} is the set of controls that include exper, credit, size, tech, tax, permit, and local,

ε_{it} is the error term.

```
data4 <- manufacturing_firms_panel_data2
```

```
# Fit OLS regression models with increasing complexity
ols_model1 <- lm(avetfp ~ Informal, data = data4)

ols_model2 <- lm(avetfp ~ Informal + log(Size), data = data4)

ols_model3 <- lm(avetfp ~ Informal + log(Size) + Experience, data = data4)

ols_model4 <- lm(avetfp ~ Informal + log(Size) + Experience + export, data =
data4)

ols_model5 <- lm(avetfp ~ Informal + log(Size) + Experience + export +
Credit, data = data4)

ols_model6 <- lm(avetfp ~ Informal + log(Size) + Experience + export +
Credit + one_product,
  data = data4
)

ols_model7 <- lm(avetfp ~ Informal + log(Size) + Experience + export +
Credit + one_product +
  Foreigntech, data = data4)

ols_model8 <- lm(avetfp ~ Informal + log(Size) + Experience + export +
Credit + one_product +
  Foreigntech + Tax_burden, data = data4)

# Store OLS models in a list
ols_models <- list(ols_model1, ols_model2, ols_model3, ols_model4,
ols_model5, ols_model6, ols_model7, ols_model8)

modelsummary(ols_models,
  output = "huxtable",
  fmt = fmt_significant(2),
  stars = TRUE,
  gof_map = c("r.squared", "adj.r.squared", "nobs", "F"),
  title = "OLS model"
)
```

Table 2: OLS model

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(Intercept)	-0.146 (0.096)	0.82** (0.27)	0.67* (0.32)	0.72* (0.33)	0.75* (0.32)	0.92** (0.33)	0.90** (0.33)	0.82* (0.32)
Informal	0.43* (0.16)	0.33* (0.16)	0.33* (0.16)	0.34* (0.16)	0.37* (0.16)	0.29+ (0.16)	0.28+ (0.16)	0.28+ (0.16)
log(Size)		-0.282*** (0.074)	-0.273*** (0.075)	-0.291*** (0.079)	-0.23** (0.08)	-0.222** (0.079)	-0.207* (0.081)	-0.247** (0.082)
Experience			0.0067 (0.0076)	0.0060 (0.0077)	0.0048 (0.0075)	0.0057 (0.0075)	0.0059 (0.0075)	0.0065 (0.0074)
export				0.0025 (0.0034)	0.0022 (0.0033)	0.0020 (0.0033)	0.0018 (0.0033)	0.0023 (0.0033)
Credit					-0.42** (0.15)	-0.44** (0.15)	-0.47** (0.16)	-0.44** (0.15)
one_product						-0.33* (0.15)	-0.32* (0.15)	-0.28+ (0.15)
Foreigntech							-0.31 (0.33)	-0.37 (0.32)
Tax_burden								0.119* (0.056)
R2	0.041	0.123	0.127	0.130	0.171	0.197	0.202	0.225
R2 Adj.	0.035	0.111	0.110	0.108	0.144	0.165	0.165	0.183
Num.Obs.	158	158	158	158	158	158	158	158
F	6.749	10.846	7.481	5.734	6.279	6.180	5.420	5.410
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001								

b. Fixed effect without time and entity effect

$$\ln tfp_{it} = \beta_0 + \beta_1 Informal_{it} + \gamma X_{it} + \alpha_i + \varepsilon_{it} \quad (2)$$

Where:

$\ln tfp_{it}$ is the dependent variable for firm i at year t ,

$Informal_{it}$ is the main independent variable for firm i at year t ,

X_{it} is the set of controls that include exper, credit, size, tech, tax, permit, and local,

