Predicting Life Expectancy using PCA and Elastic Net

FEM11149 - Introduction to Data Science

Aleksandra Tatko (648925)

October, 2025

1. Introduction

Life expectancy is considered a key indicator of human development. It reflects the combined effects of healthcare systems, socioeconomic conditions, and living standards across nations. Understanding its determinants is crucial for guiding health and development policies that support population well-being. Existing literature consistently highlights fertility rate, income, and health expenditure as major positive drivers of longevity (Roffia, Bucciol, and Hashlamoun 2023; Irandoust et al. 2025), while tuberculosis incidence, high fertility, and out-of-pocket healthcare spending are associated with poorer health outcomes (Mondal et al. 2019; Karlsson et al. 2025). Using a comprehensive dataset of economic, demographic, and health indicators, this reserach aim to build up on that literature by identifying the most reliable and interpretable predictors of life expectancy across countries. The analysis is done by comparing three predictive models - Ordinary Least Squares, Principal Component Regression (PCR), and Elastic Net regularization. Lastly, the best-performing model is applied to predict life expectancy for three countries, the Netherlands, Colombia, and Kenya, to test its out-of-sample generalization.

2. Data

The analysis combines three datasets sourced from the World Bank Database. The first includes 30 socioeconomic, demographic, and health-related variables for 160 countries. The second provides the target dependent variable, life expectancy at birth (in years) for the same countries. The third contains the same 30 predictors for three additional countries used for out-of-sample prediction. The first 2 datasets were merged by country name to create a single cross-sectional dataset with 30 variables and 160 observations used for the analysis. Economic indicators include variables such as gross national income per capita, unemployment rate, and population growth, reflecting macroeconomic conditions. Healthcare investments were captioned by variables like health expenditure (government, private, and out-of-pocket spending per capita). Public health outcomes are represented by immunization coverage, disease prevalence (e.g., tuberculosis, diabetes, hypertension), and treatment success rates. Infrastructure and living conditions are summarized by access to sanitation and safe drinking water, urbanization levels, and alcohol consumption. Together, these indicators provide a comprehensive picture of each country's health and development profile in relation to life expectancy.

3. Methodology

This study applies three regression approaches, OLS, PCR, and Elastic Net, to predict life expectancy across countries. Each method offers a distinct strategy for handling multicollinearity, balancing interpretability, and improving predictive accuracy.

Ordinary Least Squares (OLS) estimates a linear relationship between the dependent variable Y (life expectancy) and predictors $X_1, X_2, \ldots, X_k$:

$$Y_i = \beta_0 + \sum_{j=1}^{k} \beta_j X_{ji} + \varepsilon_i$$

where coefficients β_j minimize the *sum of squared residuals*. An AIC-based stepwise variable selection is often used to retain only predictors that meaningfully improved model fit, removing redundant variables to achieve a more efficient and interpretable specification. However, OLS assumes predictors are not highly correlated. When multicollinearity exists, coefficient estimates become unstable, leading to inflated standard errors and unreliable inference.

Principal Component Regression (PCR) is an unsupervised technique that mitigates multicollinearity by transforming correlated predictors into a smaller set of uncorrelated principal components (PCs). Each component is a linear combination of the original variables:

$$Z_m = a_{1m} X_1 + a_{2m} X_2 + \cdots + a_{km} X_k$$

where the weights a_{jm} (loadings) are chosen so that each successive component explains the maximum possible variance in the predictors.Because these components are orthogonal, PCR eliminates collinearity by construction. The regression model is then estimated on the first p components:

$$Y = \alpha_0 + \sum_{m=1}^{p} \alpha_m Z_m + \varepsilon$$

The number of components is selected using three complementary methods:
(1) Kaiser's criterion retains components with eigenvalues > 1, ensuring each explains more variance than an average single variable, (2) the scree plot, identifies the "elbow" point where additional components yield diminishing returns in explained variance, and (3) permutation test compares observed eigenvalues to those derived from randomly permuted data, retaining only components that explain significantly more variance than expected by chance. To evaluate the stability of component importance, bootstrap resampling is applied to estimate confidence intervals for eigenvalues, providing a measure of their sampling uncertainty and supporting a robust application of Kaiser's rule. Finally, a bootstrap test of cumulative variance explained (VAF) verifies whether the retained components jointly explain a statistically significant share of the total variance (around 70%), ensuring both efficiency and explanatory adequacy. While PCR improves estimation stability and predictive performance in the presence of multicollinearity, it sacrifices direct interpretability, since the resulting components are linear combinations of the original variables. Additionally, as PCA is based on variance maximization, it remains sensitive to outliers that can disproportionately influence component directions.

Elastic Net regularization combines LASSO (L_1) and Ridge (L_2) penalties to handle multicollinearity while performing variable selection:

$$\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (Y_i - X_i^\top \beta)^2 + \lambda \left[\alpha \sum |\beta_j| + \frac{1-\alpha}{2} \sum \beta_j^2 \right] \right\}$$

Here, λ controls the overall penalty strength—larger values lead to greater shrinkage of coefficients—while α determines the balance between LASSO ($\alpha = 1$) and Ridge ($\alpha = 0$).
This hybrid approach reduces model variance (through Ridge) and removes redundant predictors (through LASSO), making it robust in the presence of correlated variables. Optimal λ and α are selected through cross-validation, minimizing prediction error. Model performance is evaluated using three key metrics:
- R^2 — explains the proportion of variance in predicted variable captured by the model. A higher R^2

indicates that the model captures more of the variation in the data, reflecting stronger explanatory power.
- RMSE (Root Mean Squared Error) — measures the average magnitude of prediction error, penalizing large deviations more strongly. Lower RMSE values indicate higher prediction accuracy.
- MAE (Mean Absolute Error) — captures the average absolute prediction error and is less sensitive to outliers. Lower MAE values imply better fit and greater robustness.

It is generally prefered to choose models with higher R^2 and lower RMSE and MAE, as this combination reflects stronger explanatory performance and more accurate, stable predictions. Comparing these metrics across OLS, PCR, and Elastic Net highlights the trade-offs between interpretability, dimensionality reduction, and predictive robustness in modeling life expectancy.

