

Session 2: Gap estimation Practical



UNU
WIDER

Break out groups:

Maria Jousté + Amina Ebrahim + Sebastian Castillo

Kwabena Adu-Ababio + Karri Vuoristo

ATI Tax Gap Estimation Toolkit | Follow-up Training | May 7, 2025

Aim of this session

1. How to predict tax recovered from unaudited taxpayers
2. Show how to execute steps in the Toolkit Part 2 and estimate the tax gap.
3. Produce disaggregated results
4. Show how to investigate the effects of aggregate tax gaps.

We are not teaching how to use Stata merely guiding the steps to follow.

Own implementation requires contextualizing to the tax type (VAT, CIT, PAYE), and data structure and challenges.

From Part 1 to Part 2

At the end of Toolkit Part 1, you have a cleaned and matched return-audit dataset.

This dataset is the starting point for Toolkit Part 2.

In this practical, we provide the cleaned and matched VAT return-audit dataset. The VAT is used for demonstration purposes.

Exercise 1– Stata Environment setup & data import

In this exercise, you will try to implement the First Stage of the Toolkit Part 2. Ensure the random forest machine learning algorithm is installed in Stata, prepare the Stata environment, set up the global directory paths and import the data.

- a) Run the code lines 25 to 30 to prepare the Stata environment.
- b) Modify line 33 to your own file directory and run lines 34 & 35 to set the directory paths for the data and results. Open the vat_audit Stata data (.dta file)

Exercise 1 – Solutions

a) Run the code lines 25 to 30 to prepare the Stata environment.

```
set more off  
clear all  
clear matrix  
set linesize 200  
set matsize 800  
set seed 201807
```

a) Modify line 33 to your own file directory and run lines 34 & 35 to set the directory paths for the data and results. Open the vat_audit Stata data (.dta file)

```
global dir "<YOUR PATH>"  
global data "$dir\Data"  
global results "$dir\Results"  
use "$data\vat_audit_new.dta", clear // Insert the database  
to use
```

Exercise 2 – Choosing the machine learning model

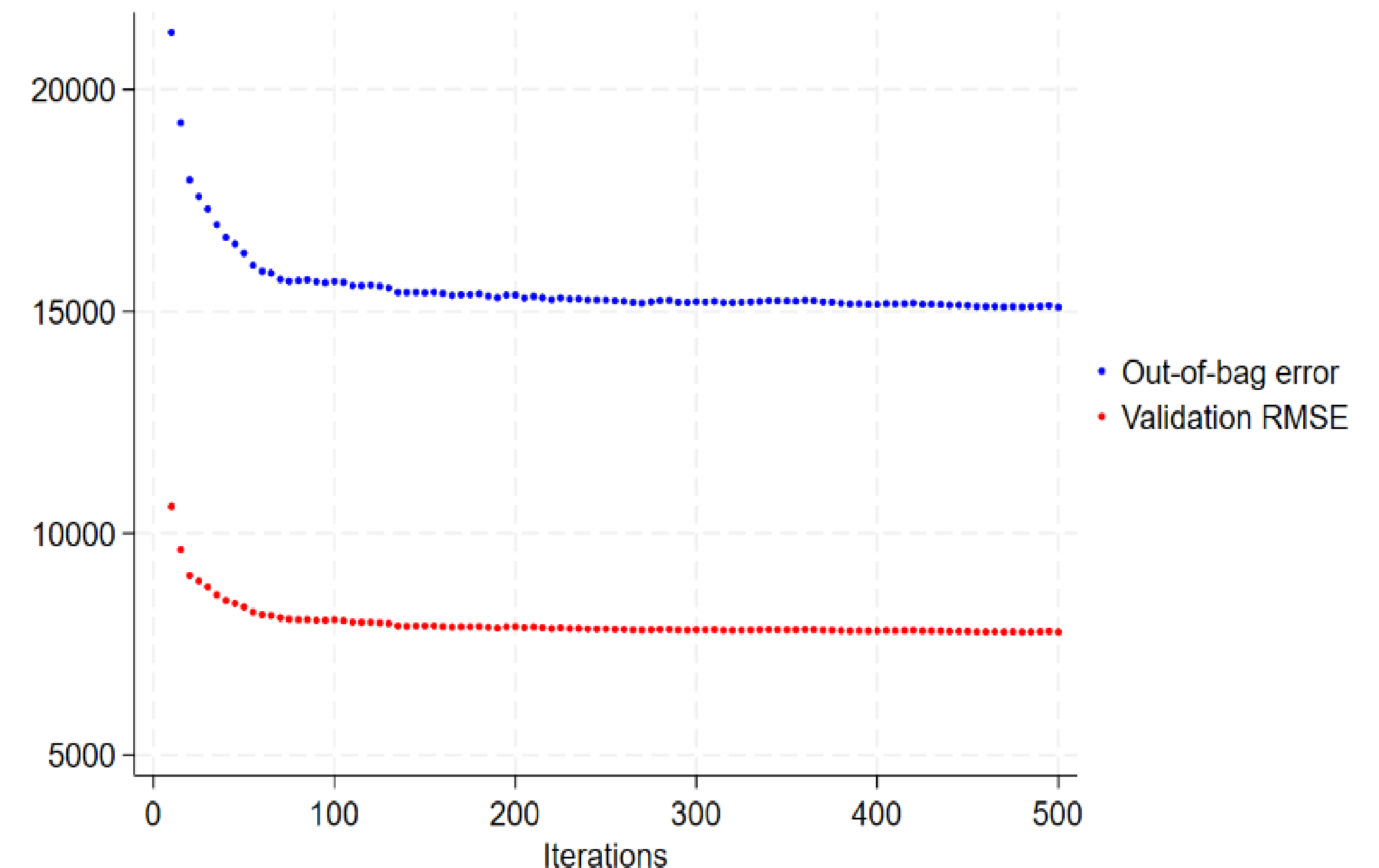
In this exercise, you will try to implement the Second Stage of the Toolkit Part 2. Set up parameters, split the data, tune parameter one (number of iterations), tune parameter (number of variables).

- a) Create a dummy variable "reg_audit" that indicates if a taxpayer is audited or not (0/1).
- b) Create an "evaded" variable with the specific liability amount from the audit outcome. Keep observations with audit outcomes.
- c) Split the sample (50/50) into two groups. This forms the training sample and the testing sample.

Exercise 2 – Choosing the machine learning model

d) In machine learning, the “number of iterations” (the number of times model parameters are updated during training) is what’s called a hyperparameter. Setting this value appropriately is important to prevent overfitting, where the model learns the training data too well but performs poorly on new, unseen data. Typically, as iterations increase, the model’s error on validation data (like “Out-of-bag error” or “Validation RMSE”) will decrease significantly at first. However, after a certain point, the error will stabilize, and further iterations may offer no substantial improvement or could even lead to worse generalization. The provided figure illustrates how the model’s “out-of-bag error” and “Validation RMSE” change with an increasing number of iterations.

