ML-Based Imputation Methods in R Package VIM

Performance and Considerations

Alexander Kowarik
Alexander.Kowarik@statistik.gv.at

Johannes Gussenbauer
Johannes.Gussenbauer@statistik.gv.at

Nina Niederhametner
Nina.Niederhametner@statistik.gv.at

useR! 2024 Salzburg, 04.04.2024

www.statistik.at

Outline

- Short introduction to R-package VIM
- Recent methodological additions to the package
- Simulation study
- Outlook

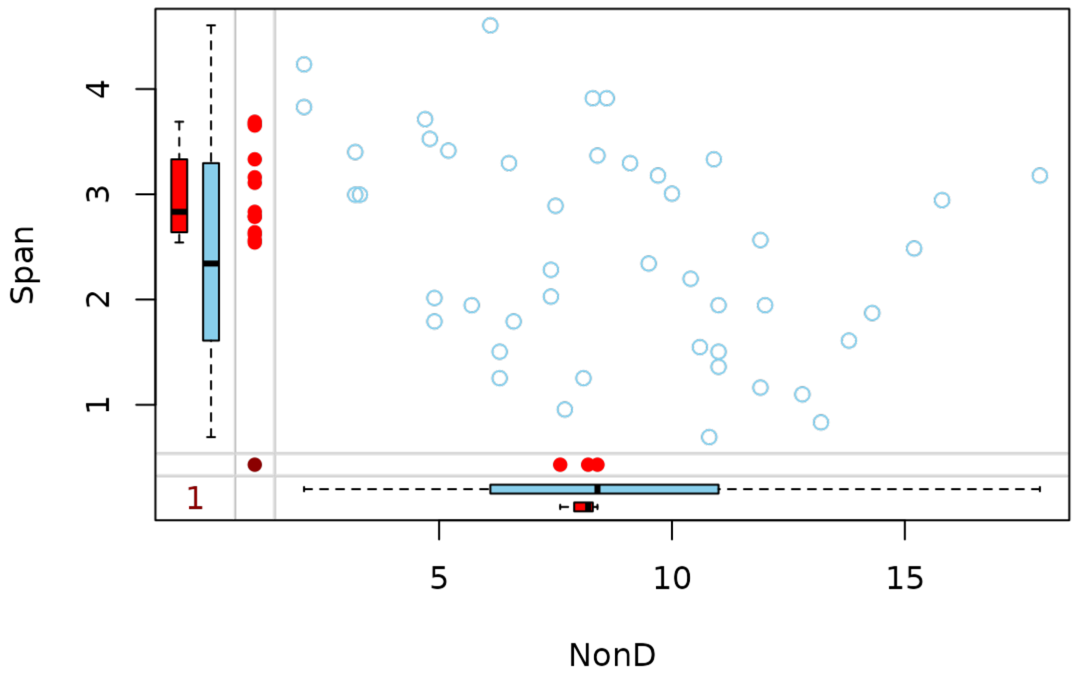
R-Package VIM

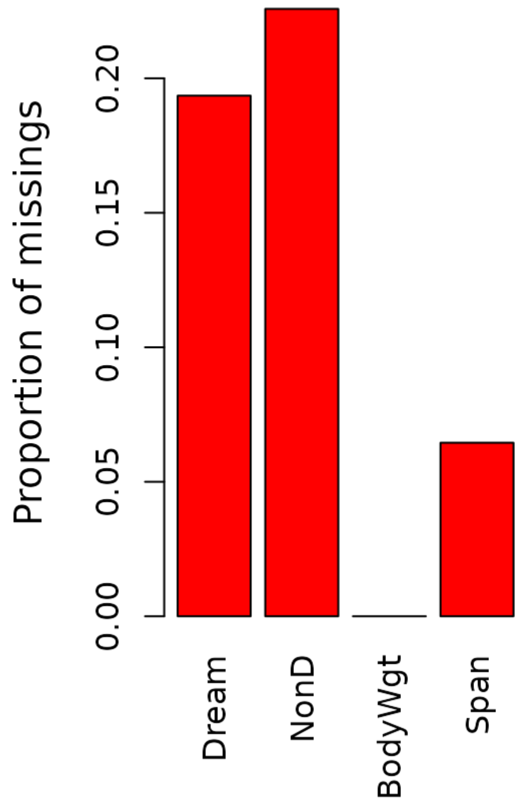
R-Package VIM