4. Results

Following best practices from *An Introduction to Statistical Learning with R* (James et al. 2023), the dataset was split into an 80/20 training–testing partition to ensure all models were evaluated on identical samples for fair performance comparison. All preprocessing steps—median imputation, log transformation, and scaling—were performed using training-set statistics and then applied to the test data to prevent leakage. Log transformation reduced skewness present in the data (*Appendix 1*), improving normality and linearity assumptions, while median imputation handled missing values robustly against outliers. Finally, predictors were standardized to a common scale to avoid dominance by large-valued variables such as income or population. These steps were applied consistently across all models.

A baseline **OLS model** was estimated using an AIC-based stepwise selection procedure to identify the most statistically relevant predictors of life expectancy. The retained predictors included fertility rate, health and government health expenditure, GNI per capita, access to water, rural and urban population shares, immunization coverage, and tuberculosis-related variables. Diagnostic tests indicated generally well-behaved residuals: the Q–Q plot suggested normal distribution while the Residuals vs Fitted plot supported the assumption of linearity and homoscedasticity (*Appendix 2 OLS*). Although the model achieved high R^2 (RMSE = 2.85, MAE = 2.25, R^2 = 0.84) and strong cross-validated fit, this performance likely reflects in-sample optimization rather than true generalization. Moreover, several predictors exhibited severe multicollinearity (VIF > 10) suggesting inflated standard errors and unstable coefficient estimates (*Appendix 2 OLS*). This implies that while the model fits the training data very well, its predictive generalization may be unreliable, and R^2 may overstate fit due to correlated predictors. This motivated a transition to **Principal Component Regression (PCR)** to to reduce dimensionality and eliminate correlations among predictors.

Table 1: Model Performance Comparison: OLS, PCR, and Elastic Net

Model	RMSE	MAE	R2
OLS (Interaction)	2.8541	2.2507	0.8354
PCR (5 PCs)	3.1598	2.4697	0.7982
Elastic Net	2.4836	2.0283	0.8753

PCA allows the original variables to be expressed as uncorrelated linear combinations, addressing redundancy while preserving most of the original information content. The number of PCs assessed using multiple diagnostics. The scree plot revealed a steep decline after the first component, followed by a flattening trend, indicating diminishing returns in explained variance beyond the fifth component. The permutation test compared observed eigenvalues to those from randomly permuted data, confirming that the first five components captured significantly more variance than expected by chance, reflecting genuine underlying structure rather than random noise. The bootstrap boxplot of eigenvalues further verified this stability, 95% confidence intervals for the first five eigenvalues were all above one (PC1 = 11.75; PC5 = 1.37), supporting their statistical robustness under Kaiser's criterion. A bootstrap histogram of the first eigenvalue showed a narrow, symmetric distribution centered around its mean, confirming the consistency of PC1 across resamples. Finally, a Variance Accounted For (VAF) test indicated that the first five components together

explained around 75% of total variance (95% CI [73.6%, 77.9 %]). (*See Appendix 3. for details about PCs selection*). The variable contribution analysis indicated that PC1 captured socioeconomic development and health investment, combining high GNI per capita, government and private health expenditure, and access to water and sanitation with low fertility and tuberculosis rates. PC2 reflected demographic and public health infrastructure, dominated by total and urban population, immunization coverage, and hypertension prevalence. PC3–PC5 represented behavioral and epidemiological variation, including alcohol consumption, diabetes, obesity, migration, and labor factors (*Appendix 4*). To ensure direct comparability with the OLS and Elastic Net models, the Principal Component Regression was implemented using an ordinary least squares function applied to the first five principal components, rather than the pcr() function, which would internally restandardize data and apply cross-validation. The model with five retained components achieved strong predictive accuracy (RMSE = 3.16, MAE = 2.47, R^2 = 0.80) (*Figure 1*). While the R^2 was slightly lower than that of the OLS model, the MAE was reduced, indicating more stable predictions and less sensitivity to large errors. PCR model trades some R^2 for stability. However, because the dataset contains several extreme outliers (*Appendix 5*), the estimates may still be influenced by cross-country heterogeneity.

To address this limitation, the next step applied **Elastic Net** regularization, which combines the strengths of Ridge and LASSO regression to stabilize coefficients and improve robustness in the presence of correlated predictors and outliers. Unlike OLS or PCR, Elastic Net performs simultaneous variable selection and coefficient shrinkage, allowing it to handle highly correlated predictors without discarding relevant information. This makes it particularly well-suited for datasets like this one, where socioeconomic and health indicators are interrelated and may contain measurement noise or extreme country-level values. Using 10-fold cross-validation repeated five times, the model simultaneously tuned the mixing parameter (alpha) and the regularization strength (lambda). The optimal values were identified as 0.11 and 1.20, respectively. This indicates that the model relied more on Ridge-style penalty (correlation shrinkage) than LASSO-style selection, choosing stability over sparsity. The final model retained 20 non-zero coefficients, reflecting a moderate level of shrinkage. *Appendix 6* shows the predictors with non-zero weights. The most influential positive coefficients included government health expenditure per capita, GNI per capita, and access to basic water and sanitation services—all of which contribute to longer life expectancy through stronger economic capacity and public health infrastructure. Conversely, tuberculosis death rate, fertility rate, and tuberculosis incidence had the strongest negative coefficients, consistent with the detrimental impact of infectious disease burden and high fertility on longevity. Elastic Net outperformed both OLS and PCR in predictive accuracy, achieving RMSE = 2.48, MAE = 2.03, and R^2 = 0.88 (*Table 1*). These improvements highlight the model's ability to manage multicollinearity, reduce overfitting, and maintain interpretability, providing a more robust and generalizable framework for explaining cross-country variation in life expectancy.

5. Conclusions and Discussion

Across the three models, predictive performance improved with each refinement. The stepwise OLS achieved the highest in-sample R^2 = 0.84 but had higher RMSE (2.85) and MAE (2.25), indicating mild overfitting and sensitivity to multicollinearity. PCR stabilized estimation but slightly reduced explanatory power (R^2 =0.80). Elastic Net delivered the best overall balance, achieving the lowest RMSE (2.48), MAE (2.03), and highest out-of-sample R^2 =0.88, confirming its superior generalization (*Appendix 7*). After validating the Elastic Net as the best-performing model, it was applied to an unseen dataset to predict life expectancy for the Netherlands, Kenya, and Colombia. As seen in *Table 2* the model produced estimates of 82.7, 66.1, and 77.6 years, respectively, closely reflecting observed cross-country differences in socioeconomic development and healthcare infrastructure. The retained predictors—government health expenditure, GNI per capita, access to clean water, and sanitation—positively influenced longevity, while fertility and tuberculosis rates had negative effects, aligning with prior research.