α_i is the fixed effect for firm i ,

ε_{it} is the error term.

```
# Convert data to a panel data object with ID and year as indices
panel_data <- pdata.frame(data4, index = c("ID", "year"))

# Fit fixed effects models with increasing complexity
fixed_model1 <- plm(avetfp ~ Informal, data = panel_data, model = "within")

fixed_model2 <- plm(avetfp ~ Informal + local, data = panel_data, model =
"within")

fixed_model3 <- plm(avetfp ~ Informal + local + Experience,
  data = panel_data,
  model = "within"
)

fixed_model4 <- plm(avetfp ~ Informal + local + Experience + Credit,
  data = panel_data,
  model = "within"
)

fixed_model5 <- plm(avetfp ~ Informal + local + Experience + Credit +
log(Size),
  data = panel_data, model = "within"
)

fixed_model6 <- plm(avetfp ~ Informal + local + Experience + Credit +
log(Size) +
  Foreigntech + export, data = panel_data, model = "within")

fixed_model7 <- plm(avetfp ~ Informal + local + Experience + Credit +
log(Size) +
  Foreigntech + export, data = panel_data, model = "within")
```

```
fixed_model8 <- plm(
  avetfp ~ Informal + local + Experience + Credit + log(Size) +
    Foreigntech + export + Tax_burden,
  data = panel_data,
  model = "within", effect = "twoways"
)

# Store fixed effects models in a list
fixed_models <- list(fixed_model1, fixed_model2, fixed_model3, fixed_model4,
fixed_model5, fixed_model6, fixed_model7, fixed_model8)

# Create a summary table for fixed effects models
modelsummary(fixed_models,
  output = "huxtable",
  fmt = fmt_significant(2),
  stars = TRUE,
  gof_map = c("r.squared", "adj.r.squared", "nobs", "F"),
  title = "Fixed effect without time and entity effect"
)
```

Table 3: Fixed effect without time and entity effect

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Informal	0.69*	0.79*	0.81**	0.77*	0.77*	0.74*	0.74*	-0.086
	(0.30)	(0.30)	(0.30)	(0.30)	(0.30)	(0.30)	(0.30)	(0.173)
local		0.57+	0.69*	0.66+	0.66+	0.65+	0.65+	0.088
		(0.33)	(0.34)	(0.33)	(0.33)	(0.34)	(0.34)	(0.187)
Experience			0.026	0.025	0.024	0.030+	0.030+	-0.0033
			(0.017)	(0.017)	(0.017)	(0.018)	(0.018)	(0.0100)
Credit				-0.47	-0.48+	-0.45	-0.45	-0.027
				(0.28)	(0.28)	(0.30)	(0.30)	(0.166)
log(Size)					-0.20	-0.13	-0.13	-0.011
					(0.33)	(0.33)	(0.33)	(0.180)
Foreigntech						-0.38	-0.38	-0.26
						(0.63)	(0.63)	(0.35)
export						-0.015	-0.015	0.0076
						(0.015)	(0.015)	(0.0085)
Tax_burden								-0.039
								(0.069)
R2	0.064	0.099	0.126	0.157	0.161	0.176	0.176	0.035
R2 Adj.	-0.883	-0.836	-0.805	-0.765	-0.780	-0.797	-0.797	-1.164
Num.Obs.	158	158	158	158	158	158	158	158
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001								

c. Fixed effect without time effect

$$\ln tfp_{it} = \beta_0 + \beta_1 Informal_{it} + \gamma X_{it} + \alpha_i + \delta_t + \varepsilon_{it} \quad (3)$$

Where:

$\ln tfp_{it}$ is the dependent variable for firm i at year t ,

$Informal_{it}$ is the main independent variable for firm i at year t ,

X_{it} is the set of controls that include exper, credit, size, tech, tax, permit, and local,