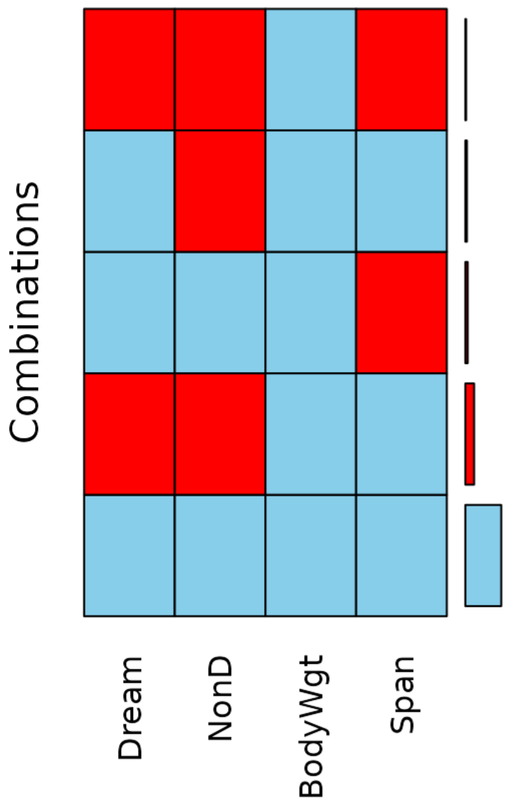
- VIM: **V**isualization and **I**mputation of **M**issing Value
- Developed for the use of tabular data
  - Records ~ rows
  - Variables ~ columns
- Contains various imputation methods
- Available on CRAN and actively developed
  - https://cran.r-project.org/web/packages/VIM/index.html
  - https://github.com/statistikat/VIM

$$\mathbf{X} = \begin{pmatrix} x_{11} & \dots & \dots & x_{1p} \\ \vdots & \text{NA} & & \vdots \\ & & & \text{NA} \\ \vdots & & \text{NA} & \vdots \\ x_{n1} & \dots & \dots & x_{np} \end{pmatrix}$$