While Elastic Net provided the most robust and interpretable predictions, its linear nature may still overlook nonlinear effects—such as diminishing returns of income or threshold effects in health expenditure. Future research could address these limitations by incorporating nonlinear or tree-based machine learning methods and by separating analyses for developed and developing countries to uncover structural asymmetries. Ex-

Table 2: Predicted Life Expectancy for Out-of-Sample Countries (Elastic Net Model)

Country	Predicted_Life_Expectancy
Netherlands	82.700
Kenya	66.100
Colombia	77.600

tending this framework to panel or time-series data would further allow investigation of dynamic changes in health and development over time, offering deeper insight into the determinants of global life expectancy.

References

Irandoust, Kamran, Rajabali Daroudi, Maryam Tajvar, and Mehdi Yaseri. 2025. "Global and Regional Impact of Health Determinants on Life Expectancy and Health-Adjusted Life Expectancy, 2000–2018: An Econometric Analysis Based on the Global Burden of Disease Study 2019." *Frontiers in Public Health* 13. https://doi.org/10.3389/fpubh.2025.1566469.

James, Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. 2023. *An Introduction to Statistical Learning: With Applications in r*. 3rd ed. New York: Springer. https://www.statlearning.com/.

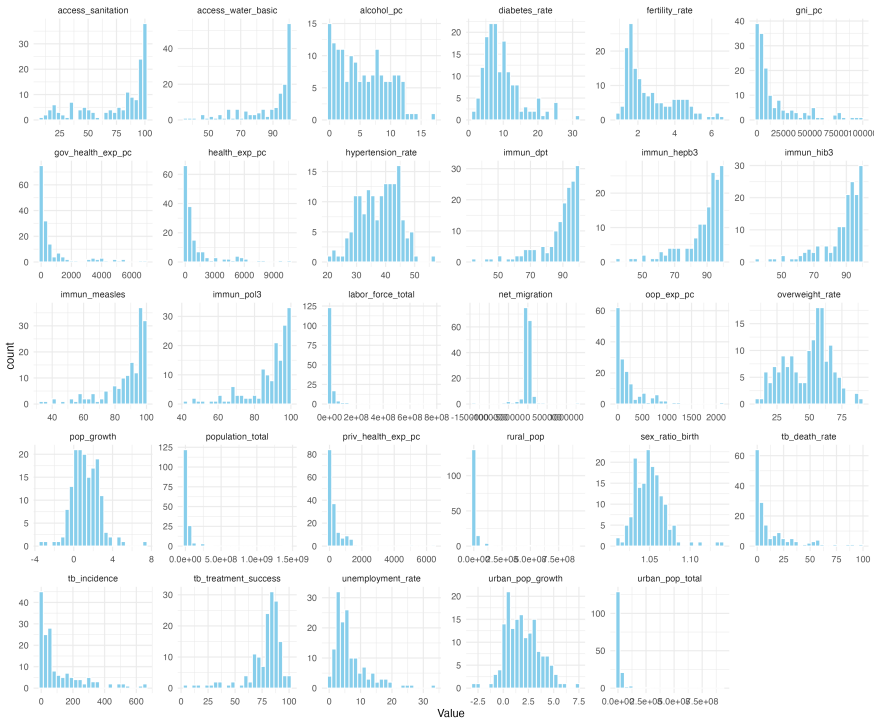
Karlsson, Omar, Angela Y. Chang, Ole F. Norheim, et al. 2025. "Priority Health Conditions and Global Life Expectancy Disparities." *JAMA Network Open* 8 (5): e2512198. https://doi.org/10.1001/jamanetworkopen.2025.12198.

Mondal, Md. Nazrul Islam, Abu Naser Muhammad Abdul Baki, Md. Nazrul Hoque, Hafiz T. A. Khan, and Md. Nuruzzaman Khan. 2019. "Exploring the Determinants of Global Life Expectancy from an Ecological Perspective." *Turkish Journal of Public Health* 17 (3): 314–25. https://doi.org/10.20518/tjph.452721.

Roffia, Paolo, Alessandro Bucciol, and Sara Hashlamoun. 2023. "Determinants of Life Expectancy at Birth: A Longitudinal Study on OECD Countries." *International Journal of Health Economics and Management* 23 (2): 189–212. https://doi.org/10.1007/s10754-022-09338-5.

Appendix

Appendix 1. Variable Distributions



Appendix 2. OLS Stepwise Regression

2.1 Stepwise OLS Coefficients

Table 3: OLS Stepwise Regression Results

	Coefficients Summary				
	Variable	Estimate	Std..Error	t.value	Pr...t..
(Intercept)	(Intercept)	72.086	0.265	271.85	0.0000
fertility_rate	fertility_rate	-1.684	0.541	-3.11	0.0023
health_exp_pc	health_exp_pc	-4.918	1.464	-3.36	0.0011
gov_health_exp_pc	gov_health_exp_pc	5.454	1.337	4.08	0.0001
gni_pc	gni_pc	1.498	0.650	2.31	0.0229
immun_pol3	immun_pol3	1.127	0.369	3.06	0.0028
tb_incidence	tb_incidence	-1.047	0.658	-1.59	0.1144
access_water_basic	access_water_basic	1.373	0.416	3.30	0.0013
hypertension_rate	hypertension_rate	-0.778	0.292	-2.67	0.0088
rural_pop	rural_pop	-1.050	0.423	-2.48	0.0145
sex_ratio_birth	sex_ratio_birth	0.675	0.309	2.18	0.0312
tb_death_rate	tb_death_rate	-1.474	0.714	-2.06	0.0412
urban_pop_total	urban_pop_total	1.475	0.393	3.76	0.0003

