An informative title for your assignment

FEM11149 - Introduction to Data Science

Name 1 (xx%), Name 2 (xx%)
Name 3 (xx%), Name 4 (xx%)

Month, 2024

Introduction

Economic growth is a central concern for policymakers, researchers, and international organizations, as it reflects a country's capacity to expand production, improve living standards, and sustain long-term development. Forecasts of Gross Domestic Product (GDP) growth are particularly important, as they inform fiscal policy, trade negotiations, and borrowing decisions. For example, governments may finance higher expenditure through debt, but the sustainability of such debt ultimately depends on the economy's future growth potential. Understanding the determinants of GDP growth is therefore essential for designing effective policies and ensuring financial stability.

This raises the central research question: **To what extent can economic and demographic variables, such as trade balance, unemployment, and population characteristics, be used to predict GDP growth?** Investigating this question is significant because it clarifies the structural factors that shape economic performance and provides a foundation for more accurate forecasting. A better understanding of these relationships enables policymakers to anticipate risks, develop sustainable fiscal strategies, and allocate resources more efficiently. Crucially, it may also help mitigate resource misallocation, where funds are directed toward ineffective or low-impact policies rather than those that foster long-term growth.

Data

The analysis is based on the data obtained from the *World Bank's Global Jobs Indicators Database* and *Balance of Payments statistics*. These sources provide internationally comparable information on a wide range of economic and demographic variables that are relevant for understanding growth dynamics. The dataset used covers 150 countries and includes 73 variables capturing different aspects of economic activity, labor market conditions, trade, and demographic structure. The key variable of interest is GDP growth, which serves as the measure of economic performance. The data includes indicators such as trade balance, unemployment, net trade in goods and services, labor force participation, and population characteristics. Most of these are expressed as percentages or ratios, allowing for straightforward interpretation and comparison across countries. For example, the trade balance is measured as a share of GDP, while unemployment is expressed as a percentage of the labor force. This standardized reporting ensures consistency across countries and reduces the need for additional transformations before analysis.

Simple descriptive statistics illustrate the variation across economies. Average GDP growth is around 2.6%, though individual countries range from sharp contractions of over –34% to expansions above 13%. Unemployment averages 8.1%, with some countries experiencing very low rates (close to 0.2%) while others face levels above 27%. Population growth is generally positive, averaging 1.4%, but ranges from slight declines (–1.9%) to rapid expansions exceeding 5% per year.

By combining macroeconomic indicators with demographic and labor market measures, the dataset provides a broad perspective on the potential drivers of GDP growth. Its cross-country coverage enables the identification of structural patterns and relationships that can inform both policy and strategic decision-making. While the dataset does not capture every possible determinant of growth, it offers a sufficiently comprehensive view to support meaningful insights into the economic and demographic factors most closely associated with growth outcomes.

Methods

Regression analysis is a widely used statistical technique for examining the relationship between dependent variable and one or more explanatory variables. In its simplest form, linear regression estimates how changes in the explanatory variables are associated with changes in the outcome. This is particularly useful in economic contexts, where the goal is often to assess the impact of multiple structural and demographic factors on a country's performance. The strength of regression analysis lies in its ability to quantify these relationships and to provide a framework for prediction. The standard approach, Ordinary Least Squares (OLS), estimates coefficients by minimizing the squared differences between observed and predicted values. OLS performs well when there are only a few explanatory variables that are relatively independent. However, when dealing with a large set of predictors or when many of them are highly correlated, several of the classical assumptions underlying OLS may be violated.

OLS relies on assumptions such as linearity of relationships, independence of errors, homoscedasticity (constant variance of errors), and the absence of perfect multicollinearity among explanatory variables. In practice, economic and demographic data often challenge these assumptions. For example, variables such as unemployment, labor participation, and population growth may be correlated, leading to multicollinearity that inflates standard errors and undermines the reliability of coefficient estimates. Similarly, the assumption of homoscedasticity may be violated if countries with different income levels exhibit systematically different variances in growth rates. Violations of these assumptions do not necessarily invalidate the model, but they do weaken its interpretability and predictive accuracy, motivating the use of alternative approaches such as regularization methods.

LASSO regression (Least Absolute Shrinkage and Selection Operator) enhances prediction by introducing a penalty on the absolute size of the coefficients. Instead of fitting all available predictors, LASSO shrinks some coefficients toward zero and can even set them exactly to zero. This property makes it particularly valuable for variable selection, as it automatically highlights the most important predictors while discarding those that contribute little to the model. Mathematically, LASSO minimizes the sum of squared errors plus a penalty term, expressed as:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

where λ controls the strength of the penalty. Larger values of λ produce simpler models with fewer predictors, while smaller values allow more variables to remain in the model.

Elastic Net regression combines features of both LASSO and Ridge regression. Ridge regression introduces a penalty on the squared size of coefficients, which shrinks them toward zero but never eliminates them entirely. This makes Ridge useful when dealing with many correlated predictors, as it distributes the effect across them rather than excluding variables. While LASSO excels at variable selection, it can behave inconsistently when predictors are highly correlated, often choosing one variable at random and discarding others. Elastic Net addresses this by blending the LASSO penalty with the Ridge penalty (which shrinks coefficients without setting them to zero). Its objective function is:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \left[\alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2 \right]$$

Here α determines the balance between the LASSO ($\alpha = 1$) and Ridge ($\alpha = 0$) penalties. This flexibility makes Elastic Net particularly effective in settings with many correlated predictors, as it can select groups of variables together while still maintaining shrinkage for stability.

For model diagnostics, the key concern is predictive power rather than simply achieving the best in-sample fit. While the coefficient of determination (R^2) provides an indication of how much of the variation in the dependent variable is explained, a high R^2 does not necessarily translate into strong predictive accuracy. To avoid overfitting, penalized regressions rely on cross-validation, which evaluates how well the model generalizes to unseen data. The performance of each model is typically assessed using a loss function, most commonly the mean squared error (MSE) or the mean absolute error (MAE). Based on this evaluation, two values of the penalty parameter λ are often reported: λ_{min}, which minimizes the chosen prediction error, and λ_{1se}, which selects a simpler model whose error is within one standard error of the minimum.

While λ_{min} achieves the lowest estimated prediction error, it can sometimes result in overfitting by tailoring the model too closely to the specific folds of the cross-validation procedure. This can reduce stability and limit the model's ability to generalize to new data, effectively capturing noise as well as signal. The "1-SE rule" mitigates this risk by selecting the largest λ within one standard error of the minimum error. This approach yields a simpler and more parsimonious model, trading a very small loss in accuracy for improved robustness and generalizability. In practice, the stability of cross-validation also depends on how the folds are generated. Setting a random seed (e.g., using `set.seed()` in R) ensures that results are reproducible, since different random splits of the data can otherwise lead to slightly different values of λ_{min} and λ_{1se}. Using different seeds can therefore yield different models, especially when predictors are highly correlated, which highlights the importance of interpreting the results with caution and relying on methods that improve stability, such as Elastic Net. For this reason, results should not be viewed as exact, but rather as indicators of broader economic patterns.

Together, LASSO and Elastic Net provide a powerful framework for analyzing high-dimensional economic data. LASSO highlights the most relevant drivers of the dependent variable, while Elastic Net improves stability when predictors are correlated. By applying these methods, it becomes possible to balance interpretability and predictive accuracy, ensuring that the resulting models are both theoretically informative and practically useful.