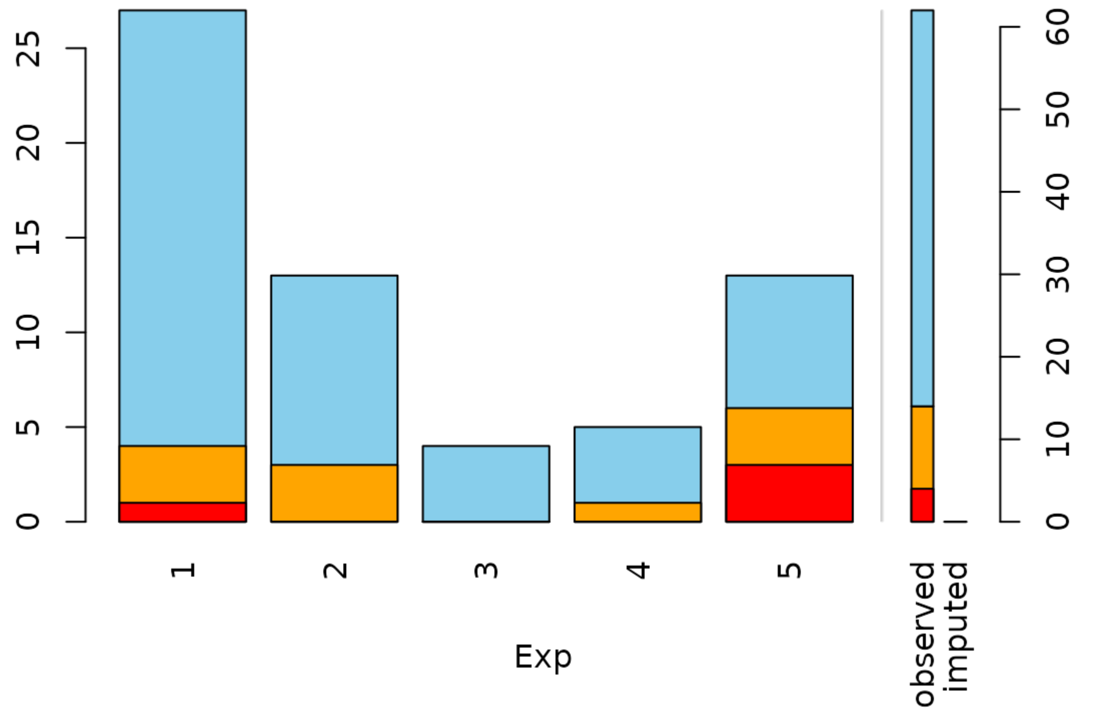
Visualization of missing values

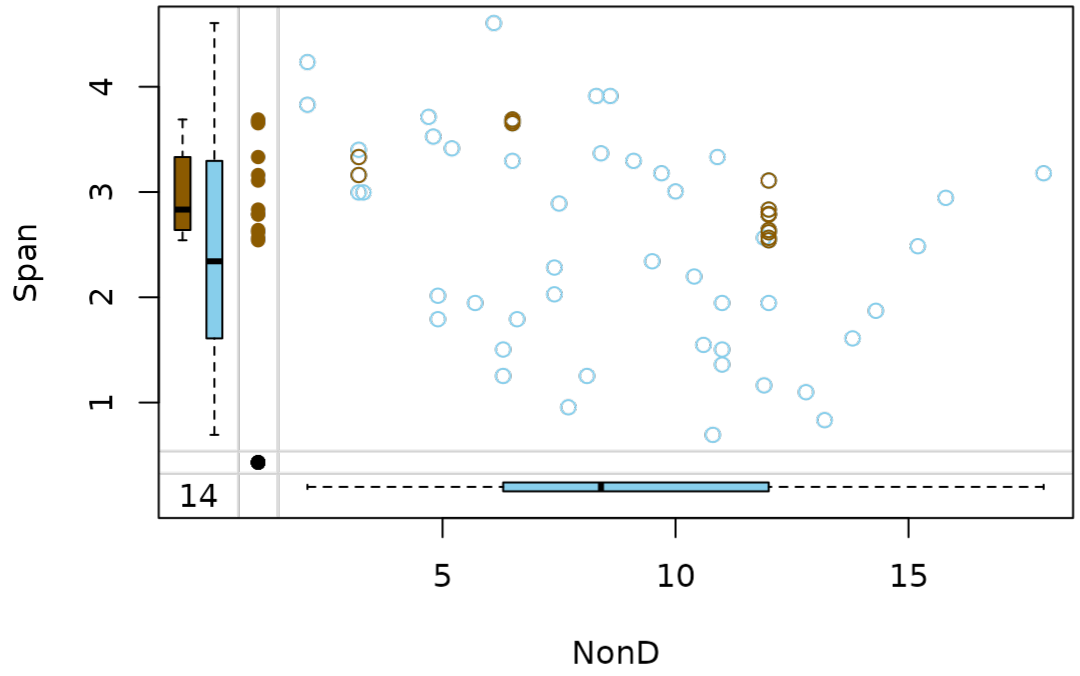






Visualization of imputed values





Imputation methods available

- Donor based method

```r
library(VIM)
data(sleep) # example data from package


kNN(sleep, variable = ..., k = 5, dist_var = ..., ...)

hotdeck(sleep, variable = ..., ord_var = ..., domain_var = ..., ...)

matchImpute(sleep, variable = ...,  match_var = ..., ...)
```