α_i is the fixed effect for firm i ,

δ_t is the time effect for year t ,

ε_{it} is the error term.

```
# Convert data to a panel data object with ID and year as indices
panel_data <- pdata.frame(data4, index = c("ID", "year"))

# Fit fixed effects models with a time effect and increasing complexity
fixed_model1 <- plm(avetfp ~ Informal,
  data = panel_data,
  model = "within", effect = "time"
)

fixed_model2 <- plm(avetfp ~ Informal + Experience,
  data = panel_data,
  model = "within", effect = "time"
)

fixed_model3 <- plm(avetfp ~ Informal + Experience + Credit,
  data = panel_data, model = "within", effect = "time"
)

fixed_model4 <- plm(avetfp ~ Informal + Experience + Credit + Size,
  data = panel_data, model = "within", effect = "time"
)

fixed_model5 <- plm(avetfp ~ Informal + Experience + Credit + Size +
  Foreigntech,
  data = panel_data, model = "within", effect = "time"
)

fixed_model6 <- plm(avetfp ~ Informal + Experience + Credit + Size +
  Foreigntech
  + Tax_burden, data = panel_data, model = "within", effect = "time")
```

```
fixed_model7 <- plm(
  avetfp ~ Informal + Experience + Credit + Size + Foreigntech +
    Tax_burden + Bussiness_permit,
  data = panel_data,
  model = "within", effect = "time"
)

fixed_model8 <- plm(
  avetfp ~ Informal + Experience + Credit + Size + Foreigntech +
    Tax_burden + Bussiness_permit + local,
  data = panel_data,
  model = "within", effect = "time"
)

# Store all fixed effects models in a list
fixed_models <- list(
  fixed_model1, fixed_model2, fixed_model3, fixed_model4,
  fixed_model5, fixed_model6, fixed_model7, fixed_model8
)

# Create and format a summary table for fixed effects models with time effect
modelsummary(fixed_models,
  output = "huxtable",
  fmt = fmt_significant(2),
  stars = TRUE,
  gof_map = c("r.squared", "adj.r.squared", "nobs", "F"),
  title = "Fixed effect with time effect"
) %>%
  set_width(1)
```

Table 4: Fixed effect with time effect

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Informal	0.10 (0.12)	0.10 (0.12)	0.12 (0.11)	0.066 (0.110)	0.04 (0.11)	0.034 (0.110)	0.04 (0.11)	0.026 (0.108)
Experience		0.0048 (0.0054)	0.0033 (0.0053)	0.00056 (0.00517)	0.0009 (0.0051)	0.00059 (0.00512)	0.00032 (0.00518)	0.0026 (0.0051)
Credit			-0.34** (0.11)	-0.26* (0.10)	-0.3** (0.1)	-0.3** (0.1)	-0.30** (0.11)	-0.27** (0.10)
Size				0.00250** (0.00069)	0.00226** (0.00068)	0.00218** (0.00069)	0.0022** (0.0007)	0.00173* (0.00069)
Foreigntech					-0.52* (0.22)	-0.50* (0.22)	-0.49* (0.22)	-0.41+ (0.22)
Tax_burden						-0.033 (0.040)	-0.03 (0.04)	-0.026 (0.039)
Bussiness_permit							-0.024 (0.057)	-0.0056 (0.0564)
local								0.34** (0.11)
R2	0.005	0.010	0.070	0.144	0.174	0.178	0.179	0.226
R2 Adj.	-0.008	-0.009	0.045	0.116	0.142	0.140	0.135	0.179
Num.Obs.	158	158	158	158	158	158	158	158
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001								

d. Fixed effect with time and entity effect

$$\ln tfp_{it} = \beta_0 + \beta_1 Informal_{it} + \gamma X_{it} + \alpha_i + \delta_t + \varepsilon_{it} \quad (4)$$

Where:

$\ln tfp_{it}$ is the dependent variable for firm i at year t ,

$Informal_{it}$ is the main independent variable for firm i at year t ,

X_{it} is the set of controls that include exper, credit, size, tech, tax, permit, and local,