Results

To assess the determinants of GDP growth, we estimated an OLS regression model using key economic and demographic variables that we deemed relevant through economic theory and statistical methods (like stepwise and variance inflation factors), with and without the inclusion of net trade in goods and services. Table 5 (in Appendix A) presents the results of these two specifications. The comparison shows that adding net trade does not improve the model: the coefficient on net trade is effectively zero and not statistically significant, and the explanatory power of the model even falls slightly when it is included (Adjusted R^2 decreases from 0.137 to 0.131). This suggests that net trade does not contribute additional explanatory value for GDP growth in this dataset. Instead, factors such as life expectancy, employment, and GDP per capita remain the more relevant predictors of growth.

Adding nonlinear and interaction terms does affect the model - refer to table 6 in Appendix A - to a limited extent. The adjusted R^2 increases slightly compared to the baseline specification (from 0.131 without nonlinearities to 0.137 with them), indicating a modest improvement in explanatory power. Importantly, the squared GDP per capita term becomes statistically significant, suggesting a convergence effect: countries with higher income levels grow more slowly, but the nonlinear term implies diminishing returns as economies develop. Similarly, the export value index and life expectancy remain strong predictors of growth.

On the other hand, the interaction term between unemployment and labor force participation does not reach significance, nor does squared population growth, meaning they add little explanatory value. The residual standard error and overall fit remain close to the baseline model, which shows that while nonlinear terms enrich the interpretation of certain variables (especially GDP per capita), they do not drastically improve model performance.

Overall, the models perform reasonably well, but the diagnostic plots highlight some issues that should be taken into account. The residuals are generally centered around zero, yet a few unusual observations appear as outliers, suggesting that some countries may exert more influence on the results than others. Among the diagnostics, the Scale–Location plot (Figure X) is particularly important because it shows that the variance of the errors changes with fitted values, a sign of heteroskedasticity. This means the model explains some parts of the data more consistently than others, which can make the reported standard errors less reliable. To address this, we rely on robust standard errors and check for influential points to ensure that no single country drives the findings. With these adjustments, the results remain reliable, but they should be interpreted as indicating general patterns rather than precise predictions. **need to include the plot and make it proper**

To test the robustness of the models, the models were re-estimated the model using LASSO, the results showed that only a small number of variables consistently matter for explaining GDP growth. At a moderate penalty, LASSO kept factors like employment, life expectancy, unemployment, exports, and GDP per capita, while shrinking most of the other variables to zero. This means those extra variables don't add much predictive power. When the penalty is made stronger, the model drops everything and reduces to just the average growth rate, showing that the data do not strongly support keeping many predictors. In comparison to the OLS model, which included every variable, LASSO highlights the ones that really drive the results and removes the noise. In practice, this makes the model simpler, less prone to overfitting, and more focused on the variables that actually matter.

Standardizing predictors before applying LASSO is important because the penalty is applied to the scale of the coefficients. When variables are standardized, the changes are on a comparable scale, so the penalty is lower and the model can retain meaningful predictors. Without standardization, variables measured in larger units are penalized more heavily, which can cause all coefficients to shrink to zero regardless of their predictive power. As shown in Table 1, without standardization being applied, most coefficients go to zero, while with standardization some predictors remain in the model. This demonstrates that standardization ensures the penalty works fairly across variables.

While the LASSO results are informative, they were estimated on a reduced set of predictors to illustrate the role of variable selection and standardization. To further improve upon these baseline models, we now employ the full dataset and introduce additional forms of regularization. In particular, ridge regression and elastic net allow us to balance shrinkage and selection in different ways, potentially yielding models that are both more stable and more accurate in prediction.

Cross-validation identified an optimal penalty of .min = 0.1827, which minimizes prediction error, and a more restrictive value of .1se = 2, which favors a simpler model. At .min, the LASSO retains a broad set of predictors, including demographic variables such as adolescent fertility rates, population structure, and life expectancy, along with labor market indicators like employment in industry, labor force participation, and unemployment. Institutional and economic measures, such as the export value index, tax-related indicators, and credit information, also remain in the model. These results suggest that a wide range of socioeconomic factors can contribute to explaining GDP growth when the model is allowed more flexibility. However, at .1se, the penalty is stronger, and all coefficients shrink to zero. This sharp reduction illustrates why relying solely on .min can sometimes lead to overfitting: the model captures more noise in an attempt to minimize error. The 1-SE rule mitigates this risk by favoring a more parsimonious model that sacrifices a small amount of predictive accuracy in exchange for greater robustness. Economically, this highlights that although factors like human capital (life expectancy, employment patterns) and trade (exports) appear relevant under weaker penalization, their effects are not strong or consistent enough to survive stricter regularization in our dataset.

For Elastic Net, cross-validation selected a mixing parameter of = 0.4, which indicates that the best-performing model combines both LASSO (variable selection) and Ridge (shrinkage) elements, with a slightly stronger tilt toward Ridge. The optimal penalty values were .min = 0.3163, minimizing the mean squared error, and .1se = 2.2320, which yields a more parsimonious model. Compared to LASSO, Elastic Net provides greater stability by shrinking correlated predictors together rather than arbitrarily selecting only one of them. This is particularly relevant in a dataset like ours, where many economic and demographic indicators are strongly correlated (e.g., labor market variables, population measures). In practice, the results suggest that while LASSO highlights individual variables as potential drivers of GDP growth, Elastic Net balances this by keeping groups of related predictors, making it less sensitive to data noise and more reliable for generalization. Thus, Elastic Net serves as a compromise between the sparsity of LASSO and the stability of Ridge regression, offering a more robust framework for capturing the complex relationships in the data.

Table 1: Optimal tuning parameters for LASSO and Elastic Net

Method	Metric	Value
LASSO	λ.min	0.1827
LASSO	λ.1se	2.0000
Elastic Net	α	0.4000
Elastic Net	λ.min	0.3163
Elastic Net	λ.1se	2.2320

Table 2: Test Performance (RMSE, MAE, R^2, SD_y)

Model	RMSE	MAE	R2	SD_y
Baseline (mean of train)	6.570	2.925	-0.053	6.486
Lasso (lambda.min)	6.584	3.002	-0.057	6.486
Lasso (lambda.1se)	6.570	2.925	-0.053	6.486
Elastic Net =0.4 (lambda.min)	**6.552**	**3.057**	**-0.047**	**6.486**
Elastic Net =0.4 (lambda.1se)	6.570	2.925	-0.053	6.486

At first glance, it might seem enough to run a regular regression and simply remove the variables that don't look significant. But this approach has some important problems. Standard regression struggles when variables are highly related to each other, or when there are many variables compared to the number of observations. In those cases, the estimates can jump around a lot depending on small changes in the data, which makes the model unreliable.

Penalized regression solves this by adding a "penalty" that discourages the model from becoming too complex. This keeps the estimates more stable and helps the model focus on the variables that really matter. It also avoids the risk of overfitting, which can happen if we keep testing and removing variables until only a few remain — that kind of trial-and-error tends to fit the current dataset too closely, but performs poorly when applied to new data.

Another key advantage is that penalized regression can automatically deal with situations where traditional regression would fail entirely — for example, when there are more variables than countries in the dataset, or when variables are so correlated that standard regression cannot separate their effects. In these situations, methods like LASSO or Elastic Net still produce workable, interpretable models.

In short, penalized regression is not just a more complicated way of doing regression — it's a safer and more reliable approach that helps us avoid misleading conclusions. Instead of giving us a model that only looks good on paper, it provides results that are more likely to hold up in practice, which is essential for making sound policy and business decisions.