2.2 10-Fold Cross-Validation Results

Table 4: 10-Fold Cross-Validation Results for OLS Stepwise Model

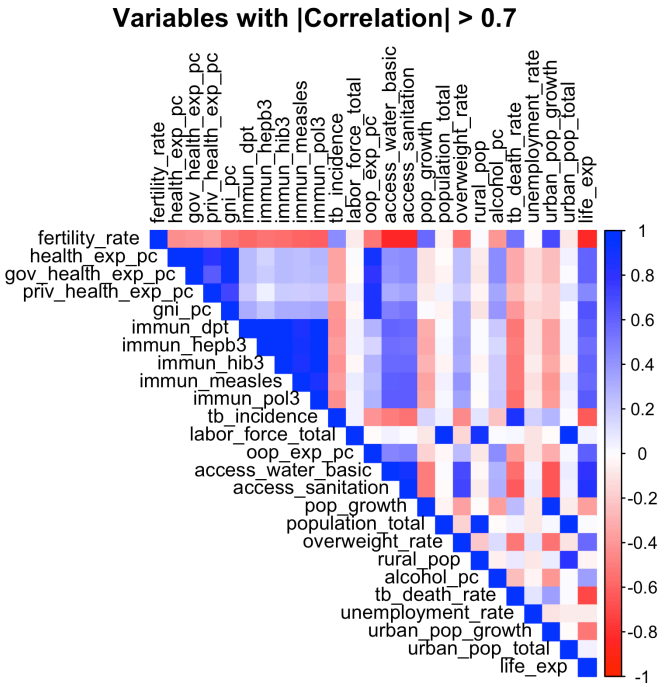
RMSE	R^2	MAE	SD(RMSE)	SD(R^2)	SD(MAE)
3.09	0.887	2.386	1.345	0.055	0.876

2.3 Variance Inflation Factors (VIF)

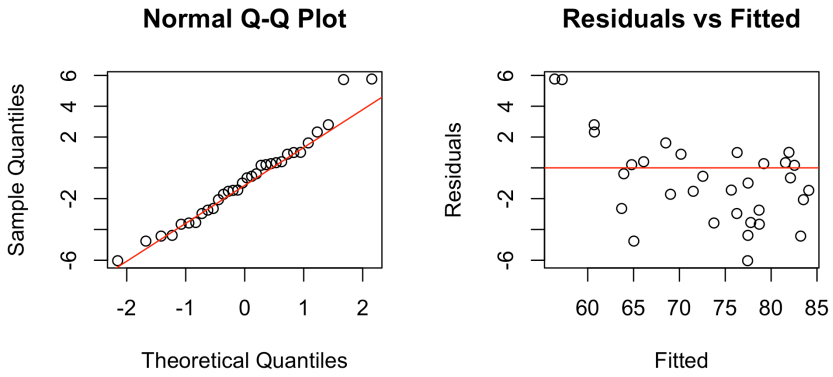
Table 5: Variance Inflation Factors (VIF) for OLS Predictors

	Variable	VIF
1	fertility_rate	4.130
2	health_exp_pc	30.240
3	gov_health_exp_pc	25.210
4	gni_pc	5.950
5	immun_pol3	1.920
6	tb_incidence	6.110
7	access_water_basic	2.440
8	hypertension_rate	1.200
9	rural_pop	2.520
10	sex_ratio_birth	1.350
11	tb_death_rate	7.190
12	urban_pop_total	2.180

2.4 Correlation Matrix

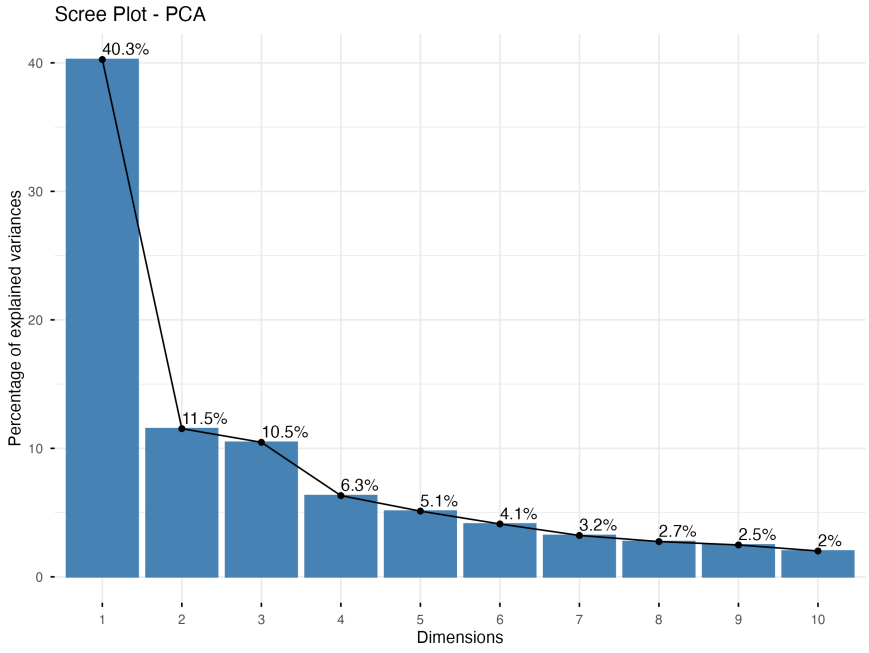


2.5 Residual Diagnostics

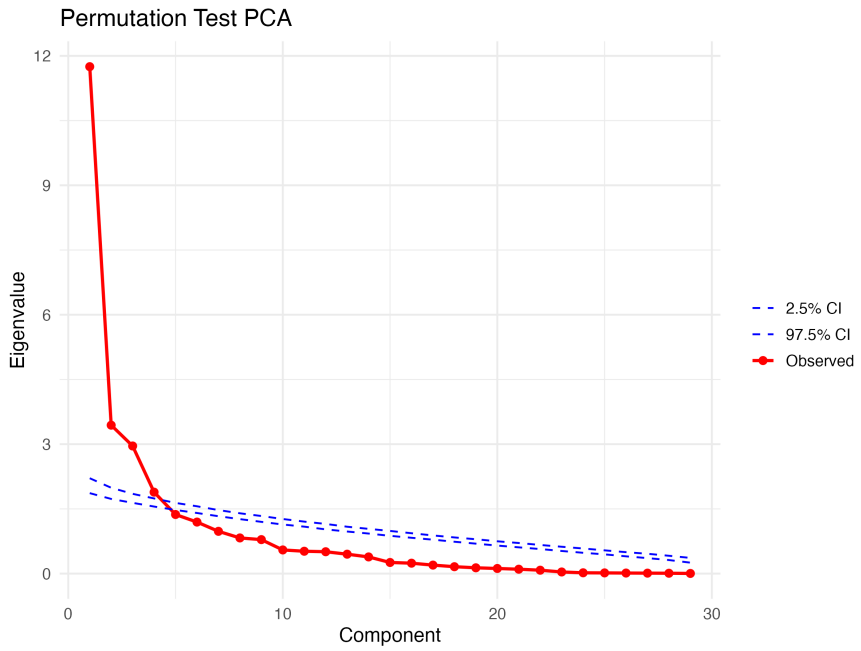


Appendix 3. Principal Component Analysis (PCA)