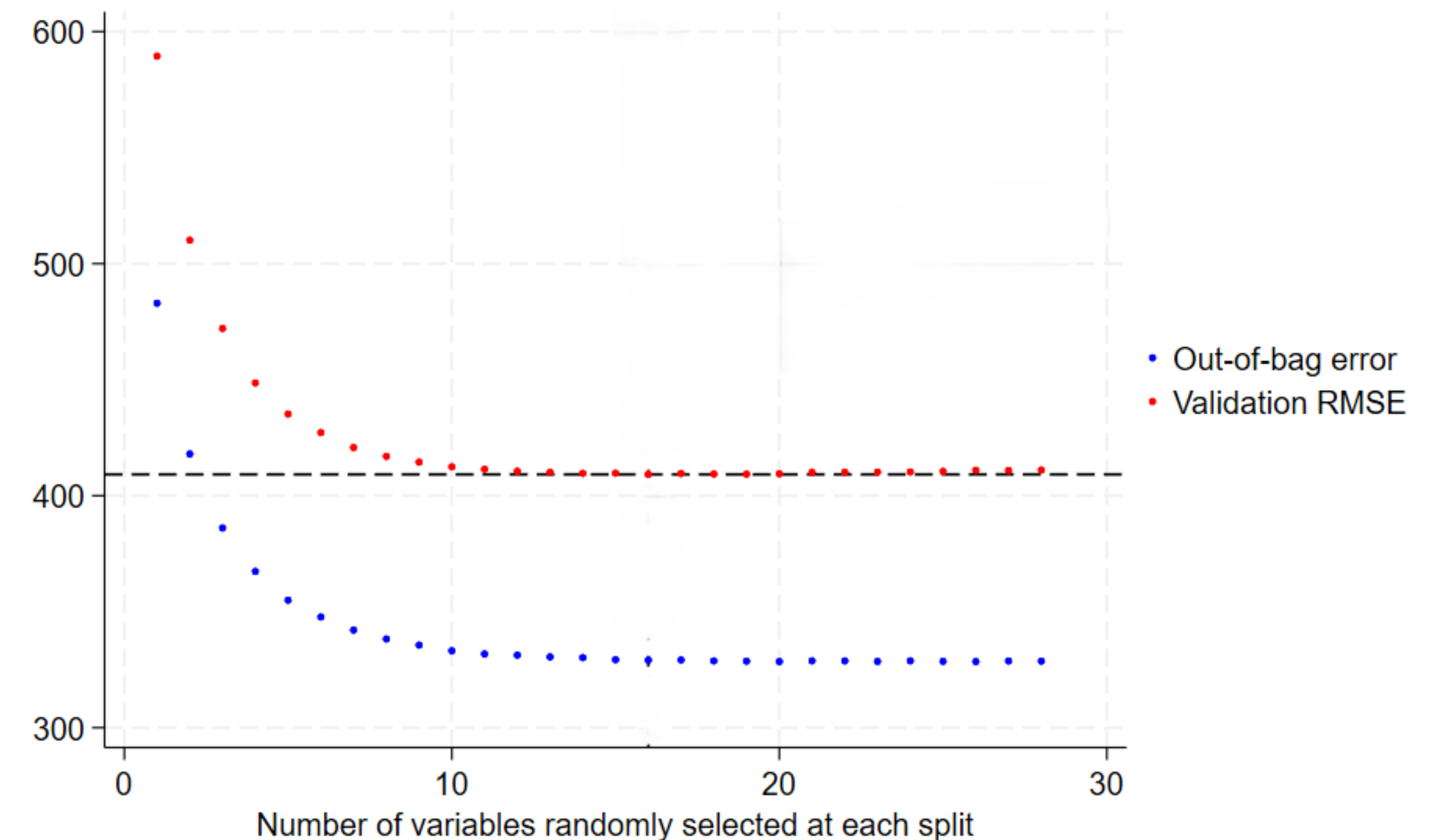
At approximately what number of iterations do both metrics level out? How many iterations should we train our model on?



Exercise 2 – Choosing the machine learning model

e) During model training, random forest will iteratively select a random subset of variables, often called “features”. While the model has a mechanism to assess and weigh the importance of each feature for its predictions, a challenge can emerge related to the sheer number of features used. Similar to how excessive iterations can be detrimental, including too many features, even if their importance is evaluated, can negatively affect the model via overfitting.

The following figure shows the model's performance over the number of variables. From the graph, select a number where it appears the Out-of-bag error and Validation RMSE stabilise. The value is used as the number of variables in the prediction model in the next exercise.



Exercise 2 – Solutions

- a) Create a dummy variable "reg_audit" that indicates if a taxpayer is audited or not (0/1).

```
generate reg_audit = (audit==1)
tab reg_audit
```

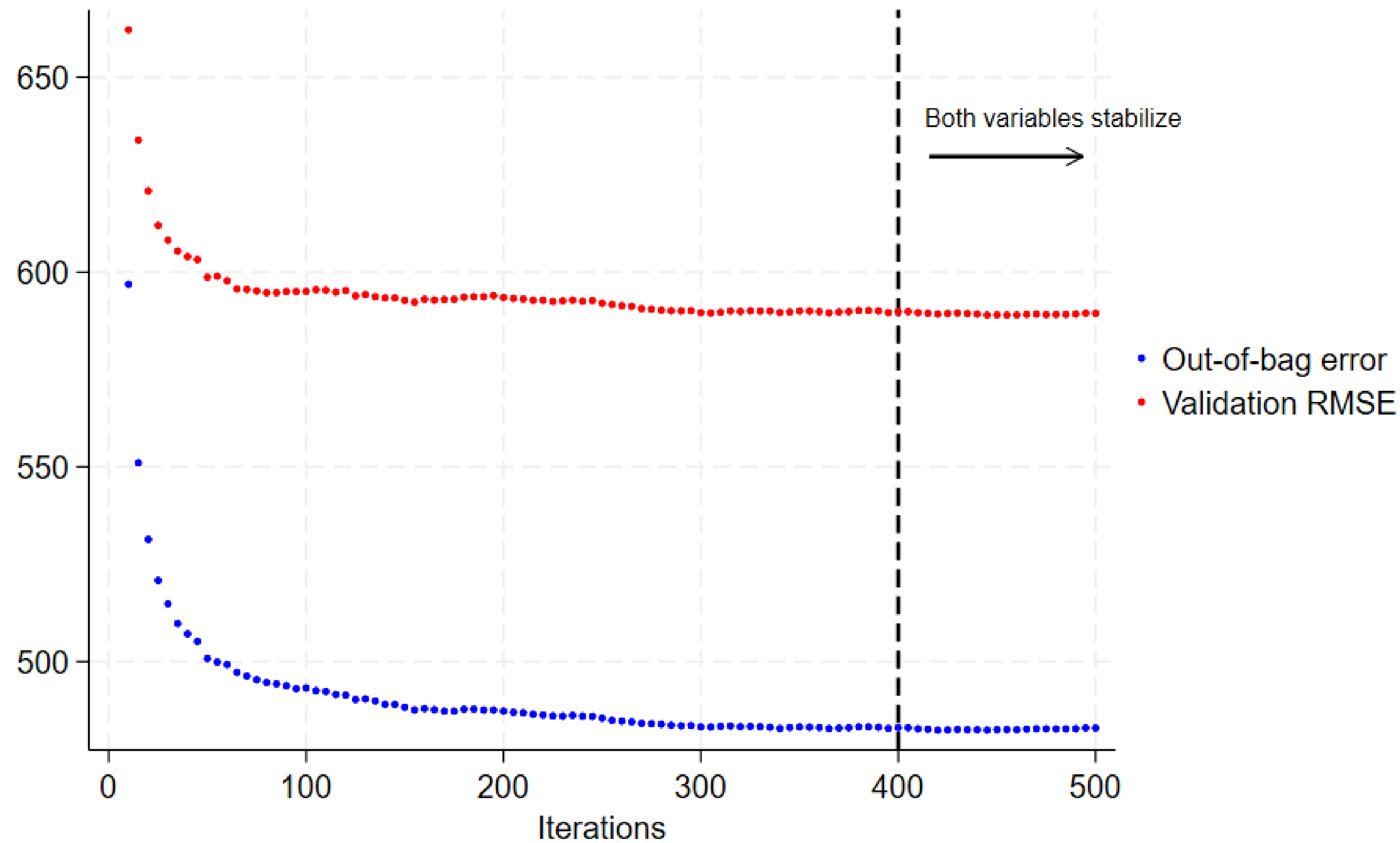
- b) Create an "evaded" variable with the specific liability amount from the audit outcome. Keep observations with audit outcomes.

```
generate evaded = vat_evasion
keep if evaded !=. & reg_audit==1
```