Additionally, we wanted to check whether some countries tend to grow faster than most others. To make that precise, we defined a binary target **Growing more** (1 if annual GDP growth $>$ **2.7%**, 0 otherwise) and fit logistic regression with a **Ridge** penalty, comparing the cross-validated choices λ_{min} and λ_{1se}. As reported in **Table 3**, the λ_{1se} model outperforms λ_{min} on the test set—higher **Accuracy** (0.525 vs 0.475) and, more importantly, lower **LogLoss** (0.6871 vs 0.8054) and **Brier** (0.2471 vs 0.2774), with **AUC** also improving (0.563 vs 0.464).

Table 3: Ridge (logistic) — Test Metrics

Model	Accuracy	LogLoss	Brier	AUC
Ridge (lambda.min)	0.475	0.8054	0.2774	0.464
Ridge (lambda.1se)	0.525	0.6871	0.2471	0.563

When we repeat the analysis with a different train–test split (**Table 4**), performance shifts: in **Run 1**, λ_{min} yields better **LogLoss**, **Brier**, and **AUC** (0.6553, 0.2321, 0.619), while λ_{1se} slightly improves **Accuracy** (0.575 vs 0.525). In **Run 2**, the pattern returns to what we saw in **Table 3**, with λ_{1se} leading across metrics.

This split-to-split variability—visible in **Table 4**—is exactly what we would expect with a modest sample and a threshold near many observations. Averaging the two runs in **Table 4**, λ_{1se} offers better **Accuracy** and calibration (lower **LogLoss/Brier**), whereas λ_{min} retains a slight **AUC** advantage. Taken together with **Table 3**, we favor the more regularized λ_{1se} model for its stability and better-calibrated probabilities.

If we could add data, we would first prioritize **more countries (rows)** to reduce variance and stabilize results; then we would add **relevant** predictors (institutions, investment, demographics, trade exposure, external shocks) to capture more of the true drivers. Predicting **continuous GDP growth** is preferable when magnitudes matter for budgeting and scenario analysis, whereas predicting **Growing more** is more useful for threshold-based decisions where an actionable probability—"will this country exceed 2.7%?"—is exactly what is needed.

Table 4: Ridge (logistic) — Test Metrics (different seeds)

	Run 1				Run 2			
Model	Accuracy	LogLoss	Brier	AUC	Accuracy	LogLoss	Brier	AUC
Ridge (lambda.min)	0.525	0.6553	0.2321	0.619	0.475	0.8054	0.2774	0.464
Ridge (lambda.1se)	0.575	0.6821	0.2445	0.500	0.525	0.6871	0.2471	0.563

Appendix

Appendix A

Table 7: Comparison of LASSO Coefficients With and Without Standardization

	No.Standardization	Standardization
(Intercept)	2.609816	-6.626631
`Fertility rate, total (births per woman)`	0.000000	0.000000
`Employers, total (% of total employment) (modeled ILO estimate)`	0.000000	-0.398982
`Labor force participation rate, total (% of total population ages 15+) (modeled ILO estimate)`	0.000000	0.002144
`Life expectancy at birth, male (years)`	0.000000	0.164192
`Unemployment, total (% of total labor force) (modeled ILO estimate)`	0.000000	-0.061668
`Export value index (2000 = 100)`	0.000000	0.001245
`Population growth (annual %)`	0.000000	0.000000
`Population growth (annual %)^2`	0.000000	0.000000
`GDP per capita, PPP (constant 2011 international $)`	0.000000	-0.000045

	No.Standardization	Standardization
GDP per capita, PPP (constant 2011 international $)^2	0.000000	0.000000
Net trade in goods and services (BoP, current US$)	0.000000	0.000000
Wage and salaried workers, total (% of total employment) (modeled ILO estimate)	0.000000	-0.000340
Unemp x LFP	0.000000	0.000000

Table 8: Variable selection across LASSO and Elastic Net

Variable	L_min	L_1se	ENet_min	ENet_1se
(Intercept)	T	T	T	T
Adolescent fertility rate (births per 1,000 women ages 15-19)	T	F	T	F
Contributing family workers, female (% of female employment) (modeled ILO estimate)	T	F	T	F
Depth of credit information index (0=low to 8=high)	T	F	T	F
Employers, male (% of male employment) (modeled ILO estimate)	T	F	T	F
Employment in industry, male (% of male employment) (modeled ILO estimate)	T	F	T	F
Export value index (2000 = 100)	T	F	T	F
GDP per capita, PPP (constant 2011 international $)	T	F	T	F
Labor force participation rate, female (% of female population ages 15+) (modeled ILO estimate)	T	F	T	F
Labor force participation rate, male (% of male population ages 15+) (modeled ILO estimate)	T	F	T	F
Life expectancy at birth, male (years)	T	F	T	F
Own-account workers, total (% of male employment) (modeled ILO estimate)	T	F	T	F
Population ages 15-64, total	T	F	T	F
Population ages 65 and above (% of total)	T	F	T	F
Rural population	T	F	T	F
Tax payments (number)	T	F	T	F
Time required to start a business (days)	T	F	T	F
Time to prepare and pay taxes (hours)	T	F	T	F
Unemployment, male (% of male labor force) (modeled ILO estimate)	T	F	T	F
Unemployment, youth male (% of male labor force ages 15-24) (modeled ILO estimate)	T	F	T	F
Access to electricity (% of population)	F	F	T	F
Contributing family workers, total (% of total employment) (modeled ILO estimate)	F	F	T	F
Employers, female (% of female employment) (modeled ILO estimate)	F	F	T	F
Employment in industry (% of total employment) (modeled ILO estimate)	F	F	T	F
Export volume index (2000 = 100)	F	F	T	F
Labor force, total	F	F	T	F
Own-account workers, male (% of male employment) (modeled ILO estimate)	F	F	T	F
Time required to enforce a contract (days)	F	F	T	F

Table 5: Comparison of OLS Models: With vs. Without Net Trade

	Dependent variable:	
	GDP growth (annual %)	
	With Net Trade	Without Net Trade
	(1)	(2)
Fertility rate	−0.140	−0.136
	(0.643)	(0.639)
Employers (%)	−0.477***	−0.477***
	(0.123)	(0.122)
Labor force part. (%)	0.010	0.010
	(0.040)	(0.040)
Life expectancy (male)	0.261***	0.262***
	(0.093)	(0.093)
Unemployment (%)	−0.066	−0.066
	(0.064)	(0.064)
Export value index	0.002*	0.002*
	(0.001)	(0.001)
Population growth (%)	0.202	0.196
	(0.479)	(0.474)
GDP per capita (PPP)	−0.0001**	−0.0001**
	(0.00003)	(0.00003)
Net trade (BoP, US$)	0.000	
	(0.000)	
Wage & salaried workers (%)	−0.010	−0.010
	(0.026)	(0.026)
Intercept	−12.514	−12.574
	(8.524)	(8.475)
Observations	150	150
R^2	0.189	0.189
Adjusted R^2	0.131	0.137
Residual Std. Error	4.250 (df = 139)	4.235 (df = 140)
F Statistic	3.242*** (df = 10; 139)	3.627*** (df = 9; 140)
Note:		*p<0.1; **p<0.05; ***p<0.01