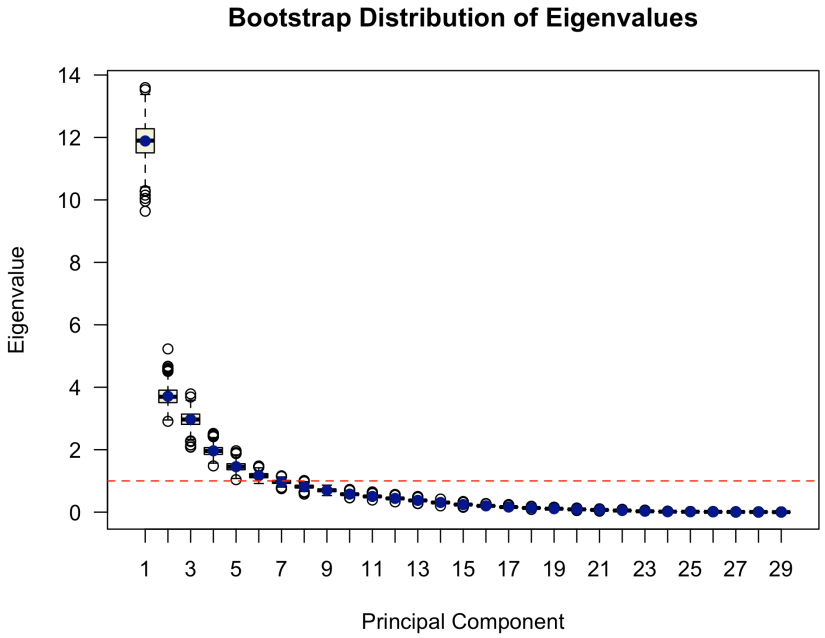
3.1 Scree Plot



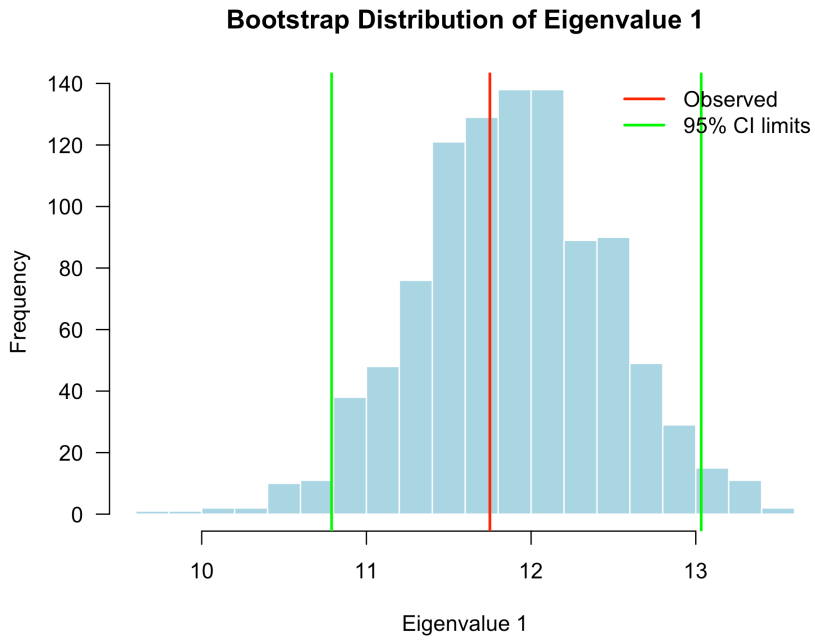
3.2 Permutation Test (Observed vs Random Eigenvalues)