Imputation methods available

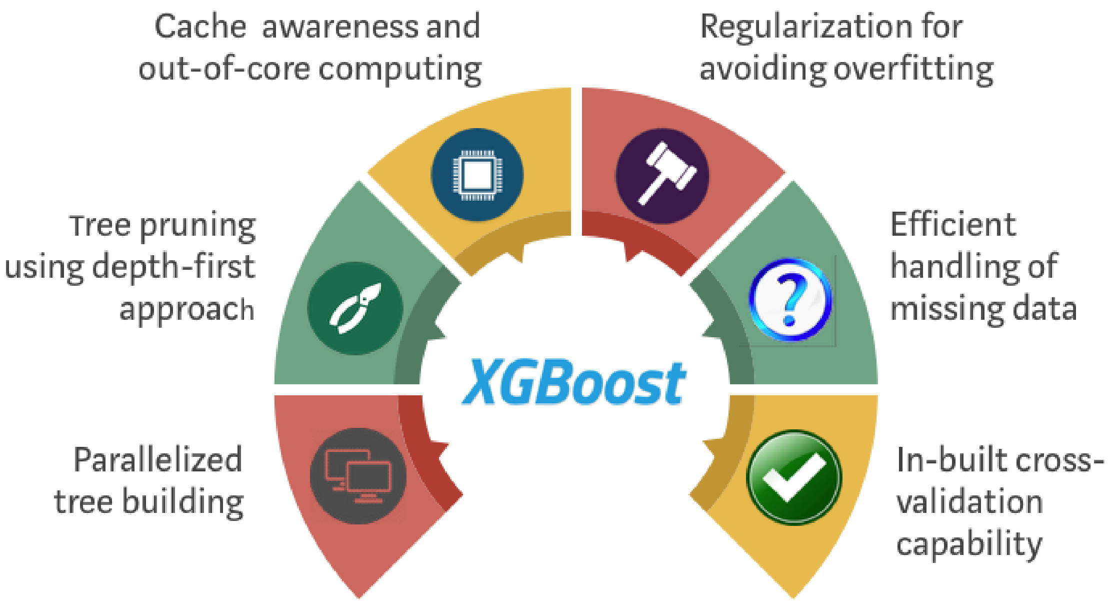
- Model based methods

```
regressionImp(formula = ..., data = sleep, family = ..., robust = ..., ...)

# Iterative robust model-based imputation
irmi(x = sleep, maxit = 100, noise = ..., robust =  ..., ...)

rangerImpute(formula = ..., data = sleep, ...)
```

- Option to add random noise
- Sample from predicted probabilities for categorical variables

Impute with XGBoost (Chen and Guestrin 2016)

- Gradient Tree boosting
- Available for R and Python
- Parallelisation
- **Strong out of the box method**



Impute with transformers

Impute missing values with Large Language Models:

- → Convert tabular data to text
- → Train a Transformer Model with the text inputs of observations that do not contain missing values for the target variable
- → Transformer generates the missing values based on the given variable values

Age	Country	Salary
30	Austria	40,300.03
54	Germany	107,000.40
24	Spain	38,000.55
40	Austria	**NA**

`„30, Austria, 40300.03“`
`„54, Germany, 107000.40“`
`„24, Spain, 38000.55“`

Tokenizer

Transformer

`„40, Austria,` **`68000`**`“`

Age	Country	Salary
30	Austria	40,300.03
54	Germany	107,000.40
24	Spain	38,000.55
40	Austria	**68,000**

Text pre-processing & Tokenization

- **Categorical variables:** one token per category
- **Numeric variables:** one token per digit, additionally „-” and „.“ if applicable

Age	Country	Salary
30	Austria	40,300.03
54	Germany	107,000.40
24	Spain	38,000.55

„30“ „Austria“ „040300.03“
„54“ „Germany“ „107000.40“
„24“ „Spain“ „038005.55“

„30,Austria,040300.03“
„54,Germany,107000.40“
„24,Spain,038005.55“

Tokenizer		
Token	**Token ID**	**Column**
Austria	1	Country
Germany	2	Country
Spain	3	Country
0	4	Age
1	**5**	**Age**
…	…	…
9	13	Age
1	**14**	**Salary**
2	15	Salary
…	…	…

Identical tokens from different columns are assigned different Token IDs

XGBoost and transformer in VIM

```
xgboostImpute(formula = ..., data = ..., ...)
```

```
transformerImpute(data = sleep, target = ..., cat_vars = NULL, ...)
```