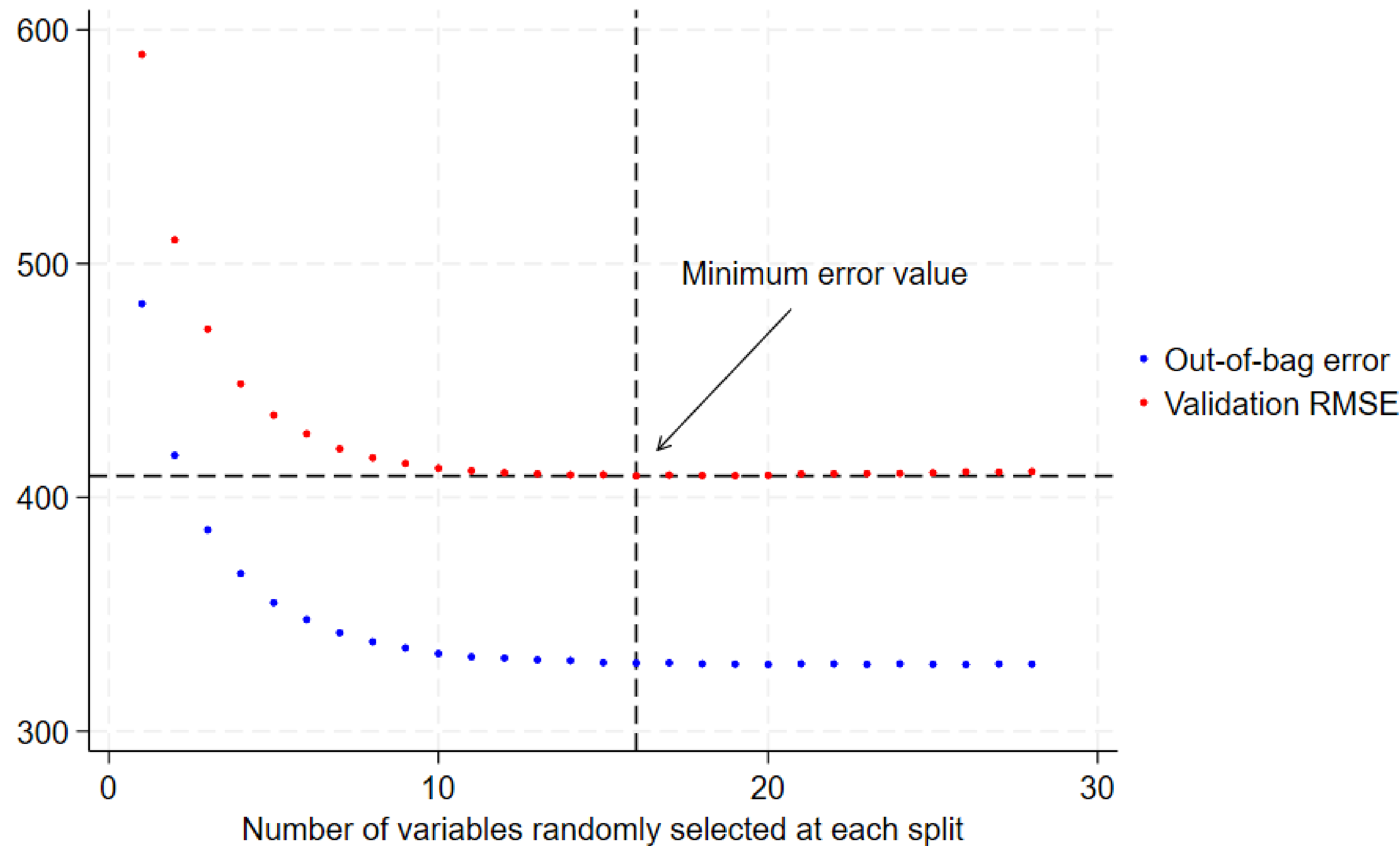
- c) Split the sample (50/50) into two groups. This forms the training sample and the testing sample.

```
split sample, gen(u) split(1 1) balance(evaded) rround
sort u, stable
```