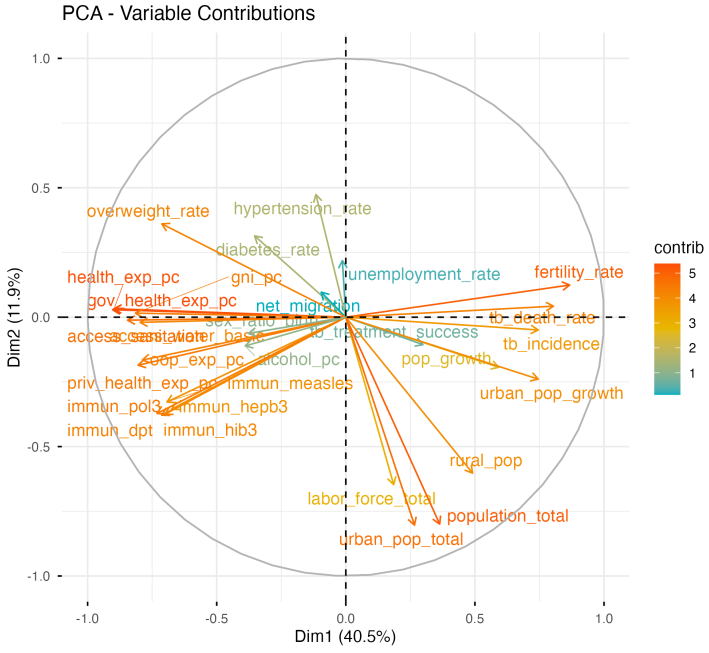
3.3 Bootstrap Boxplot of Eigenvalues



3.4 Bootstrap Histogram (First Eigenvalue)

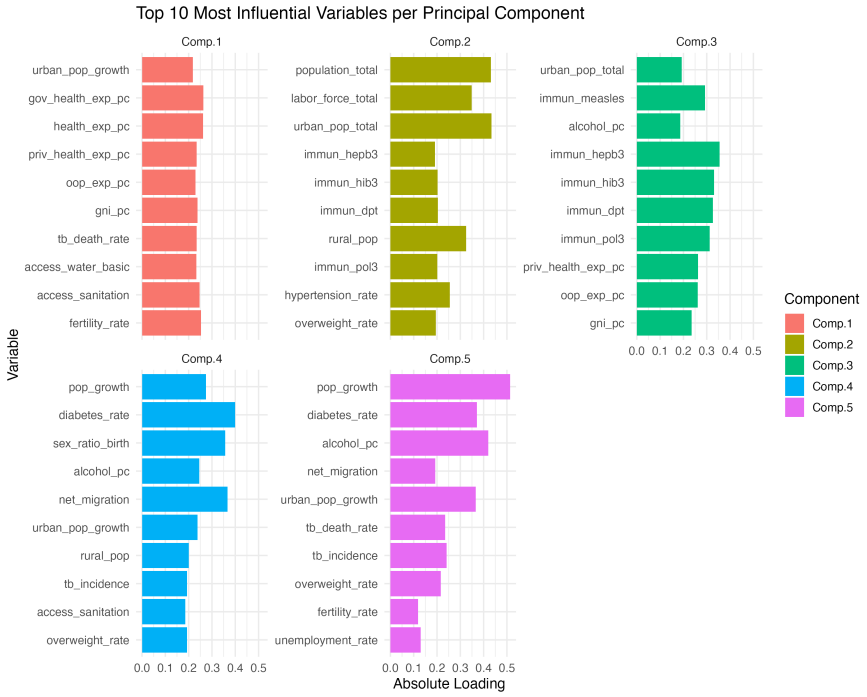


3.5 Variable Contribution Plot

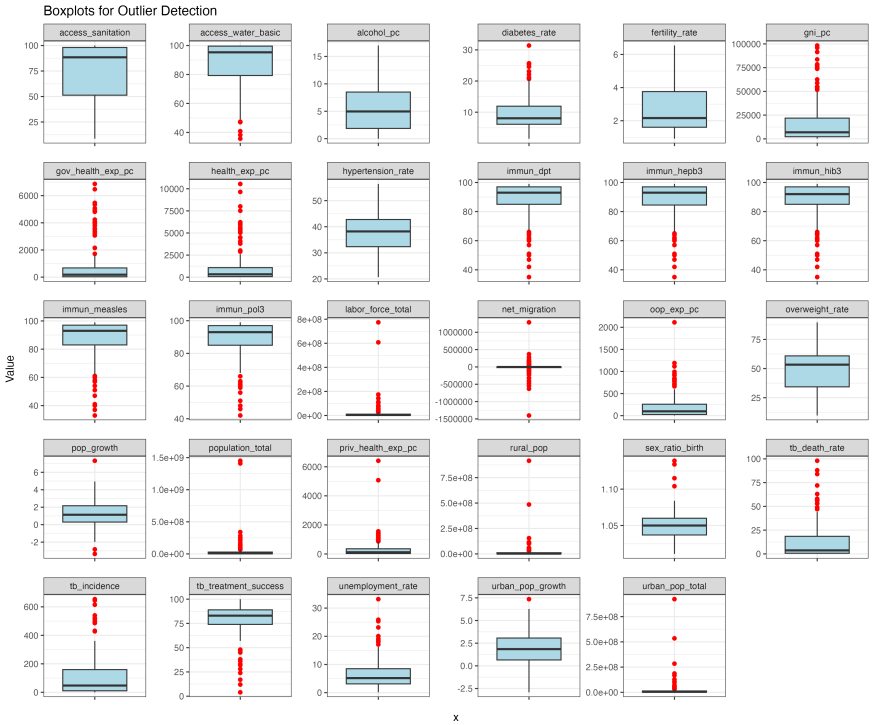


Appendix 4. PCA Loadings

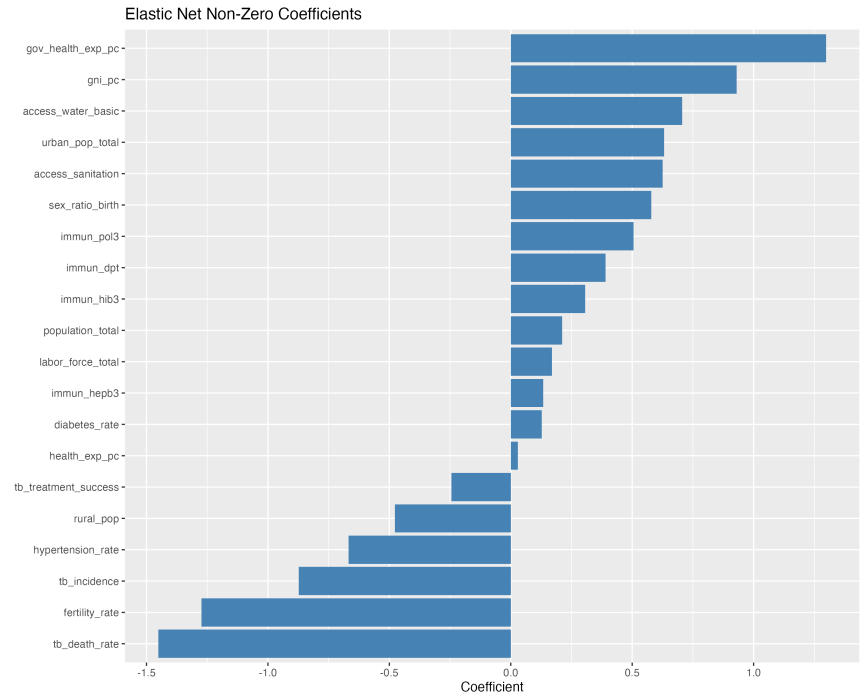
4.1 Top 10 Variables per Principal Component