- xgboostImpute() already available for latest CRAN release
- transformerImpute() not yet fully implemented (based on R packages keras, transformer)

Simulation Study

Simulation Study

- Aim: Test multiple methods against each other, including **xgboost** and **transformer**

- Data: Richframe ~ housing register containing all registered persons in Austria living in private households
  - Variety of variable: Geographic variables, variables on household structure, sociodemographic variables, …

HID	NUTS2	age	sex	Citizenship	Education	Yearly Income
1	AT13	30	m	AT	Post-Secondary	25000
2	AT32	56	m	EU	Secondary	28000
2	AT32	52	f	AT	Tertiary	32000
2	AT32	18	f	AT	Secondary	0

Simulation Study

- Take sample from Richframe ➔ add missing values for a specific variable ➔ apply imputation method ➔ compare results

- Apply different missing mechanisms
  - MCAR: randomly draw position of missing values
  - MAR: Occurence of missing value depends on other observed variables
  - MNAR: Occurence of missing values depends on the variables itself

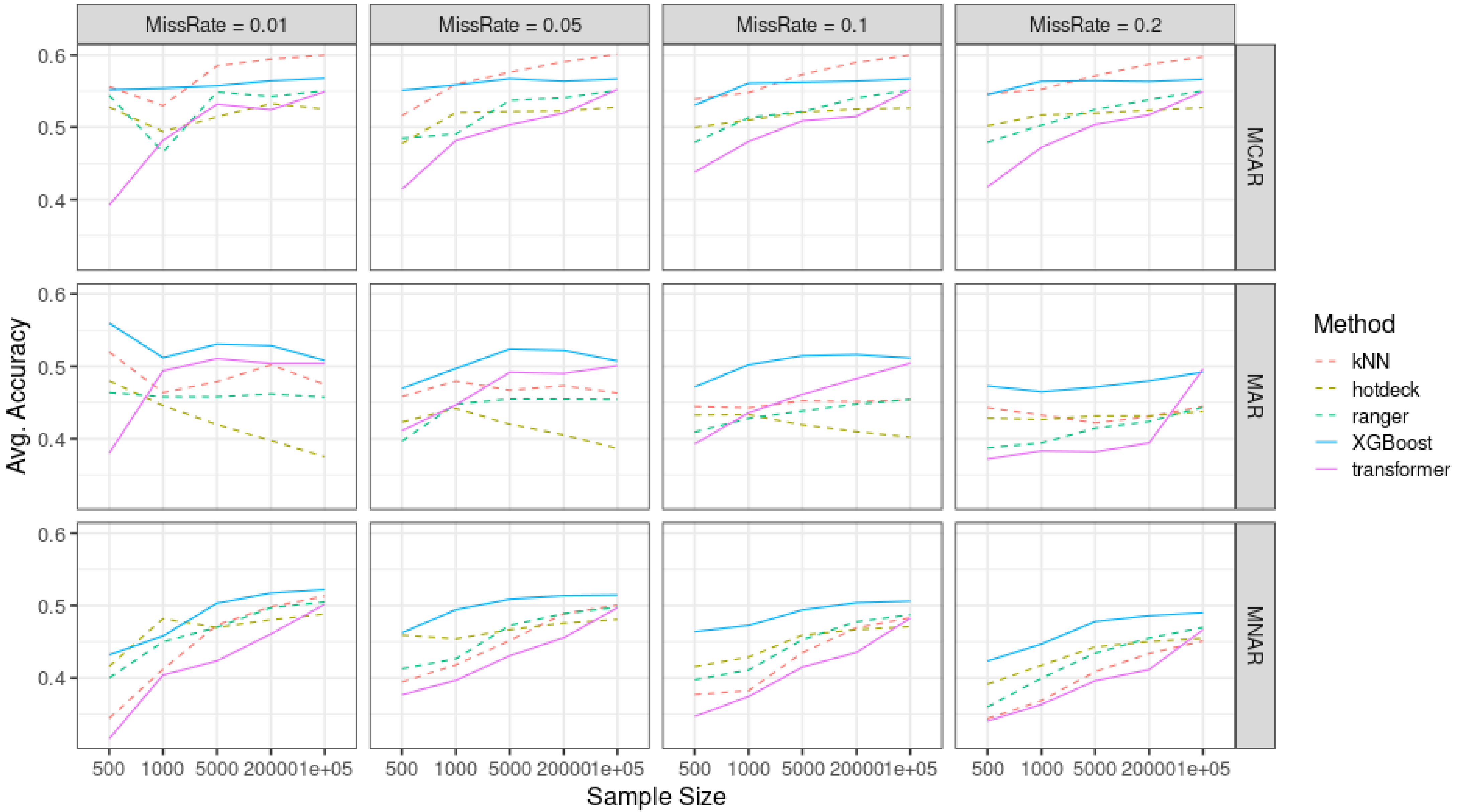
- Simulate MAR or MNAR we derived occurence of missing value from typical non response patterns
  - Higher response rates: rural areas, higher education, higher yearly income
  - Lower response rates: urban areas, lower education, migration background, low or very high income