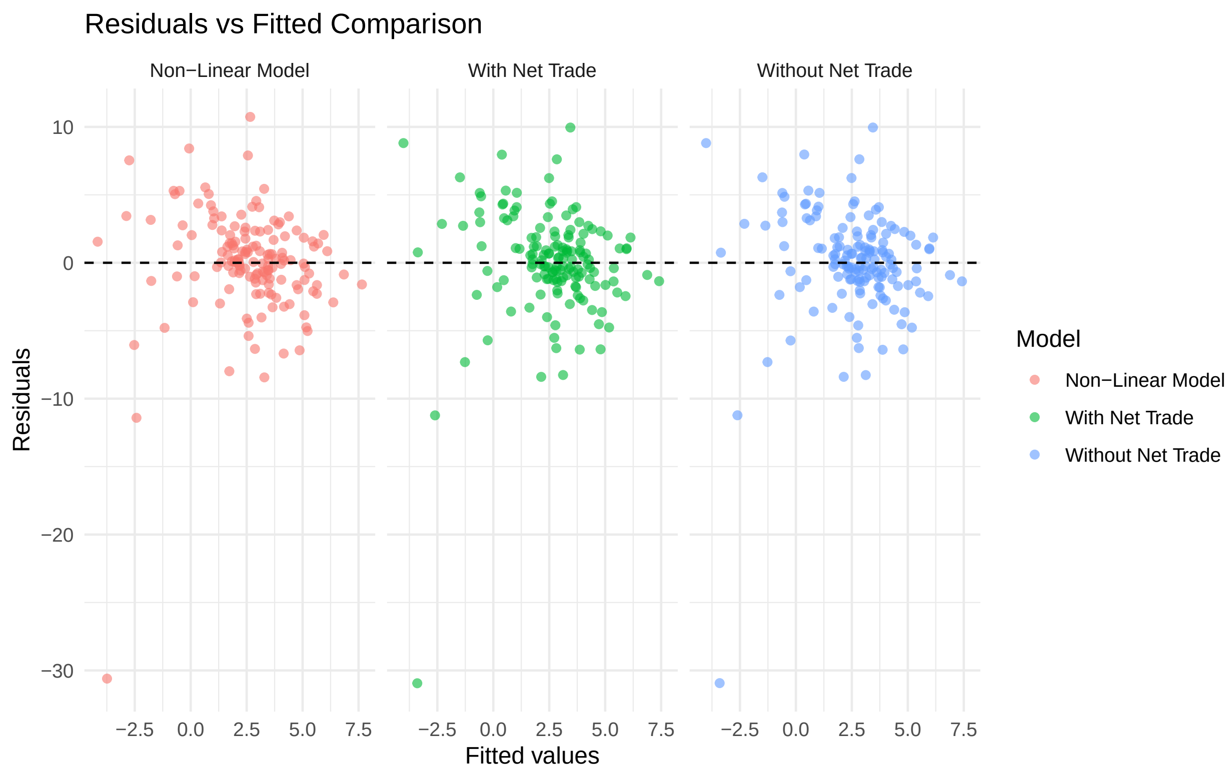
Table 6: OLS Regression with Nonlinear and Interaction Effects

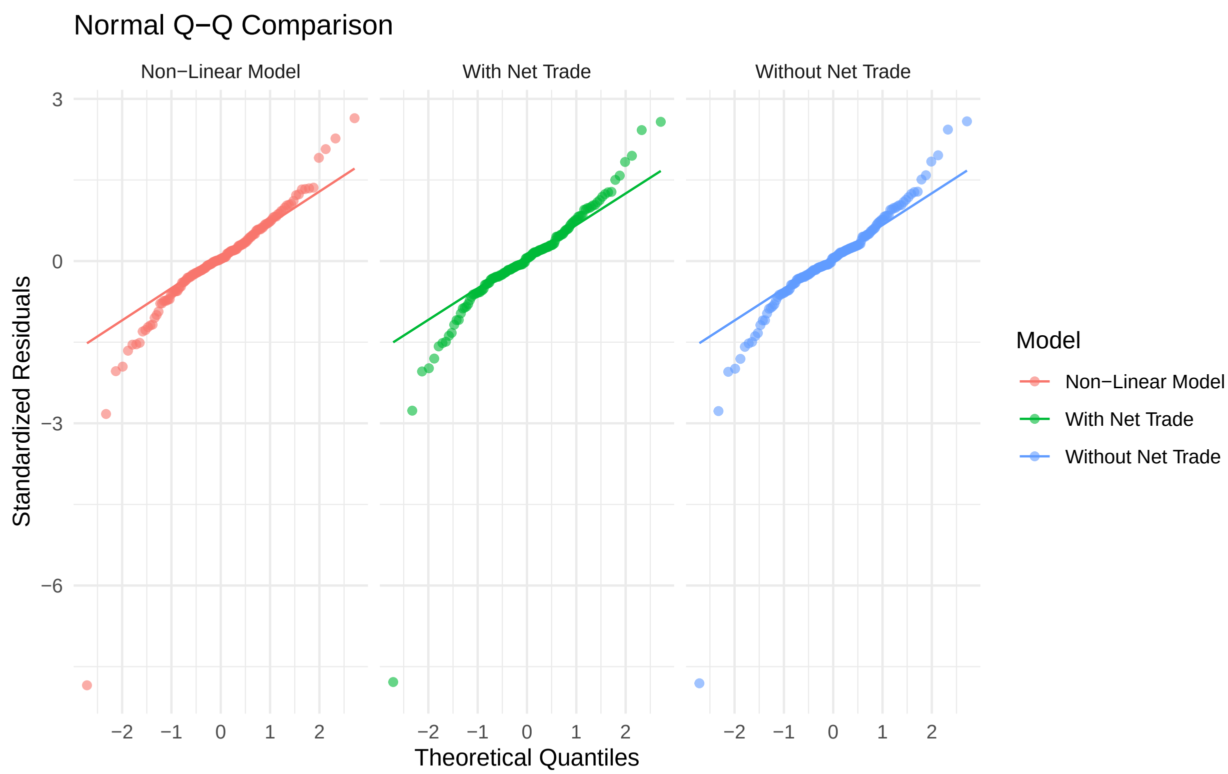
	Dependent variable:
	GDP growth (annual %)
Fertility rate	-0.127
	(0.651)
Employers (%)	-0.500^{***}
	(0.123)
Labor force participation (%)	-0.009
	(0.058)
Life expectancy (male)	0.303^{***}
	(0.096)
Unemployment (%)	-0.262
	(0.342)
Export value index	0.002^{**}
	(0.001)
Population growth (%)	0.011
	(0.658)
Population growth $(\%)^2$	0.049
	(0.186)
GDP per capita (PPP)	-0.0002^{**}
	(0.0001)
GDP per capita $(\text{PPP})^2$	0.003^{*}
	(0.001)
Wage & salaried workers (%)	0.007
	(0.028)
Unemp $\times$ LFP	0.003
	(0.006)
Constant	-13.929
	(8.753)
Observations	150
R^2	0.206
Adjusted R^2	0.137
F Statistic	2.967^{***} (df = 12; 137)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 9: Variables Selected under .min vs .1se