α_i is the fixed effect for firm i ,

δ_t is the time effect for year t ,

ε_{it} is the error term.

```
# Convert data to a panel data object with ID and year as indices
panel_data <- pdata.frame(data4, index = c("ID", "year"))

# Fit fixed effects models with both time and industry effects
fixed_model1 <- plm(avetfp ~ Informal, data = panel_data, model = "within")

fixed_model2 <- plm(avetfp ~ Informal + Experience,
  data = panel_data,
  model = "within", effect = "twoways"
)

fixed_model3 <- plm(avetfp ~ Informal + Experience + Credit,
  data = panel_data,
  model = "within", effect = "twoways"
)

fixed_model4 <- plm(avetfp ~ Informal + Experience + Credit + Size,
  data = panel_data,
  model = "within", effect = "twoways"
)

fixed_model5 <- plm(avetfp ~ Informal + Experience + Credit + Size +
  Foreigntech,
  data = panel_data, model = "within", effect = "twoways"
)

fixed_model6 <- plm(
  avetfp ~ Informal + Experience + Credit + Size + Foreigntech +
  Tax_burden,
  data = panel_data, model = "within",
  effect = "twoways"
```

```
)

fixed_model7 <- plm(
  avetfp ~ Informal + Experience + Credit + Size + Foreigntech
  + Tax_burden + Bussiness_permit,
  data = panel_data,
  model = "within", effect = "twoways"
)

fixed_model8 <- plm(
  avetfp ~ Informal + Experience + Credit + Size + Foreigntech +
  Tax_burden + Bussiness_permit + local,
  data = panel_data, model = "within", effect = "twoways"
)

# Store all fixed effects models in a list
fixed_models <- list(
  fixed_model1, fixed_model2, fixed_model3, fixed_model4,
  fixed_model5, fixed_model6, fixed_model7, fixed_model8
)

# Create and format a summary table for fixed effects models with time
# and industry effects
modelsummary(fixed_models,
  output = "huxtable",
  fmt = fmt_significant(2),
  stars = TRUE,
  gof_map = c("r.squared", "adj.r.squared", "nobs", "F"),
  title = "Fixed effect with time and entity effect"
)
```

Table 5: Fixed effect with time and entity effect

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Informal	0.69*	-0.10	-0.10	-0.096	-0.097	-0.10	-0.096	-0.077
	(0.30)	(0.16)	(0.16)	(0.164)	(0.164)	(0.17)	(0.166)	(0.173)
Experience		-0.0020	-0.0020	-0.0011	-0.00014	-0.00078	-0.001	0.00021
		(0.0087)	(0.0087)	(0.0088)	(0.00884)	(0.00897)	(0.009)	(0.00952)
Credit			0.0069	0.03	-0.00036	0.017	0.02	0.018
			(0.1542)	(0.16)	(0.15887)	(0.163)	(0.16)	(0.165)
Size				0.0020	0.0022	0.0022	0.0022	0.0022
				(0.0021)	(0.0021)	(0.0021)	(0.0021)	(0.0022)
Foreigntech					-0.34	-0.29	-0.29	-0.30
					(0.33)	(0.34)	(0.35)	(0.35)
Tax_burden						-0.035	-0.032	-0.030
						(0.066)	(0.067)	(0.068)
Bussiness_permit							-0.031	-0.025
							(0.077)	(0.079)
local								0.085
								(0.188)
R2	0.064	0.006	0.006	0.018	0.032	0.036	0.038	0.040
R2 Adj.	-0.883	-1.054	-1.081	-1.084	-1.082	-1.103	-1.128	-1.152
Num.Obs.	158	158	158	158	158	158	158	158
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001								

e. Random Effect

$$\ln tfp_{it} = \beta_0 + \beta_1 \text{Informal}_{it} + \gamma X_{it} + \alpha_i + \varepsilon_{it} \quad (5)$$

Where:

$\ln tfp_{it}$ is the dependent variable for firm i at year t ,

Informal_{it} is the main independent variable for firm i at year t ,

X_{it} is the set of controls that include exper, credit, size, tech, tax, permit and local