Simulation Study
Parameter Setup

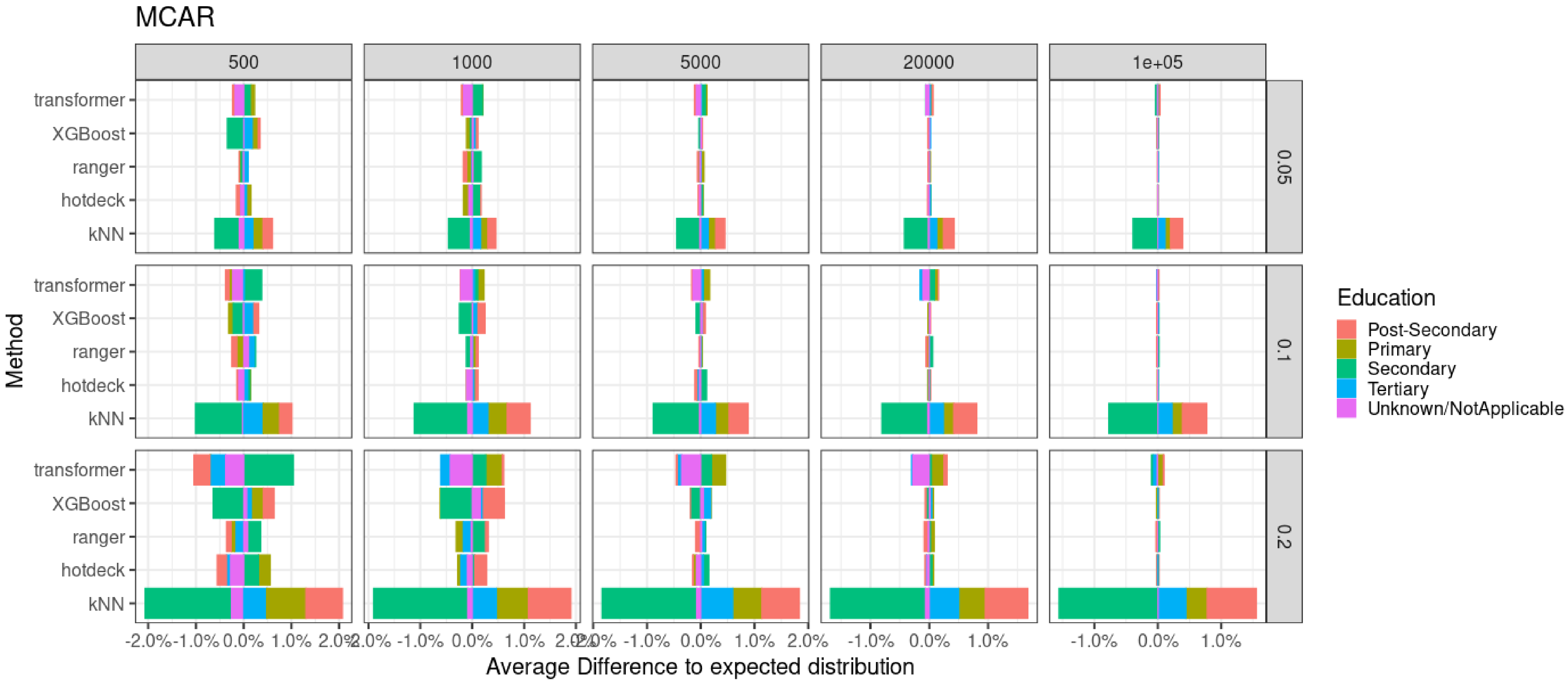
- Test methods
  kNN(), hotdeck(), rangerImpute(), xgboostImpute(), transformerImputer()
- Sample n
  500, 1000, 5000 , 20000 , 100000
- Missing rate r
  0.01, 0.05, 0.1, 0.2
- Variables to impute
  Education, Citizenship, Yearly Income
- Missing mechanism
  MCAR, MNAR, MAR