Variable	Coef_min	Coef_1se
(Intercept)	-1.7794971	3.001157
Adolescent fertility rate (births per 1,000 women ages 15-19)	-0.0039180	NA
Contributing family workers, female (% of female employment) (modeled ILO estimate)	-0.0067494	NA
Depth of credit information index (0=low to 8=high)	0.0269231	NA
Employers, male (% of male employment) (modeled ILO estimate)	-0.1109801	NA
Employment in industry, male (% of male employment) (modeled ILO estimate)	0.0328390	NA
Export value index (2000 = 100)	0.0013346	NA
GDP per capita, PPP (constant 2011 international $)	-0.0000529	NA
Labor force participation rate, female (% of female population ages 15+) (modeled ILO estimate)	-0.0413561	NA
Labor force participation rate, male (% of male population ages 15+) (modeled ILO estimate)	0.0076160	NA
Life expectancy at birth, male (years)	0.1067299	NA
Own-account workers, total (% of male employment) (modeled ILO estimate)	0.0079536	NA
Population ages 15-64, total	0.0000000	NA
Population ages 65 and above (% of total)	-0.0107295	NA
Rural population	0.0000000	NA
Tax payments (number)	0.0261170	NA
Time required to start a business (days)	-0.0061819	NA
Time to prepare and pay taxes (hours)	-0.0031672	NA
Unemployment, male (% of male labor force) (modeled ILO estimate)	-0.0353272	NA
Unemployment, youth male (% of male labor force ages 15-24) (modeled ILO estimate)	-0.0287594	NA