Exercise 2 – Solutions



Exercise 2 – Solutions



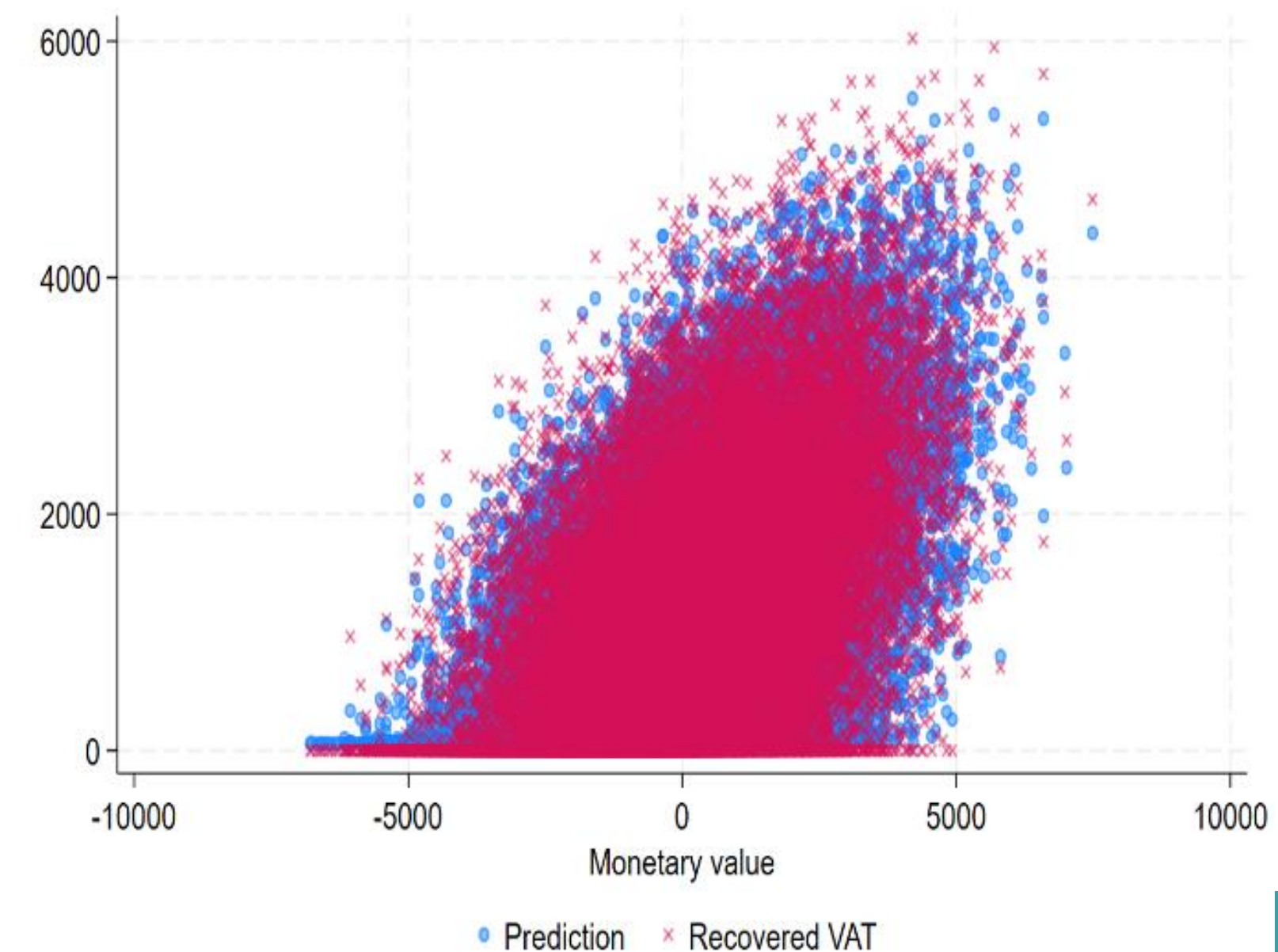
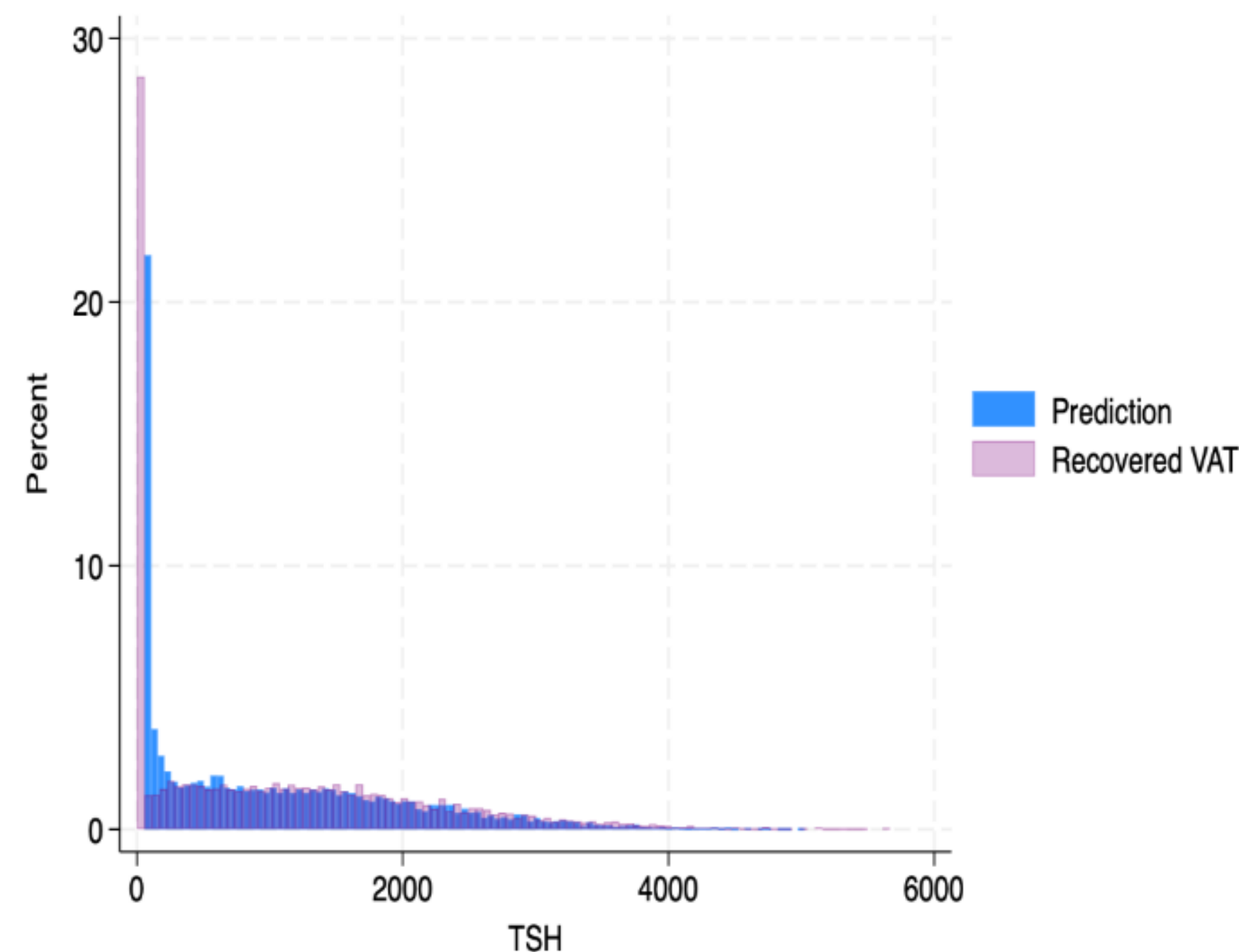
Exercise 3 – Testing the model

In this exercise, you will try to implement the Third Stage of the Toolkit Part 2. Before applying the model to the full dataset, the model is tested and checked.

- a) Run the random forest model with the parameters identified in Exercise 2 on the training sample.

Exercise 3 – Testing the model

- b) During model evaluation, the toolkit generates diagnostic plots that track prediction accuracy as model complexity changes. The following figure shows the model's accuracy by comparing predicted and actual assessment values. What should you be looking for in these graphs to confirm that the prediction is accurate?