α_i is the random effect for firm i ,

ε_{it} is the error term.

```
# Create a panel data object with "ID" and "year" as panel identifiers
panel_data <- pdata.frame(data4, index = c("ID", "year"))

# Run random effects models with increasing complexity
random_model1 <- plm(avetfp ~ Informal, data = panel_data, model = "random")

random_model2 <- plm(avetfp ~ Informal + log(Size), data = panel_data, model = "random")

random_model3 <- plm(avetfp ~ Informal + log(Size) + Experience,
  data = panel_data, model = "random"
)

random_model4 <- plm(avetfp ~ Informal + log(Size) + Experience + export,
  data = panel_data, model = "random"
)

random_model5 <- plm(avetfp ~ Informal + log(Size) + Experience + export +
  Credit,
  data = panel_data, model = "random"
)

random_model6 <- plm(avetfp ~ Informal + log(Size) + Experience + export +
  Credit +
  one_product, data = panel_data, model = "random")

random_model7 <- plm(avetfp ~ Informal + log(Size) + Experience + export +
  Credit +
  one_product + Foreigntech, data = panel_data, model = "random")

random_model8 <- plm(
  avetfp ~ Informal + log(Size) + Experience + export + Credit +
```

```
    one_product + Foreigntech + Tax_burden,  
    data = panel_data,  
    model = "random"  
)  
  
# Store random effects models in a list  
random_models <- list(  
  random_model1, random_model2, random_model3, random_model4,  
  random_model5, random_model6, random_model7, random_model8  
)  
  
# Generate and format a table for the random effects models  
modelsummary(random_models,  
  output = "huxtable",  
  fmt = fmt_significant(2),  
  stars = TRUE,  
  gof_map = c("r.squared", "adj.r.squared", "nobs", "F"),  
  title = "Random effect model"  
)
```


Table 6: Random effect model

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(Intercept)	-0.146 (0.096)	0.82** (0.27)	0.67* (0.32)	0.72* (0.33)	0.75* (0.32)	0.92** (0.33)	0.90** (0.33)	0.82* (0.32)
Informal	0.43* (0.16)	0.33* (0.16)	0.33* (0.16)	0.34* (0.16)	0.37* (0.16)	0.29+ (0.16)	0.28+ (0.16)	0.28+ (0.16)
log(Size)		-0.282*** (0.074)	-0.273*** (0.075)	-0.291*** (0.079)	-0.23** (0.08)	-0.222** (0.079)	-0.207* (0.081)	-0.247** (0.082)
Experience			0.0067 (0.0076)	0.0060 (0.0077)	0.0048 (0.0075)	0.0057 (0.0075)	0.0059 (0.0075)	0.0065 (0.0074)
export				0.0025 (0.0034)	0.0022 (0.0033)	0.0020 (0.0033)	0.0018 (0.0033)	0.0023 (0.0033)
Credit					-0.42** (0.15)	-0.44** (0.15)	-0.47** (0.16)	-0.44** (0.15)
one_product						-0.33* (0.15)	-0.32* (0.15)	-0.28+ (0.15)
Foreigntech							-0.31 (0.33)	-0.37 (0.32)
Tax_burden								0.119* (0.056)
R2	0.041	0.123	0.127	0.130	0.171	0.197	0.202	0.225
R2 Adj.	0.035	0.111	0.110	0.108	0.144	0.165	0.165	0.183
Num.Obs.	158	158	158	158	158	158	158	158
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001								