Appendix B






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

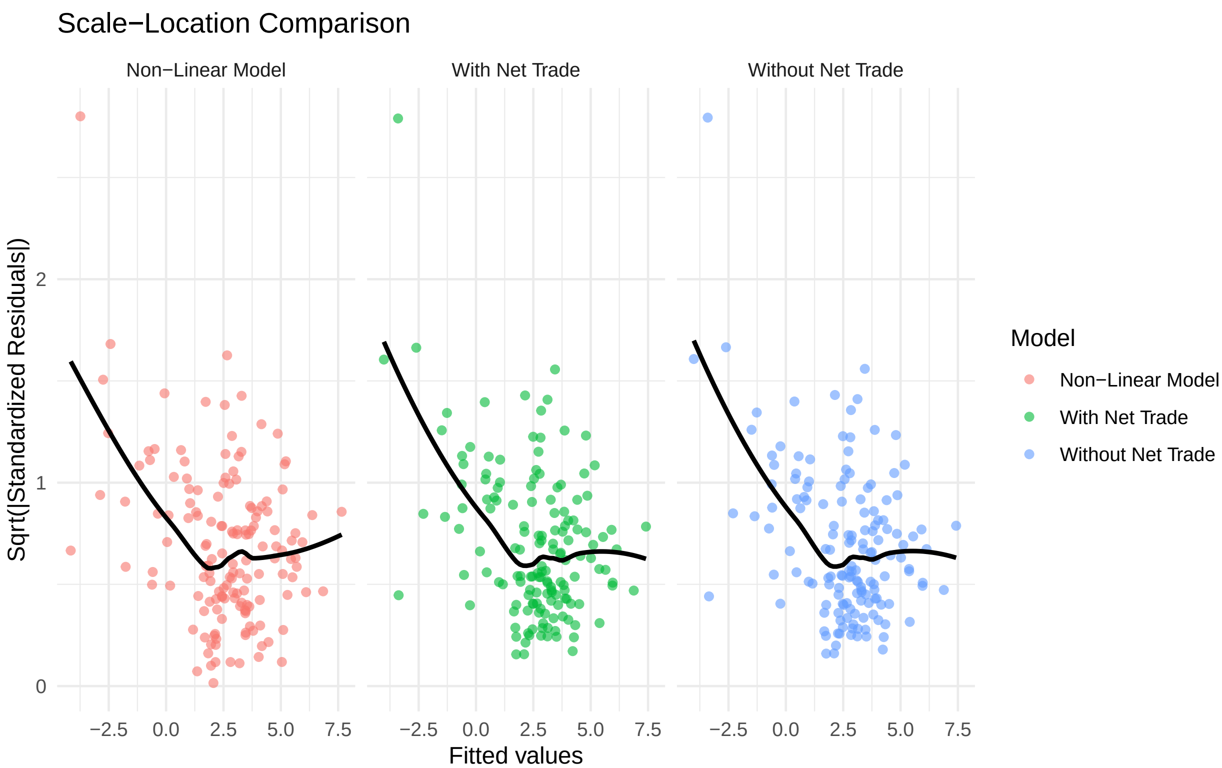


Figure 1: Residuals vs Fitted for all specifications

Appendix C

```
## ----message=FALSE, warning=FALSE, include=FALSE-----------------------------
# Directory setup
path = dirname(rstudioapi::getSourceEditorContext()$path) # Path is directory of this file
setwd(path)      # Set working directory
# install pacman if you don't have it yet
if (!require("pacman")) install.packages("pacman")
# load all packages at once
pacman::p_load(
  tidyverse, caret, glmnet, car, lmtest, tinytex, broom,
  ggplot2, dplyr, tidyverse, kableExtra, knitr,stargazer)
# no scientific notations
options(scipen = 999)
rm(list = ls())
# Read data
stopifnot(file.exists("Data/a1_data_group_9.csv"))
df <- readr::read_csv("Data/a1_data_group_9.csv", show_col_types = FALSE)


## ----include=FALSE-----------------------------------------------------------
# Data Splitting.
set.seed(254646)
n1 <- 110
idx <- sample(1:nrow(df), n1)
```

```r
data_clean <- df
dfm_train <- data_clean[idx, ]
dfm_test <- data_clean[-idx, ]
Xvars_mat_train <- model.matrix(`GDP growth (annual %)` ~.-Country,data = dfm_train)[, -1]
Yvar_train      <- dfm_train$`GDP growth (annual %)`
Xvars_mat_test  <- model.matrix(`GDP growth (annual %)` ~.-Country,data = dfm_test)[, -1]
Yvar_test       <- dfm_test$`GDP growth (annual %)`


## ----include=FALSE------------------------------------------------------------
# OLS Regression
# Run linear regression with Net Trade
model <- lm(`GDP growth (annual %)` ~
              `Fertility rate, total (births per woman)` +
              `Employers, total (% of total employment) (modeled ILO estimate)` +
              `Labor force participation rate, total (% of total population ages 15+) (modeled ILO esti
              `Life expectancy at birth, male (years)` +
              `Unemployment, total (% of total labor force) (modeled ILO estimate)` +
              `Export value index (2000 = 100)` +
              `Population growth (annual %)` +
              `GDP per capita, PPP (constant 2011 international $)` +
              `Net trade in goods and services (BoP, current US$)` +
              `Wage and salaried workers, total (% of total employment) (modeled ILO estimate)`,
            data = df,
            singular.ok = FALSE)

# Model without Net Trade
model_without <- update(model, . ~ . - `Net trade in goods and services (BoP, current US$)`, singular.o



## ----include=FALSE------------------------------------------------------------
# OLS Regression with non linear and interaction effects

# --- 1) Engineer features: nonlinear + interaction ---
df2 <- within(df, {
  `GDP per capita, PPP (constant 2011 international $)^2` <-
    (`GDP per capita, PPP (constant 2011 international $)`)^2
  `Population growth (annual %)^2` <- (`Population growth (annual %)`^2)
  `Unemp x LFP` <-
    `Unemployment, total (% of total labor force) (modeled ILO estimate)` *
    `Labor force participation rate, total (% of total population ages 15+) (modeled ILO estimate)`
})
# --- 2) Build linear regression with nonlinear effects (no intercept column) ---
model_ext <- lm(
  `GDP growth (annual %)` ~
    `Fertility rate, total (births per woman)` +
    `Employers, total (% of total employment) (modeled ILO estimate)` +
    `Labor force participation rate, total (% of total population ages 15+) (modeled ILO estimate)` +
    `Life expectancy at birth, male (years)` +
    `Unemployment, total (% of total labor force) (modeled ILO estimate)` +
    `Export value index (2000 = 100)` +
    `Population growth (annual %)` +
```

```r
    `Population growth (annual %)^2` +   # nonlinear term (already in df2)
    `GDP per capita, PPP (constant 2011 international $)` +
    `GDP per capita, PPP (constant 2011 international $)^2` +  # nonlinear term (convergence)
    `Wage and salaried workers, total (% of total employment) (modeled ILO estimate)` +
    `Unemp x LFP`,                            # interaction term (already in df2)
  data = df2,
  singular.ok = FALSE
)


## ----echo=FALSE----------------------------------------------------------------


# Put models into a list
models <- list(
  "With Net Trade" = model,
  "Without Net Trade" = model_without,
  "Non-Linear Model" = model_ext
)

# Convert each model's data into a tidy dataframe
df_models <- lapply(names(models), function(name) {
  augment(models[[name]]) %>%
    mutate(Model = name)
}) %>% bind_rows()

# --- Residuals vs Fitted plot ---
gg_resfit <- ggplot(df_models, aes(.fitted, .resid, color = Model)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  facet_wrap(~Model) +
  labs(title = "Residuals vs Fitted Comparison",
       x = "Fitted values",
       y = "Residuals") +
  theme_minimal()

# --- Normal Q-Q plot ---
gg_qq <- ggplot(df_models, aes(sample = .std.resid, color = Model)) +
  stat_qq(alpha = 0.6) +
  stat_qq_line() +
  facet_wrap(~Model) +
  labs(title = "Normal Q-Q Comparison",
       x = "Theoretical Quantiles",
       y = "Standardized Residuals") +
  theme_minimal()

# --- Scale-Location plot (sqrt(|residuals|) vs fitted) ---
gg_scale <- ggplot(df_models, aes(.fitted, sqrt(abs(.std.resid)), color = Model)) +
  geom_point(alpha = 0.6) +
  geom_smooth(se = FALSE, color = "black") +
  facet_wrap(~Model) +
  labs(title = "Scale-Location Comparison",
       x = "Fitted values",
```

```r
      y = "Sqrt(|Standardized Residuals|)") +
  theme_minimal()
```

```r
## ----include=FALSE---------------------------------------------------------
# LASSO Regression

# Prepare predictors (X) and response (y)
X_lasso <- model.matrix(
  `GDP growth (annual %)` ~
    `Fertility rate, total (births per woman)` +
    `Employers, total (% of total employment) (modeled ILO estimate)` +
    `Labor force participation rate, total (% of total population ages 15+) (modeled ILO estimate)` +
    `Life expectancy at birth, male (years)` +
    `Unemployment, total (% of total labor force) (modeled ILO estimate)` +
    `Export value index (2000 = 100)` +
    `Population growth (annual %)` +
    `Population growth (annual %)^2` +         # Nonlinear
    `GDP per capita, PPP (constant 2011 international $)` +
    `GDP per capita, PPP (constant 2011 international $)^2` +   # Nonlinear (convergence effect)
    `Net trade in goods and services (BoP, current US$)` +
    `Wage and salaried workers, total (% of total employment) (modeled ILO estimate)` +
    `Unemp x LFP`,                              # Interaction
  data = df2
)[, -1]  # drop intercept

y_lasso <- df2$`GDP growth (annual %)`

# --- 3) Cross-validated LASSO ---
set.seed(2354245)                    # reproducibility
cv_lasso_nl_int <- cv.glmnet(
  X_lasso, y_lasso,
  alpha = 1,                     # LASSO
  nfolds = 10,
  standardize = TRUE)

# --- 4) Inspect results ---
cv_lasso_nl_int$lambda.min
cv_lasso_nl_int$lambda.1se

coef(cv_lasso_nl_int, s = "lambda.min")
coef(cv_lasso_nl_int, s = "lambda.1se")
```

```r
## ----include=FALSE---------------------------------------------------------
# Importance of Standardization
# LASSO without standardization
set.seed(2134)
```

```r
cv_no_std <- cv.glmnet(X_lasso, y_lasso, alpha = 1, nfolds = 10, standardize = FALSE)
coef(cv_no_std, s = "lambda.min")
# LASSO with standardization (default in glmnet)
set.seed(2134)