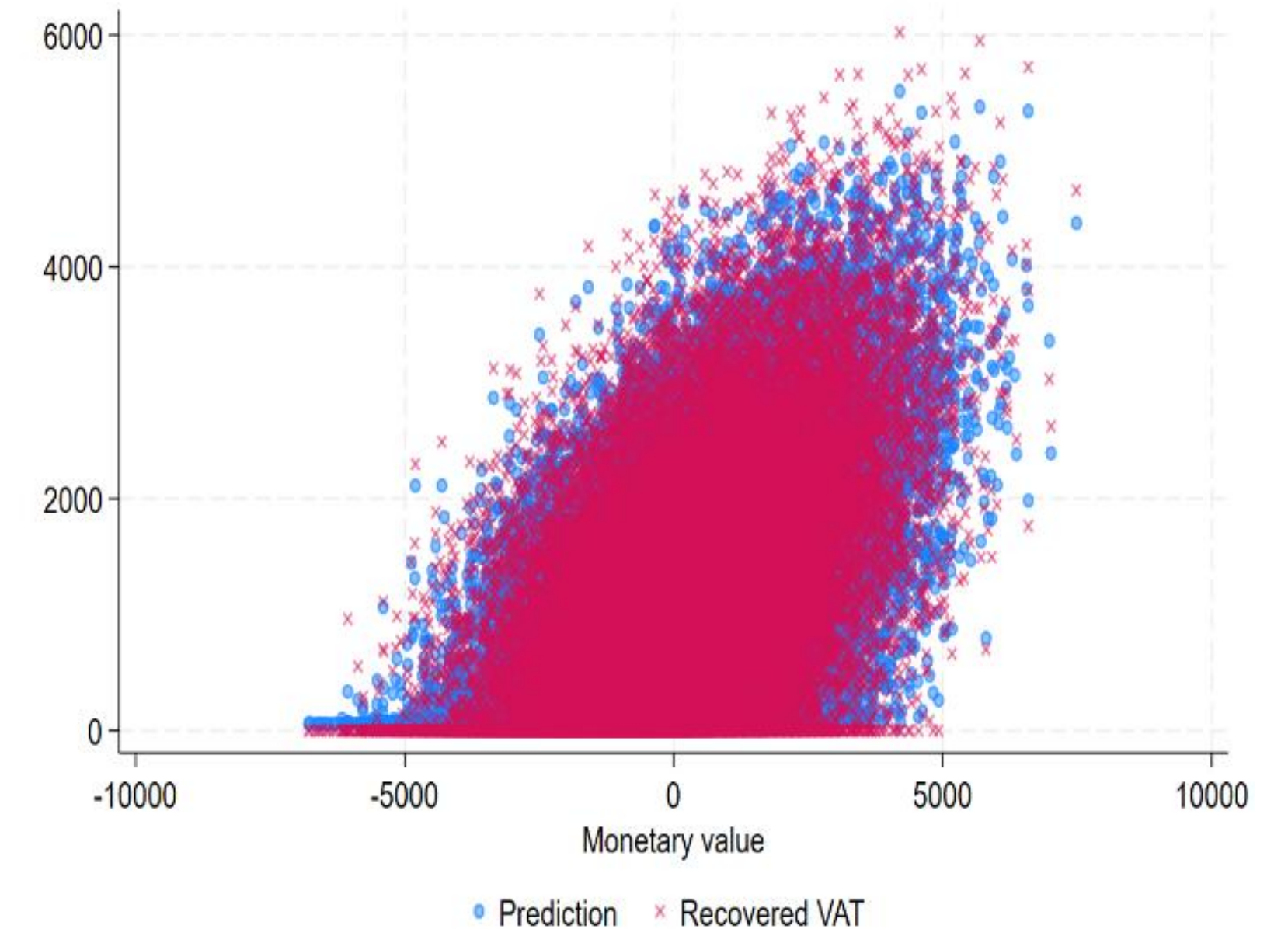
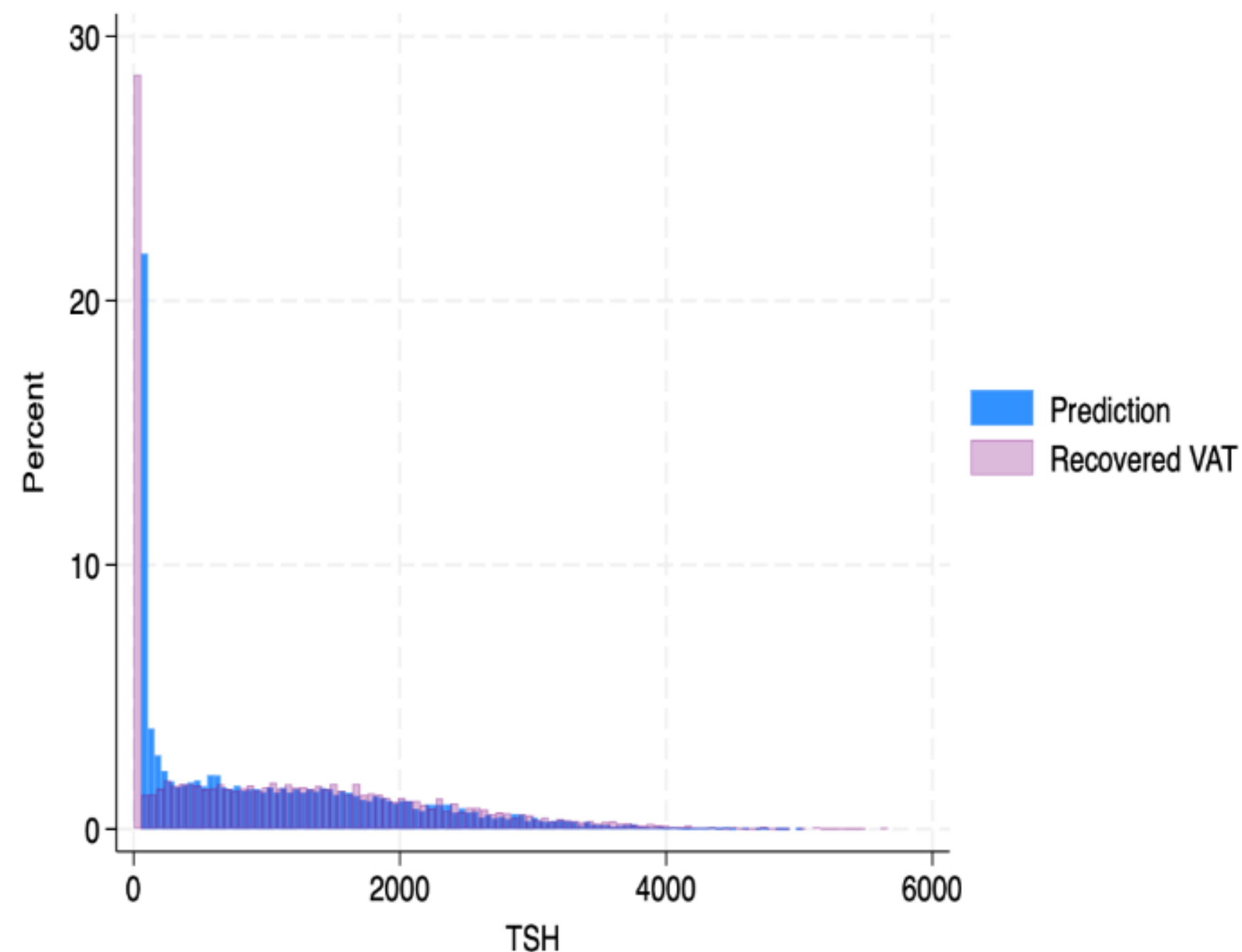
Exercise 3 – Solutions

- a) Run the random forest model with the parameters identified in Exercise 2 on the training sample.

```
rforest evaded rtn_year tax_sales vat_tax_sales zero_rate_sales exempt_sales  
import_sales vat_import_sales export_sales capital_sales nondeduc_sales  
taxunregistered_sales vat_taxunregistered_sales total_sales total_vat_sales  
tax_purchases vat_tax_purchases zero_rate_purchases exempt_purchases  
import_purchases vat_import_purchases export_purchases capital_purchases  
nondeduc_purchases taxunregistered_purchases vat_taxunregistered_purchases  
total_purchases total_vat_purchases vat_declared city region isic sector  
tax_region if u==1, type(reg) iter(400) numvars(18)  
  
ereturn list
```

Exercise 3 – Solutions

Answer: The histogram shows the distribution of predicted values (in blue) and actual recovered VAT values (in purple). The closer the distributions of the predicted and actual values, the more accurate the model.



The scatterplot displays the predicted values (in blue) and actual recovered VAT values (in red) against the declared tax amounts. The closer the distributions of the predicted and actual values, the more accurate the model. The clustering of points indicates the relationship between the predicted and actual values, showing how well the model predicts VAT recovery. A strong alignment of the blue and red points along a diagonal line would indicate good predictive performance.

Exercise 4 – Applying the model to the full dataset

In this exercise, you will try to implement the Fourth Stage of the Toolkit Part 2. The tested model is applied to the full dataset, and a predicted tax assessed is created.

- a) Repeat the steps in Exercise 2. Open the dataset, create a dummy variable "reg_audit" for all audits greater than zero. Create a new variable "evaded" from the "vat_evasion" variable.
- b) Create an indicator variable that flags records where the evaded variable is not missing and reg_audit is equal to 1 (indicating audited regions). Create a new variable, mimic_tax, as a copy of the evaded variable, and replace the missing values in evaded with zero in mimic_tax.
- c) Run the random forest model with the parameters identified in Exercise 2 on the full sample.
- d) Predict the amount of undeclared tax (assessment) for unaudited records. Replace the assessment variable with the evaded amount for the audited records. The prediction is done! Save the dataset with the predictions to calculate the tax gap.