→ Repeat many times

Results - Education

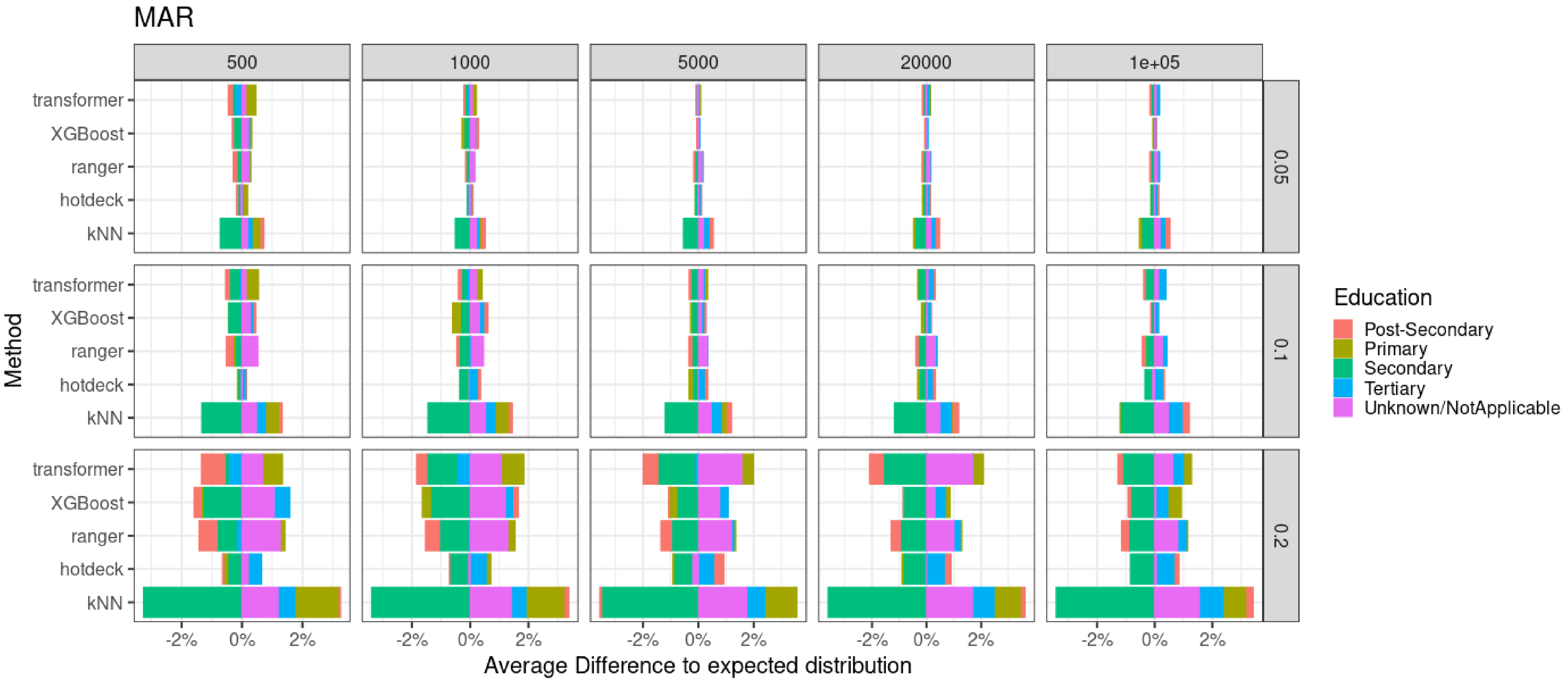


Results - Education