cv_std <- cv.glmnet(X_lasso, y_lasso, alpha = 1, nfolds = 10, standardize = TRUE)
coef(cv_std, s = "lambda.min")


## ----echo=FALSE-------------------------------------------------------------------
# Extract coefficients
coef_no_std <- as.matrix(coef(cv_no_std, s = "lambda.min"))
coef_std    <- as.matrix(coef(cv_std, s = "lambda.min"))

# Combine into a comparison table
comparison <- data.frame(
  Variable = rownames(coef_no_std),
  `No Standardization` = round(coef_no_std[,1], 6),
  Standardization = round(coef_std[,1], 6)
)




## ----echo=FALSE-------------------------------------------------------------------
# LASSO
set.seed(254646)
# Grid search for lamda
lambda_grid <- seq(0.001, 2, length.out = 100)

cv_lasso <- cv.glmnet(
  Xvars_mat_train, Yvar_train,
  family = "gaussian",
  alpha = 1,     # LASSO
  lambda = lambda_grid,
  nfolds = 10,
  type.measure = "mse",
  standardize = TRUE)

best_lambda_min  <- cv_lasso$lambda.min
best_lambda_1se  <- cv_lasso$lambda.1se


## ----echo=FALSE-------------------------------------------------------------------
# Extract coefficients
coef_min <- as.matrix(coef(cv_lasso, s = "lambda.min"))
coef_1se <- as.matrix(coef(cv_lasso, s = "lambda.1se"))

# Get selected variables (non-zero coefficients)
sel_min <- coef_min[coef_min[,1] != 0, , drop = FALSE]
sel_1se <- coef_1se[coef_1se[,1] != 0, , drop = FALSE]

# Create data frames
df_min <- data.frame(Variable = rownames(sel_min), Coef_min = sel_min[,1])
```

```r
df_1se <- data.frame(Variable = rownames(sel_1se), Coef_1se = sel_1se[,1])

# Merge into one comparative table
comparative_tbl <- merge(df_min, df_1se, by = "Variable", all = TRUE)




## ----echo=FALSE-----------------------------------------------------------
# Functions for metrics.
rmse <- function(y, p) sqrt(mean((y - p)^2))
mae  <- function(y, p) mean(abs(y - p))
r2   <- function(y, p) 1 - sum((y - p)^2) / sum((y - mean(y))^2)

row_metrics <- function(name, y, p) {
  data.frame(
    Model = name,
    RMSE  = rmse(y, p),
    MAE   = mae(y, p),
    R2    = r2(y, p),
    SD_y  = sd(y))}

nz_coefs <- function(cv_obj, s) {
  cf <- as.matrix(coef(cv_obj, s = s))
  out <- data.frame(Variable = rownames(cf), Coef = cf[,1], row.names = NULL)
  subset(out, Coef != 0)}

# Lasso Predictions.
pred_lasso_min <- as.numeric(predict(cv_lasso, newx = Xvars_mat_test, s = "lambda.min"))
pred_lasso_1se <- as.numeric(predict(cv_lasso, newx = Xvars_mat_test, s = "lambda.1se"))
met_lasso_min  <- row_metrics("Lasso (lambda.min)",  Yvar_test, pred_lasso_min)
met_lasso_1se  <- row_metrics("Lasso (lambda.1se)",  Yvar_test, pred_lasso_1se)
sel_lasso_min  <- nz_coefs(cv_lasso, "lambda.min")
sel_lasso_1se  <- nz_coefs(cv_lasso, "lambda.1se")

# Baseline (Mean of the Variable)
pred_base <- rep(mean(Yvar_train), length(Yvar_test))
met_base  <- row_metrics("Baseline (mean of train)", Yvar_test, pred_base)


## ----include=FALSE--------------------------------------------------------
# ELASTIC NET
set.seed(2546)
# Grid Search for alpha and
alpha_grid <- seq(0.1, 0.9, by = 0.1)

# Train the model
cv_list <- lapply(alpha_grid, function(a)
  cv.glmnet(
    Xvars_mat_train, Yvar_train,
    family = "gaussian",
    alpha  = a,
    nfolds = 10,
```

```r
    type.measure = "mse",
    standardize = TRUE))

cv_mins <- sapply(cv_list, function(cv) min(cv$cvm))
best_i  <- which.min(cv_mins)
best_a  <- alpha_grid[best_i]
best_cv <- cv_list[[best_i]]

cat("\nElastic Net - best alpha:", best_a, "| lambda.min:", signif(best_cv$lambda.min, 4),
    "| lambda.1se:", signif(best_cv$lambda.1se, 4), "\n")


## ----echo=FALSE------------------------------------------------------------
# Existing LASSO table
lam_tbl <- data.frame(
  Method = c("LASSO", "LASSO"),
  Metric = c("$ $.min", "$ $.1se"),
  Value  = c(signif(best_lambda_min, 4), signif(best_lambda_1se, 4))
)

# Add Elastic Net results
elastic_tbl <- data.frame(
  Method = c("Elastic Net", "Elastic Net", "Elastic Net"),
  Metric = c("$ $", "$ $.min", "$ $.1se"),
  Value  = c(best_a, signif(best_cv$lambda.min, 4), signif(best_cv$lambda.1se, 4))
)

# Combine
lam_tbl <- rbind(lam_tbl, elastic_tbl)

# Display
knitr::kable(lam_tbl, caption = "Optimal tuning parameters for LASSO and Elastic Net")



## ----include=FALSE---------------------------------------------------------

# Elastic Net Predictions.
pred_en_min <- as.numeric(predict(best_cv, newx = Xvars_mat_test, s = "lambda.min"))
pred_en_1se <- as.numeric(predict(best_cv, newx = Xvars_mat_test, s = "lambda.1se"))
met_en_min  <- row_metrics(paste0("Elastic Net  =", best_a, " (lambda.min)"), Yvar_test, pred_en_min)
met_en_1se  <- row_metrics(paste0("Elastic Net  =", best_a, " (lambda.1se)"), Yvar_test, pred_en_1se)
sel_en_min  <- nz_coefs(best_cv, "lambda.min")
sel_en_1se  <- nz_coefs(best_cv, "lambda.1se")


## ----echo=FALSE------------------------------------------------------------
# Metric Comparison.
metrics_all <- rbind(met_base, met_lasso_min, met_lasso_1se, met_en_min, met_en_1se)

metrics_tbl <- metrics_all %>%
  dplyr::select(Model, RMSE, MAE, R2, SD_y) %>%
  mutate(across(where(is.numeric), ~ round(., 3)))
```

```r
best_rmse_idx <- which.min(metrics_all$RMSE)

kable(metrics_tbl, caption = "Test Performance (RMSE, MAE, $R²$, $SD_y$)") %>%
  kable_styling(full_width = FALSE, position = "center", bootstrap_options = c("striped","hover","conde
  row_spec(best_rmse_idx, bold = TRUE)


## ----include=FALSE-------------------------------------------------------------
# Best features used.
top_k <- 15
top_lasso_min <- head(sel_lasso_min[order(abs(sel_lasso_min$Coef), decreasing = TRUE), ], top_k)
top_lasso_1se <- head(sel_lasso_1se[order(abs(sel_lasso_1se$Coef), decreasing = TRUE), ], top_k)
top_en_min    <- head(sel_en_min[order(abs(sel_en_min$Coef),   decreasing = TRUE), ], top_k)
top_en_1se    <- head(sel_en_1se[order(abs(sel_en_1se$Coef),   decreasing = TRUE), ], top_k)


# Amount of Variables Selected in each model.
count_nz <- function(df) sum(df$Variable != "(Intercept)")
cat("\n Amount of Variables -> Lasso min:", count_nz(sel_lasso_min),"| Lasso 1se:", count_nz(sel_lasso_
    "| ENet min:", count_nz(sel_en_min), "| ENet 1se:", count_nz(sel_en_1se), "\n")

## ----echo=FALSE----------------------------------------------------------------
# Extract variable names from each model
vars_lasso_min <- sel_lasso_min$Variable
vars_lasso_1se <- sel_lasso_1se$Variable
vars_en_min    <- sel_en_min$Variable
vars_en_1se    <- sel_en_1se$Variable

# Combine into one master list
all_vars <- unique(c(vars_lasso_min, vars_lasso_1se, vars_en_min, vars_en_1se))

# Build comparative data frame
comparative_df <- data.frame(
  Variable      = all_vars,
  L_min     = ifelse(all_vars %in% vars_lasso_min, "T", "F"),
  L_1se     = ifelse(all_vars %in% vars_lasso_1se, "T", "F"),
  ENet_min = ifelse(all_vars %in% vars_en_min, "T", "F"),
  ENet_1se = ifelse(all_vars %in% vars_en_1se, "T", "F")
)



## ----include=FALSE-------------------------------------------------------------
# Ridge and Logistic
df_logistic <- data_clean[ , !(names(data_clean) %in% "Country")]
df_logistic$GrowingMore <- as.integer(df_logistic$`GDP growth (annual %)` > 2.7)

# Data Splitting (Seed One)
set.seed(1111)
train_index <- sample.int(nrow(df_logistic), 110)
train_set <- df_logistic[train_index, ]
test_set  <- df_logistic[-train_index, ]