Exercise 4 – Solution

- a) Repeat the steps in Exercise 2. Open the dataset, create a dummy variable "reg_audit" for all audits greater than zero. Create a new variable "evaded" from the "vat_evasion" variable.

```
use "$data\vat_audit_new.dta", clear // Insert the database to use
generate reg_audit = (audit>0)
generate evaded = vat_evasion
```

- b) Create an indicator variable that flags records where the evaded variable is not missing and reg_audit is equal to 1 (indicating audited regions). Create a new variable, mimic_tax, as a copy of the evaded variable, and replace the missing values in evaded with zero in mimic_tax.

```
generate indicator = (evaded !=. & reg_audit==1)
generate mimic_tax = evaded
replace mimic_tax= 0 if evaded==.
```

Exercise 4 – Solution

- c) Run the random forest model with the parameters identified in Exercise 2 on the full sample.

```
rforest mimic_tax rtn_year city region isic sector tax_region tax_sales vat_tax_sales  
zero_rate_sales exempt_sales import_sales vat_import_sales export_sales capital_sales  
nondeduc_sales taxunregistered_sales vat_taxunregistered_sales total_sales  
total_vat_sales tax_purchases vat_tax_purchases zero_rate_purchases exempt_purchases  
import_purchases vat_import_purchases export_purchases capital_purchases  
nondeduc_purchases taxunregistered_purchases vat_taxunregistered_purchases  
total_purchases total_vat_purchases vat_declared if indicator==1, type(reg) iter(400)  
numvars(18)  
ereturn list
```

- d) Predict the amount of undeclared tax (assessment) for unaudited records. Replace the assessment variable with the evaded amount for the audited records. The prediction is done! Save the dataset with the predictions to calculate the tax gap.

```
predict vat_assessed if indicator==0  
replace vat_assessed = evaded if indicator==1 & reg_audit==1  
sum vat_assessed, d  
browse  
save "$data\vat_audit_final.dta", replace
```

Exercise 5 – Estimating the Tax Gap

This is a continuation of the Fourth Stage of the Toolkit Part 2. The VAT gap is calculated.

- a) Open the dataset. Collapse the assessed (`vat_assessed`) and VAT declaration (`vat_declared`) variables using the return year (`rtn_year`) variable to get year level estimates. Browse the dataset.
- b) Generate the potential VAT (`pot_tax`) by summing the assessed amounts (`vat_assessed`) and VAT declared (`vat_declared`) variables. Generate the tax gap variable. The tax gap formula is $\text{taxgap} = \text{assessed} / \text{potential tax}$ (multiply by 100 for a percentage).
- c) Plot the VAT gap by the return year (`rtn_year`) and save the graph. What can you say about the VAT gap?
- d) You are asked for the monetary value of the tax gap. Plot the tax potential in monetary terms.

Exercise 5 – Solutions

- a) Open the dataset. Collapse the assessed (vat_assessed) and VAT declaration (vat_declared) variables using the return year (rtn_year) variable to get year level estimates. Browse the dataset.

```
use "$data\vat_audit_final.dta", clear
collapse (sum) vat_assessed vat_declared, by(rtn_year)
browse
```

- b) Generate the potential VAT (pot_tax) by summing the assessed amounts (vat_assessed) and VAT declared (vat_declared) variables. Generate the tax gap variable. The tax gap formula is $\text{taxgap} = \text{assessed} / \text{potential tax}$ (multiply by 100 for a percentage).

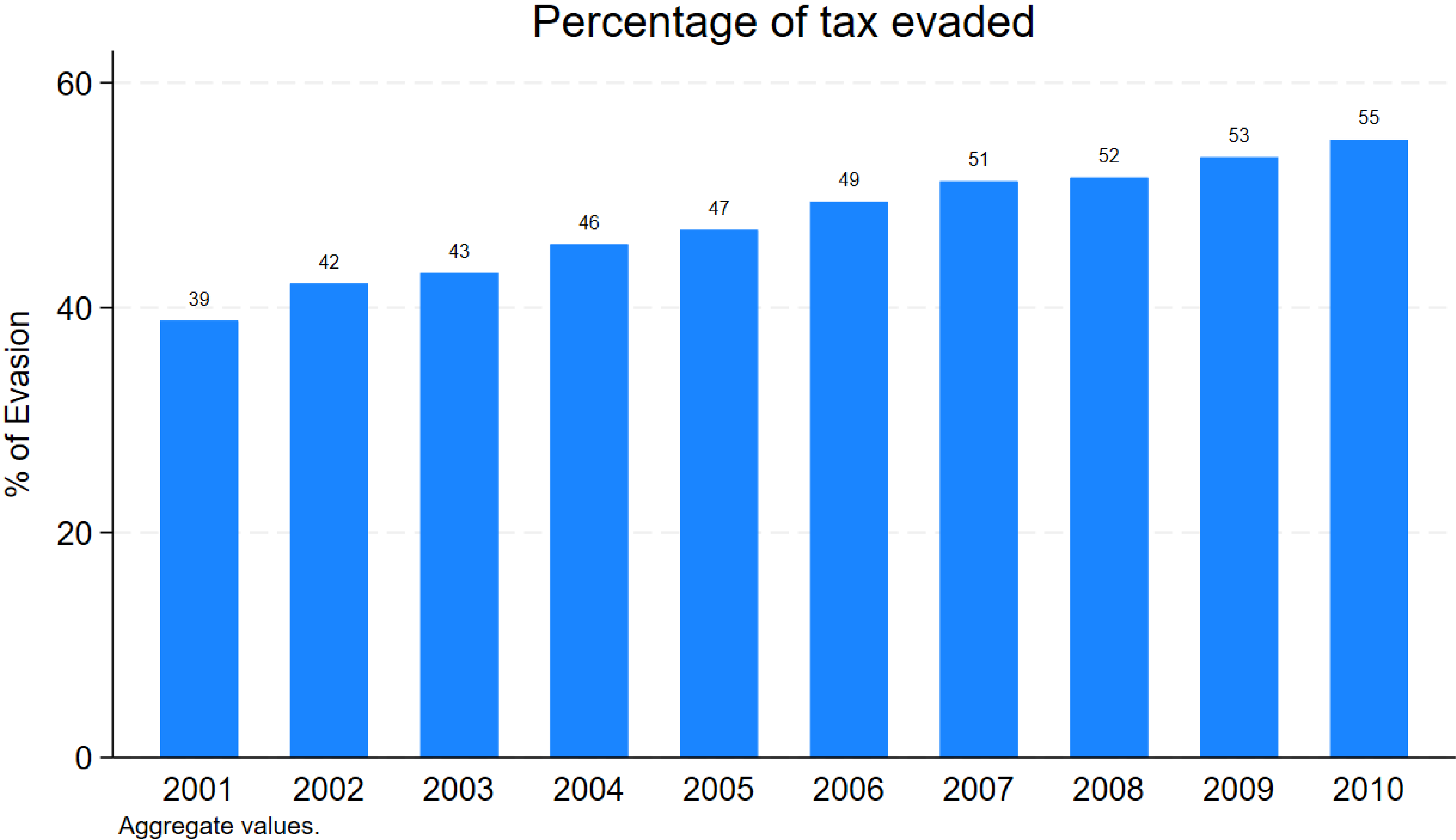
```
generate pot_tax = vat_assessed + abs(vat_declared)
generate vat_gap_years = (vat_assessed / pot_tax) * 100
browse
```

Exercise 5 – Estimating the Tax Gap

- c) Plot the VAT gap by the return year (rtn_year) and save the graph. What can you say about the VAT gap?

```
graph bar (mean) vat_gap_years, over(rtn_year) blabel(bar,  
size(vsmall) format(%20.0fc)) legend( label(1 "TAX") ) ytitle(%  
of Evasion) title("Percentage of tax evaded") note("Aggregate  
values.")  
graph export "$results\vat_gap_year.png", replace
```