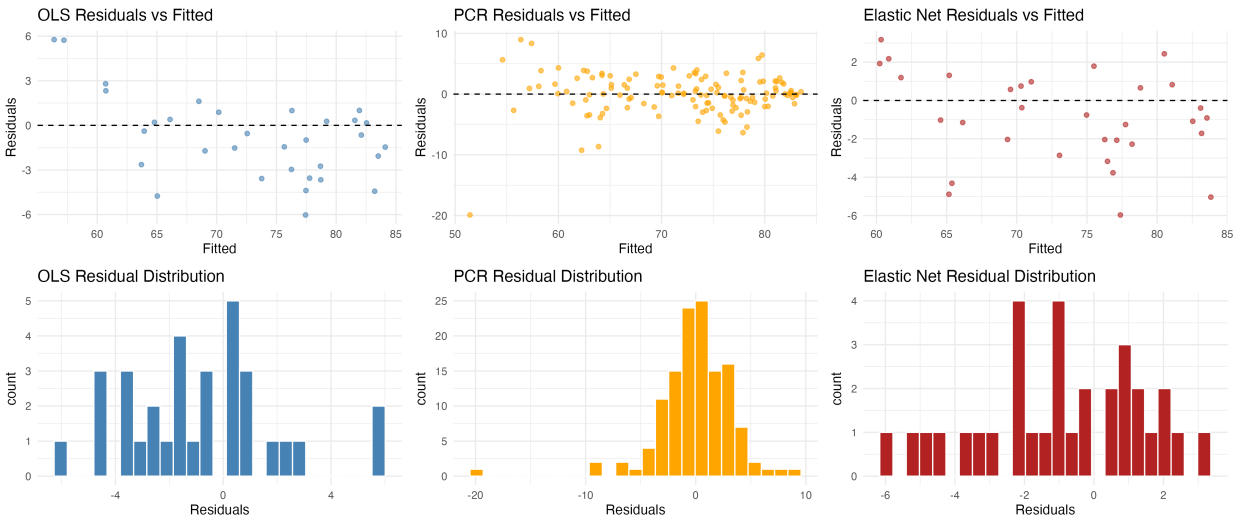
Appendix 5. Outlier Detection



Appendix 6. Elastic Net Coefficients



Appendix 7. Residual Diagnostics



Appendix 8. Modelling - Code

```r
# Full reproducible R code excerpt - descriptive preprocessing removed for brevity: still visible in .R
# ---------------- Helper Function: Pre- processing & Train/Test Split ----------------
to_log <- c("gni_pc", "gov_health_exp_pc", "health_exp_pc", "priv_health_exp_pc", "oop_exp_pc", "popula
set.seed(123)
idx_global <- createDataPartition(data_full$life_exp, p = 0.8, list = FALSE)
train_global <- data_full[idx_global, ]
test_global  <- data_full[-idx_global, ]
preprocess_data <- function(train_raw, test_raw, to_log, target = "life_exp") {
  train_medians <- train_raw %>% summarise(across(where(is.numeric), ~ median(.x, na.rm = TRUE))) # Imp
  impute_fun <- function(df) mutate(df, across(where(is.numeric), ~ ifelse(is.na(.x), train_medians[[cu
  train <- impute_fun(train_raw)
  test  <- impute_fun(test_raw)
  # Log transform (training reference)
  log_fun <- function(df, ref) mutate(
    df,
    across(all_of(to_log),
           ~ log1p(.x - min(ref[[cur_column()]], na.rm = TRUE) + 1)))
  train <- log_fun(train, train)
  test  <- log_fun(test, train)
  # Scale predictors only
  pred_cols <- setdiff(names(train), target)
  scale_fun <- function(df, ref) mutate(
    df,
    across(all_of(pred_cols),
           ~ (.x - mean(ref[[cur_column()]], na.rm = TRUE)) /
             sd(ref[[cur_column()]], na.rm = TRUE)))
  train_scaled <- scale_fun(train, train)
  test_scaled  <- scale_fun(test, train)
  # Reattach target (unscaled)
  train_scaled[[target]] <- train[[target]]
  test_scaled[[target]]  <- test[[target]]
  list(train = train_scaled, test = test_scaled)
}
splits_all <- preprocess_data(train_global, test_global, to_log)
train_all <- splits_all$train
test_all  <- splits_all$test
# ------------------- BASELINE: OLS -------------------
ols_full <- lm(life_exp ~ ., data = train_all)
ols_step <- stepAIC(ols_full, direction = "both", trace = FALSE)
step_pred  <- predict(ols_step, newdata = test_all)
step_resid <- test_all$life_exp - step_pred
par(mfrow = c(1,2)) # --- OLS Diagnostics ---
qqnorm(step_resid); qqline(step_resid, col="red")
plot(step_pred, step_resid, main="Residuals vs Fitted", xlab="Fitted", ylab="Residuals")
abline(h=0, col="red")
par(mfrow = c(1,1))
step_metrics <- list(
  rmse = sqrt(mean(step_resid^2)),
  mae  = mean(abs(step_resid)),
  r2   = 1 - sum(step_resid^2) /
    sum((test_all$life_exp - mean(test_all$life_exp))^2))
```

```r
set.seed(123)
cv_ols_step <- train(life_exp ~ ., data = train_all[, c(all.vars(formula(ols_step)))],
                     method = "lm", trControl = trainControl(method = "cv", number = 10))
# ------------------- PCA & PRINCIPAL COMPONENT REGRESSION ------------------
X_train <- train_all %>% dplyr::select(-life_exp)
y_train <- train_all$life_exp
X_test  <- test_all %>% dplyr::select(-life_exp)
y_test  <- test_all$life_exp
X_train <- X_train[, sapply(X_train, function(x) sd(x, na.rm = TRUE) > 0)]
X_test  <- X_test[, names(X_train)]
X_train[is.na(X_train)] <- 0
X_test[is.na(X_test)] <- 0
res_pca <- princomp(X_train, cor = TRUE, scores = TRUE)
p_scree <- fviz_eig(res_pca, addlabels = TRUE, main = "Scree Plot - PCA (Training Data)")
ggsave("pca_scree_plot_corrected.png", plot = p_scree, width = 8, height = 6, dpi = 300)
# --- 1. Permutation Test (compare observed vs random eigenvalues) ---
permtestPCA <- function(data, nperm = 1000) {
  set.seed(123)
  eigen_real <- eigen(cor(data, use = "pairwise.complete.obs"))$values
  eigen_perm <- replicate(nperm, {
    perm_data <- apply(data, 2, sample)
    eigen(cor(perm_data, use = "pairwise.complete.obs"))$values})
  ci_lower <- apply(eigen_perm, 1, quantile, 0.025)
  ci_upper <- apply(eigen_perm, 1, quantile, 0.975)
  df_perm <- data.frame(
    Component = 1:length(eigen_real),
    Observed = eigen_real,
    Lower = ci_lower,
    Upper = ci_upper)
  ggplot(df_perm, aes(x = Component)) +
    geom_line(aes(y = Observed, color = "Observed"), linewidth = 1) +
    geom_point(aes(y = Observed, color = "Observed"), size = 2) +
    geom_line(aes(y = Lower, color = "2.5% CI"), linetype = "dashed") +
    geom_line(aes(y = Upper, color = "97.5% CI"), linetype = "dashed") +
    scale_color_manual(values = c("Observed" = "red", "2.5% CI" = "blue", "97.5% CI" = "blue")) +
    labs(title = "Permutation Test PCA", x = "Component", y = "Eigenvalue", color = "") +
    theme_minimal(base_size = 13)}
# --- 2. Bootstrap Test for Eigenvalues (stability + Kaiser rule) ---
my_boot_pca <- function(x, ind) {
  res <- princomp(x[ind, ], cor = TRUE)