y_train <- train_set$GrowingMore
```

```r
y_test  <- test_set$GrowingMore
X_train <- model.matrix(GrowingMore ~ . -`GDP growth (annual %)`, data=train_set)[, -1]
X_test  <- model.matrix(GrowingMore ~ . - `GDP growth (annual %)`, data=test_set )[,-1]


## ----include=FALSE------------------------------------------------------------
cv_ridge <- cv.glmnet(
  X_train, y_train,
  family = "binomial",
  alpha  = 0,
  nfolds = 10,
  type.measure = "deviance",
  standardize = TRUE)

lam_min <- cv_ridge$lambda.min
lam_1se <- cv_ridge$lambda.1se
cat("lambda.min =", signif(lam_min,4), " | lambda.1se =", signif(lam_1se,4), "\n")

# Prediction on Test Using both variables.
p_min <- as.numeric(predict(cv_ridge, newx = X_test, s = "lambda.min", type = "response"))
p_1se <- as.numeric(predict(cv_ridge, newx = X_test, s = "lambda.1se", type = "response"))

# Classify at 0.5
pred_min <- ifelse(p_min >= 0.5, 1, 0)
pred_1se <- ifelse(p_1se >= 0.5, 1, 0)


## ----include=FALSE------------------------------------------------------------
# Metrics for performance.
logloss <- function(y, p) { p <- pmin(pmax(p, 1e-15), 1-1e-15); -mean(y*log(p) + (1-y)*log(1-p)) }
brier   <- function(y, p) mean((p - y)^2)

acc_min  <- mean(pred_min == y_test)
acc_1se  <- mean(pred_1se == y_test)
ll_min   <- logloss(y_test, p_min)
ll_1se   <- logloss(y_test, p_1se)
br_min   <- brier(y_test, p_min)
br_1se   <- brier(y_test, p_1se)

# optional AUC
get_auc <- function(y,p){
  if (requireNamespace("pROC", quietly = TRUE)) as.numeric(pROC::auc(y, p)) else NA_real_}
auc_min <- get_auc(y_test, p_min)
auc_1se <- get_auc(y_test, p_1se)

results <- data.frame(
  Model    = c("Ridge (lambda.min)", "Ridge (lambda.1se)"),
  Accuracy= c(acc_min, acc_1se),
  LogLoss = c(ll_min, ll_1se),
  Brier    = c(br_min, br_1se),
  AUC      = c(auc_min, auc_1se))

print(results, row.names = FALSE)
```

```r
## ----echo=FALSE------------------------------------------------------------
results_tbl <- results %>%
  transmute(Model, Accuracy = round(Accuracy, 4), LogLoss  = round(LogLoss, 4),
    Brier = round(Brier, 4), AUC = ifelse(is.na(AUC), NA, round(AUC, 3)))

kable(results_tbl, caption = "Ridge (logistic) - Test Metrics") %>%
  kable_styling(full_width = FALSE, position = "center", bootstrap_options = c("striped","hover","conde


## ----include=FALSE---------------------------------------------------------
# Final Model and Coefficientes.
coef_min <- coef(cv_ridge, s = "lambda.min")
coef_1se <- coef(cv_ridge, s = "lambda.1se")
# Example: show 10 largest (by |coef|) at lambda.1se
ix <- order(abs(as.numeric(coef_1se)) , decreasing = TRUE)
print(coef_1se[ix[1:10], , drop = FALSE])


## ----include=FALSE---------------------------------------------------------
# Data Splitting (Seed Two)
set.seed(9999)
train_index <- sample.int(nrow(df_logistic), 110)
train_set <- df_logistic[train_index, ]
test_set  <- df_logistic[-train_index, ]
y_train <- train_set$GrowingMore
y_test  <- test_set$GrowingMore
X_train <- model.matrix(GrowingMore ~ . -`GDP growth (annual %)`, data=train_set)[, -1]
X_test  <- model.matrix(GrowingMore ~ . - `GDP growth (annual %)`, data=test_set )[,-1]


## ----include=FALSE---------------------------------------------------------
# Ridge and Elastic Net.
cv_ridge <- cv.glmnet(
  X_train, y_train,
  family = "binomial",
  alpha  = 0,
  nfolds = 10,
  type.measure = "deviance",
  standardize = TRUE)

lam_min <- cv_ridge$lambda.min
lam_1se <- cv_ridge$lambda.1se
cat("lambda.min =", signif(lam_min,4), " | lambda.1se =", signif(lam_1se,4), "\n")

# Prediction on Test Using both variables.
p_min <- as.numeric(predict(cv_ridge, newx = X_test, s = "lambda.min", type = "response"))
p_1se <- as.numeric(predict(cv_ridge, newx = X_test, s = "lambda.1se", type = "response"))

# Classify at 0.5
pred_min <- ifelse(p_min >= 0.5, 1, 0)
pred_1se <- ifelse(p_1se >= 0.5, 1, 0)
```

```r
## ----include=FALSE-----------------------------------------------------------
# Metrics for performance.
logloss <- function(y, p) { p <- pmin(pmax(p, 1e-15), 1-1e-15); -mean(y*log(p) + (1-y)*log(1-p)) }
brier   <- function(y, p) mean((p - y)^2)
acc_min  <- mean(pred_min == y_test)
acc_1se  <- mean(pred_1se == y_test)
ll_min   <- logloss(y_test, p_min)
ll_1se   <- logloss(y_test, p_1se)
br_min   <- brier(y_test, p_min)
br_1se   <- brier(y_test, p_1se)
# optional AUC
get_auc <- function(y,p){
  if (requireNamespace("pROC", quietly = TRUE)) as.numeric(pROC::auc(y, p)) else NA_real_}
auc_min <- get_auc(y_test, p_min)
auc_1se <- get_auc(y_test, p_1se)

results1 <- data.frame(
  Model    = c("Ridge (lambda.min)", "Ridge (lambda.1se)"),
  Accuracy= c(acc_min, acc_1se),
  LogLoss = c(ll_min, ll_1se),
  Brier    = c(br_min, br_1se),
  AUC      = c(auc_min, auc_1se))

print(results, row.names = FALSE)

results_tbl_1 <- results1 %>%
  transmute(
    Model,
    Accuracy = round(Accuracy, 4),
    LogLoss  = round(LogLoss, 4),
    Brier    = round(Brier, 4),
    AUC      = ifelse(is.na(AUC), NA, round(AUC, 3)))

kable(results_tbl_1, caption = "Ridge (logistic) - Test Metrics") %>%
  kable_styling(full_width = FALSE, position = "center", bootstrap_options = c("striped","hover","conde


## ----include=FALSE-----------------------------------------------------------
coef_min <- coef(cv_ridge, s = "lambda.min")
coef_1se <- coef(cv_ridge, s = "lambda.1se")
ix <- order(abs(as.numeric(coef_1se)) , decreasing = TRUE)
print(coef_1se[ix[1:10], , drop = FALSE])


## ----echo=FALSE--------------------------------------------------------------
tblA <- results_tbl_1 %>%
  select(Model, Accuracy, LogLoss, Brier, AUC) %>%
  mutate(across(-Model, ~round(., 4)))
tblB <- results_tbl %>%
  select(Model, Accuracy, LogLoss, Brier, AUC) %>%
  mutate(across(-Model, ~round(., 4)))
wide <- inner_join(tblA, tblB, by = "Model", suffix = c("_run1", "_run2"))
# Table Output.
```

```r
kable(
  wide,
  col.names = c("Model", "Accuracy", "LogLoss", "Brier", "AUC",
                "Accuracy", "LogLoss", "Brier", "AUC"),
  caption = "Ridge (logistic) - Test Metrics (different seeds)"
) %>%
  kable_styling(full_width = FALSE, position = "center",
                bootstrap_options = c("striped","hover","condensed")) %>%
  add_header_above(c(" " = 1, "Run 1" = 4, "Run 2" = 4))


## ----echo=FALSE---------------------------------------------------------------


# Pretty table
kable(comparison[ , -1], caption = "Comparison of LASSO Coefficients With and Without Standardization")


## ----echo=FALSE---------------------------------------------------------------
# Display nicely
knitr::kable(comparative_df, caption = "Variable selection across LASSO and Elastic Net")


## ----echo=FALSE---------------------------------------------------------------
# Show the table
knitr::kable(comparative_tbl, caption = "Variables Selected under  .min vs  .1se")



## ----echo=FALSE, fig.width=8, fig.height=5, fig.cap="Residuals vs Fitted for all specifications"----
print(gg_resfit)
print(gg_qq)
print(gg_scale)


## ----echo=FALSE, results='asis'-----------------------------------------------
code_file <- knitr::purl("Assignment IDS.rmd", quiet = TRUE)
code <- readLines(code_file, warn = FALSE)
# Emit a proper fenced R code block so Pandoc treats it as verbatim
knitr::asis_output(paste0("```r\n", paste(code, collapse = "\n"), "\n```\n"))
```