Exercise 5 – Estimating the Tax Gap

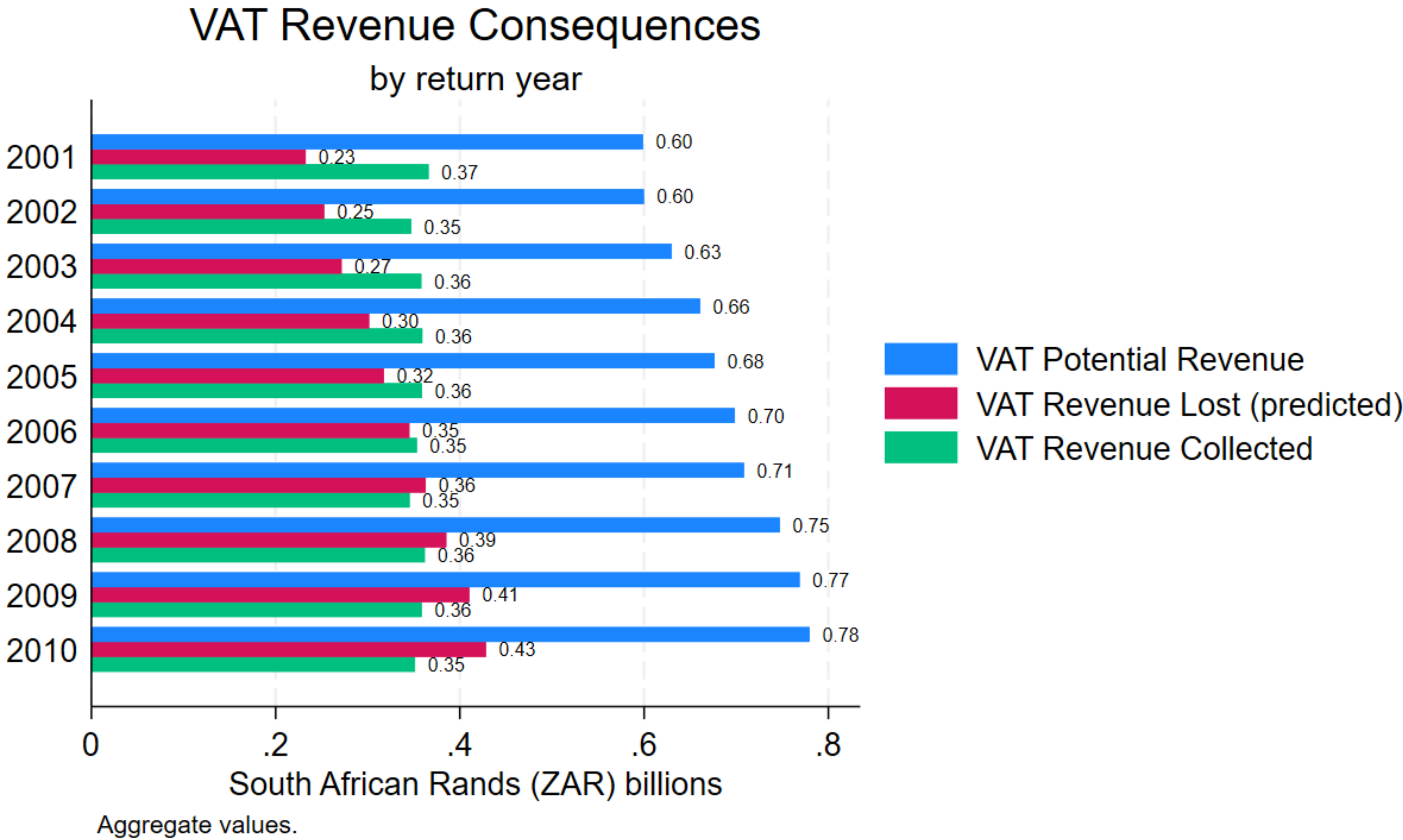


Exercise 5 – Estimating the Tax Gap

d) You are asked for the monetary value of the tax gap. Plot the tax potential in monetary terms

```
generate pot_tax_bil = pot_tax/10000000000
generate vat_ass_bil = vat_assessed/10000000000
generate vat_dec_bil = vat_declared/10000000000
graph hbar (mean) pot_tax_bil vat_ass_bil vat_dec_bil, over(rtn_year)
blabel(bar, size(vsmall) format(%20.2fc)) legend( label(1 "VAT Potential
Revenue") label(2 "VAT Revenue Lost (predicted)") label(3 "VAT Revenue
Collected")) ytitle("South African Rands (ZAR) billions") title("VAT
Revenue Consequences") subtitle("by return year") note("Aggregate
values.")
graph export "$results/vatrevenue_gap_year.png", replace
```

Exercise 5 – Estimating the Tax Gap



Bonus Exercise – Investigating compliance challenges

The advantage of the bottom-up approach is disaggregated analysis to identify compliance challenges.

- a) Calculate the tax gap by region in 2005 and graph the VAT gap by region. Hint: follow the same steps as Exercise 5 but specify the year and region.
- b) Calculate the tax gap by sector in 2010 and graph the VAT gap by sector. Hint: follow the same steps as Exercise 5 but specify the year and sector.

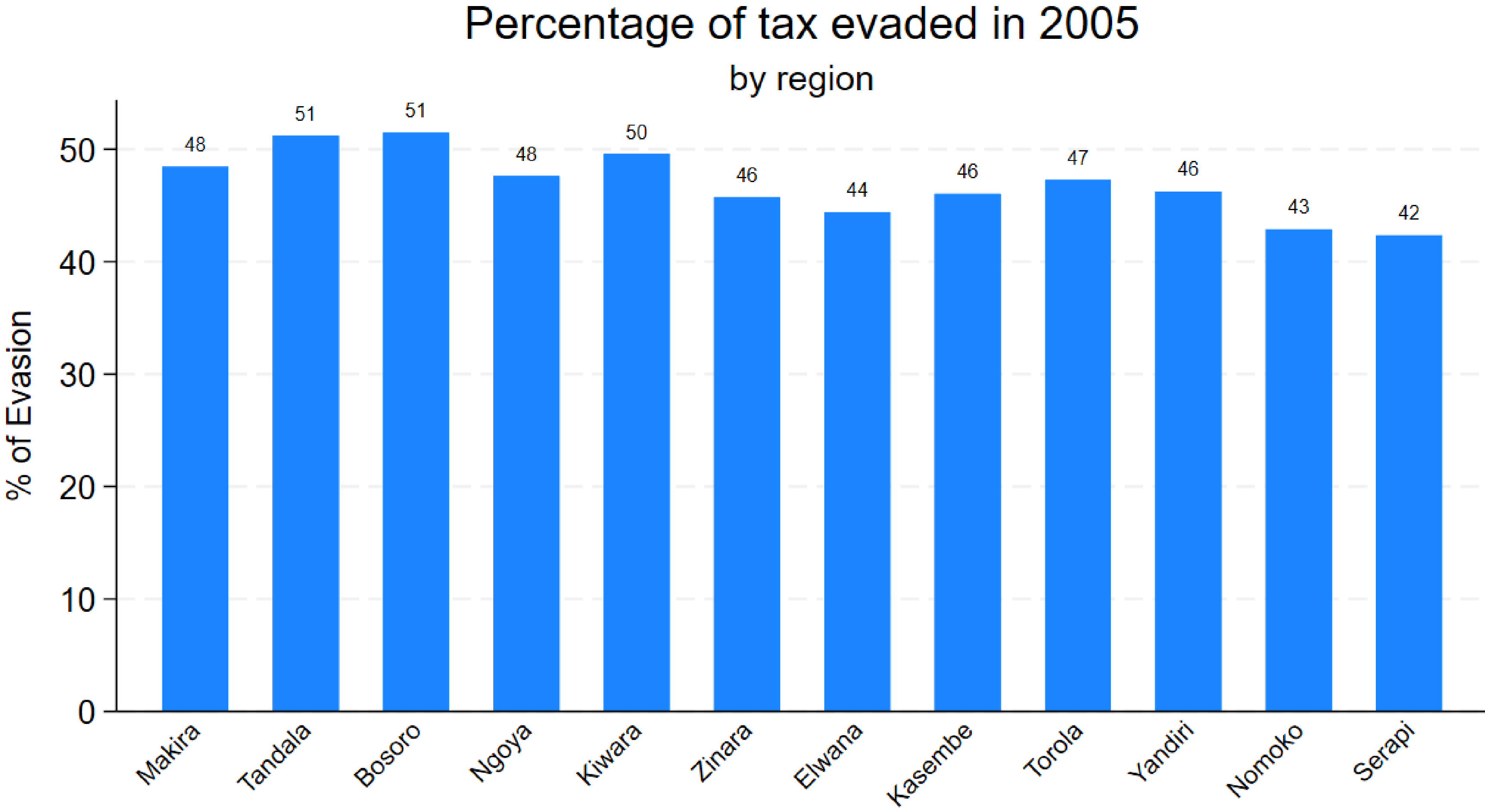
Further disaggregation is possible to answer compliance questions in your own context. For example, firm sizes, sub sectors, etc.