6. Robutness check

```
# Extract and display the fixed effects (cross-section-wise intercepts)
fixef(fixed_model8)
```

```
## 1011401033 1011401036 1011401037 1011401040 1011401041 1020101072 1020501011
## -1.451061 -1.292682 -1.571440 -1.160112 -0.742615 -0.913429 -0.828185
## 2061001004 2071301016 2071901026 2071901034 2080601056 2080610011 2083106002
## -1.278590 -0.428613 -0.003504 0.378669 -0.116113 -0.479142 0.276094
## 2083203006 2089801001 2089801009 2089801010 2089801011 2089801013 2089801014
## -0.457135 -0.560024 -1.203391 -0.520424 -1.002188 -0.117627 -1.292947
## 2089801016 2089801019 2089801021 2089801023 2089801024 2089801025 2089801027
## -0.514614 -1.197767 -0.635797 -1.071043 -0.704411 -0.342706 -0.504031
## 2089801028 2089801036 2089801037 2089801040 2089801042 2089801050 2089801057
## -0.255148 -1.012610 -1.028974 -0.873733 -1.051132 -0.976269 -0.827953
## 2089801062 2089801074 2089801076 2091001039 2091001085 2099001050 2099801018
## -0.909529 -1.012384 -1.140615 -0.587463 0.561749 -0.023397 -0.685523
## 2099801030 2099801036 2099801047 2099801049 2099801054 2099801058 2099801062
## -1.449394 -0.513606 0.156350 -1.240897 -1.012308 0.176082 -0.211988
## 2099801079 2139801064 2139801066 2139801071 2139801072 2139801073 3109801003
## -0.650104 -2.169489 -1.242342 -2.007540 -1.967621 -1.306084 -0.522495
## 3109801010 3109801011 3109801013 3109801019 3109801023 3110301056 3121401031
## -0.893606 -0.016174 -0.735261 -0.668986 -0.753699 -2.190109 -0.702719
## 3129801002 3129801003 3129801006 3129801012 3131101049 3131101088 3131701066
## -1.021530 -2.029213 -0.601941 -1.171058 -0.351587 -0.621037 -0.557012
## 3139801001 3139801004 3139801014 3139801035 3139801036 3139801037 3139801040
## -0.705524 -2.183658 -1.994557 -0.486867 -0.470434 -0.754048 -0.931196
## 3139801041 3139801049
## -1.766860 -0.061580
```

```
# Perform a Hausman test to compare Fixed Effects model with OLS model
pFtest(fixed_model8, ols_model8)
```

```
##
## F test for twoways effects
##
## data: avetfp ~ Informal + Experience + Credit + Size + Foreigntech + ...
## F = 3.2668, df1 = 79, df2 = 70, p-value = 5.119e-07
## alternative hypothesis: significant effects
```

```
# Perform a Breusch-Pagan test to check for two-way fixed effects
plmtest(fixed_model8, effect = "twoways", type = "bp")
```

```
##
## Lagrange Multiplier Test - two-ways effects (Breusch-Pagan)
##
## data: avetfp ~ Informal + Experience + Credit + Size + Foreigntech + ...
## chisq = 876.73, df = 2, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

```
# Perform Hausman test to compare fixed and random effects models
phtest(fixed_model8, random_model8)
```

```
##
## Hausman Test
##
## data: avetfp ~ Informal + Experience + Credit + Size + Foreigntech + ...
## chisq = 205.98, df = 5, p-value < 2.2e-16
## alternative hypothesis: one model is inconsistent
```

```
# Apply heteroscedasticity-robust standard errors to the random effects model
robust_model <- coeftest(random_model8, vcov = vcovHC(random_model8, type = "HCO"))
```

```
# Display the results with robust standard errors
robust_model
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.8237635  0.2623346   3.1401 0.0020369 **
## Informal     0.2823840  0.1247170   2.2642 0.0250050 *
## log(Size)    -0.2474139  0.0733777  -3.3718 0.0009515 ***
## Experience    0.0064502  0.0064290   1.0033 0.3173469
## export       0.0023021  0.0020914   1.1007 0.2727988
## Credit       -0.4408246  0.1416061  -3.1130 0.0022207 **
## one_product  -0.2778329  0.1460978  -1.9017 0.0591417 .
## Foreigntech  -0.3709806  0.3747356  -0.9900 0.3237891
## Tax_burden    0.1192318  0.0521435   2.2866 0.0236274 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```