  res$sdev^2}
set.seed(123)
fit.boot <- boot(data = X_train, statistic = my_boot_pca, R = 1000)
eigs.boot <- fit.boot$t
obs_eigs <- res_pca$sdev^2
# Plot bootstrap distributions
boxplot(eigs.boot, col = "beige", las = 1,
        main = "Bootstrap Distribution of Eigenvalues",
        xlab = "Principal Component", ylab = "Eigenvalue")
points(colMeans(eigs.boot), pch = 19, col = "darkblue")
abline(h = 1, col = "red", lty = 2)
# --- Histogram of Bootstrap Distribution for the First Eigenvalue ---
hist(eigs.boot[, 1],
```

```
     xlab = "Eigenvalue 1",
     main = "Bootstrap Distribution of Eigenvalue 1",
     col = "lightblue", border = "white",
     las = 1, breaks = 25)
perc.alpha <- quantile(eigs.boot[, 1], c(0.025, 0.975))
abline(v = perc.alpha, col = "green", lwd = 2)
abline(v = obs_eigs[1], col = "red", lwd = 2)
legend("topright", legend = c("Observed", "95% CI limits"),
       col = c("red", "green"), lty = 1, lwd = 2, bty = "n")
# Compute 95% CI for each eigenvalue
ci_eigs <- apply(eigs.boot, 2, quantile, c(0.025, 0.975))
ci_table <- data.frame(
  Component = 1:length(obs_eigs),
  Observed = round(obs_eigs, 3),
  CI_Lower = round(ci_eigs[1, ], 3),
  CI_Upper = round(ci_eigs[2, ], 3))
# --- 3. Bootstrap Test for total variance  70% (VAF test) ---
boot_vaf <- apply(eigs.boot, 1, function(x) sum(x[1:5]) / sum(x))
ci_vaf <- quantile(boot_vaf, c(0.025, 0.975))
# --- 4. Decide number of components to retain ---
n_components <- sum(ci_eigs[1, ] > 1)
n_components <- 5
# Project data onto the first 5 PCs
pc_train <- predict(res_pca, newdata = X_train)
pc_test  <- predict(res_pca, newdata = X_test)
train_scores <- as.data.frame(pc_train[, 1:n_components, drop = FALSE])
test_scores  <- as.data.frame(pc_test[, 1:n_components, drop = FALSE])
train_scores$life_exp <- y_train
test_scores$life_exp  <- y_test
# Fit regression on principal components
pcr_model <- lm(life_exp ~ ., data = train_scores)
pcr_pred <- predict(pcr_model, newdata = test_scores)
pcr_resid <- test_scores$life_exp - pcr_pred
# Performance metrics
pcr_rmse <- sqrt(mean(pcr_resid^2))
pcr_mae  <- mean(abs(pcr_resid))
pcr_r2   <- 1 - sum(pcr_resid^2) / sum((y_test - mean(y_test))^2)
# Top contributing variables per component (PC1-PC5)
loadings_df <- as.data.frame(res_pca$loadings[, 1:n_components]) %>%
  rownames_to_column("Variable") %>%
  pivot_longer(-Variable, names_to = "Component", values_to = "Loading") %>%
  mutate(AbsLoading = abs(Loading)) %>%
  group_by(Component) %>%
  slice_max(order_by = AbsLoading, n = 10) %>%
  ungroup()
p_top_vars <- ggplot(loadings_df, aes(x = reorder(Variable, AbsLoading)))
# ---------------- Elastic Net ----------------------
ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 5)
alpha_grid <- seq(0, 1, length = 10)
lambda_seq <- 10^seq(2, -3, length = 100)
set.seed(123)
elasticNet <- train(
  life_exp ~ ., data = train_all,
```

```
  method = "glmnet",
  tuneGrid = expand.grid(alpha = alpha_grid, lambda = lambda_seq),
  trControl = ctrl, standardize = TRUE)
best <- elasticNet$bestTune
final_en <- glmnet(
  x = as.matrix(dplyr::select(train_all, -life_exp)),
  y = train_all$life_exp,
  alpha = best$alpha, lambda = best$lambda, standardize = TRUE)
pred_en <- as.numeric(predict(final_en, newx = as.matrix(dplyr::select(test_all, -life_exp))))
enet_resid <- test_all$life_exp - pred_en
print(best$alpha)
print(best$lambda)
en_metrics <- list(
  rmse = sqrt(mean(enet_resid^2)),
  mae  = mean(abs(enet_resid)),
  r2   = 1 - sum(enet_resid^2) / sum((test_all$life_exp - mean(test_all$life_exp))^2))
# View important variables
enet_coef <- coef(final_en)
coef_df <- data.frame(
  Variable = rownames(enet_coef),
  Coefficient = as.numeric(enet_coef)) %>%
  filter(Coefficient != 0 & Variable != "(Intercept)") %>%
  arrange(desc(abs(Coefficient)))
# ------------------- MODEL PERFORMANCE COMPARISON -------------------
comparison <- tibble(
  Model = c("OLS (Interaction)", "PCR (5 PCs)", "Elastic Net"),
  RMSE  = c(ols_metrics$rmse, pcr_rmse, en_metrics$rmse),
  MAE   = c(ols_metrics$mae,  pcr_mae,  en_metrics$mae),
  R2    = c(ols_metrics$r2,   pcr_r2,   en_metrics$r2)) %>%
  mutate(across(where(is.numeric), ~ round(.x, 4)))
# --------- Predicting on unseen data -------
new_data_split <- preprocess_data(train_global, predictions, to_log)
pred_final <- new_data_split$test
pred_final_aligned <- pred_final[, names(dplyr::select(train_all, -life_exp))]
final_predictions <- predict(
  final_en,
  newx = as.matrix(pred_final_aligned),
  s = best$lambda)
predicted_df <- data.frame(
  Country = predictions$Country,
  Predicted_Life_Expectancy = round(as.numeric(final_predictions), 1))
```