Bonus Exercise – Solutions

- a) Calculate the tax gap by region in 2005 and graph the VAT gap by region.
Hint: follow the same steps as Exercise 5 but specify the year and region.

```
use "$data\vat_audit_final.dta", clearcollapse (sum) vat_assessed  
vat_declared if rtn_year==2005, by(region) generate pot_tax =  
vat_assessed+abs(vat_declared)generate vat_gap_region =  
(vat_assessed/pot_tax)*100graph bar (mean) vat_gap_region, over(region,  
label(angle(45) labsize(small))) blabel(bar, size(vsmall)  
format(%20.0fc)) legend( label(1 "TAX") ) ytitle(% of Evasion)  
title("Percentage of tax evaded in 2005") subtitle("by region")  
note("Aggregate values.")  
graph export "$results/vat_gap_loc.png", replace
```

Bonus Exercise – Solutions



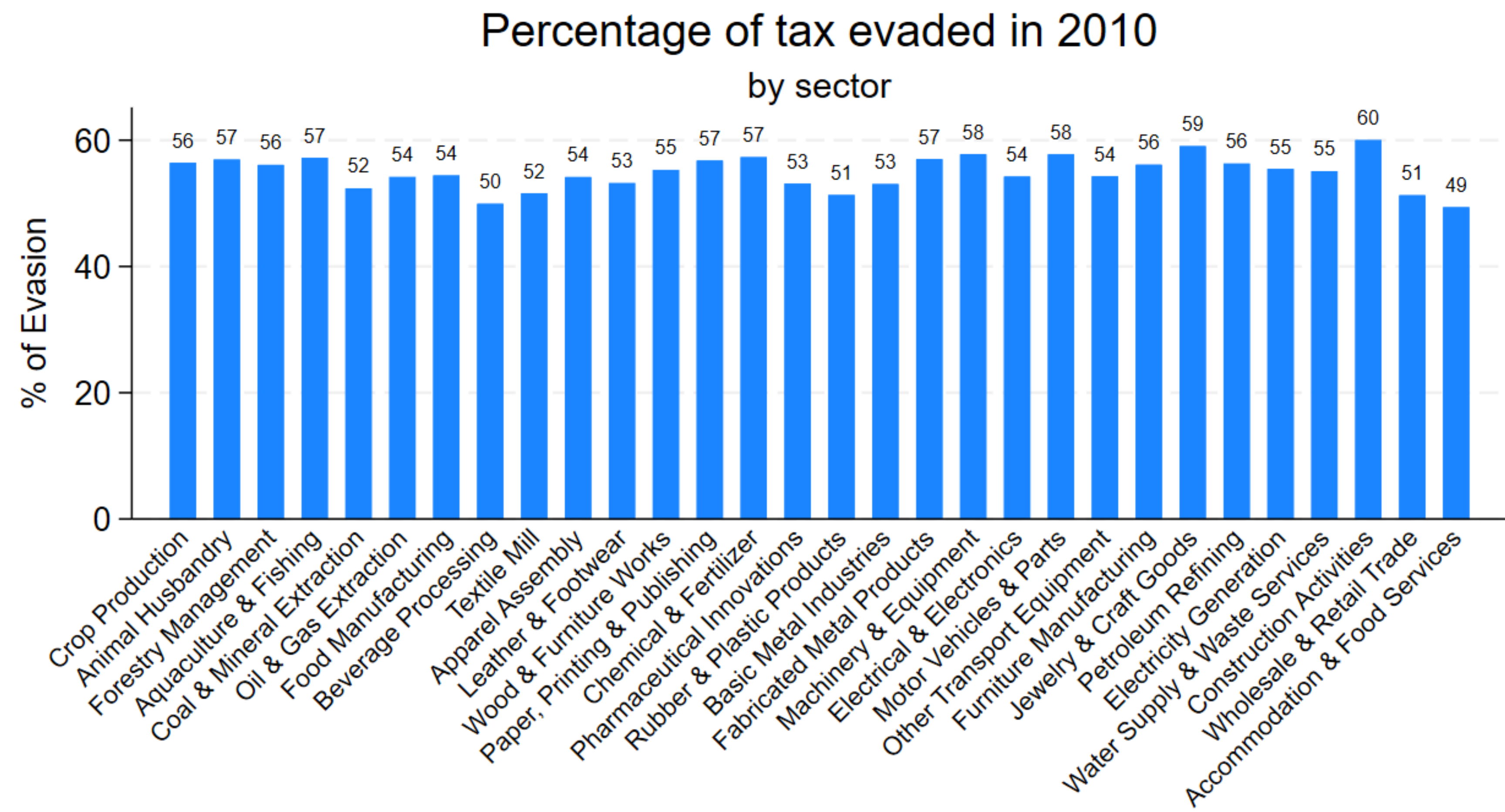
Aggregate values.

Bonus Exercise – Solutions

- b)** Calculate the tax gap by sector in 2010 and graph the VAT gap by sector.
Hint: follow the same steps as Exercise 5 but specify the year and sector.

```
use "$data\vat_audit_final.dta", clearcollapse (sum) vat_assessed  
vat_declared if rtn_year==2010, by(isic) generate pot_tax =  
vat_assessed+abs(vat_declared)generate vat_gap_sector =  
(vat_assessed/pot_tax)*100graph bar (mean) vat_gap_sector, over(isic,  
label(angle(45) labsize(small))) blabel(bar, size(vsmall)  
format(%20.0fc)) legend( label(1 "TAX") ) ytitle(% of Evasion)  
title("Percentage of tax evaded in 2010") subtitle("by sector")  
note("Aggregate values.")  
graph export "$results/vat_gap_sec.png", replace
```

Bonus Exercise – Solutions



Aggregate values.

Test your knowledge at home: Estimate CIT & PAYE Gaps

Repeat exercises using CIT and PAYE datasets and variable list, and parameters below:

CIT variables: rtn_year fixed_assets depreciation investment loans provisions dividends reserves debts sales_turnover wages total_purchases profits operating_cost taxes_paid donations total_profit taxable_profits cit_payment city region isic sector tax_region

CIT parameters: 400 iterations (insert figure in syntax “iter(400)”) and 20 variable replacements (insert figure in syntax “numvar(20)”)

PAYE variables: rtn_year number_employers total_gross_payment total_nontax_payment total_tax_deductions total_payroll_tax total_chargeable_payments paye_tax paye_payment city region isic sector.

PAYE parameters: 400 iterations (insert figure in syntax “iter(400)”) and 6 variable replacements (insert figure in syntax “numvar(6)”)

Thank you!



**UNU
WIDER**

More information available online

wider.unu.edu

Follow us on LinkedIn Facebook and Twitter **@UNUWIDER**