Difference in distribution of classes

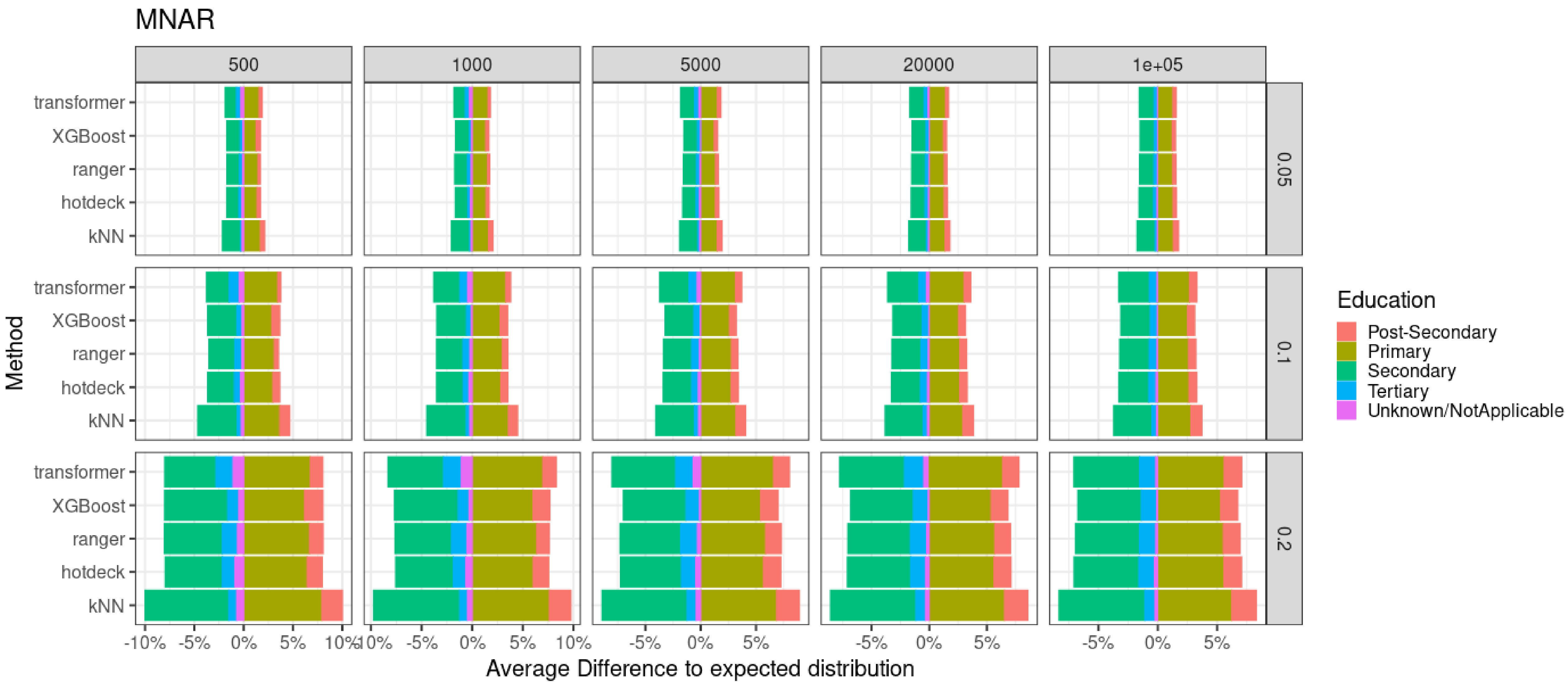


Results - Education

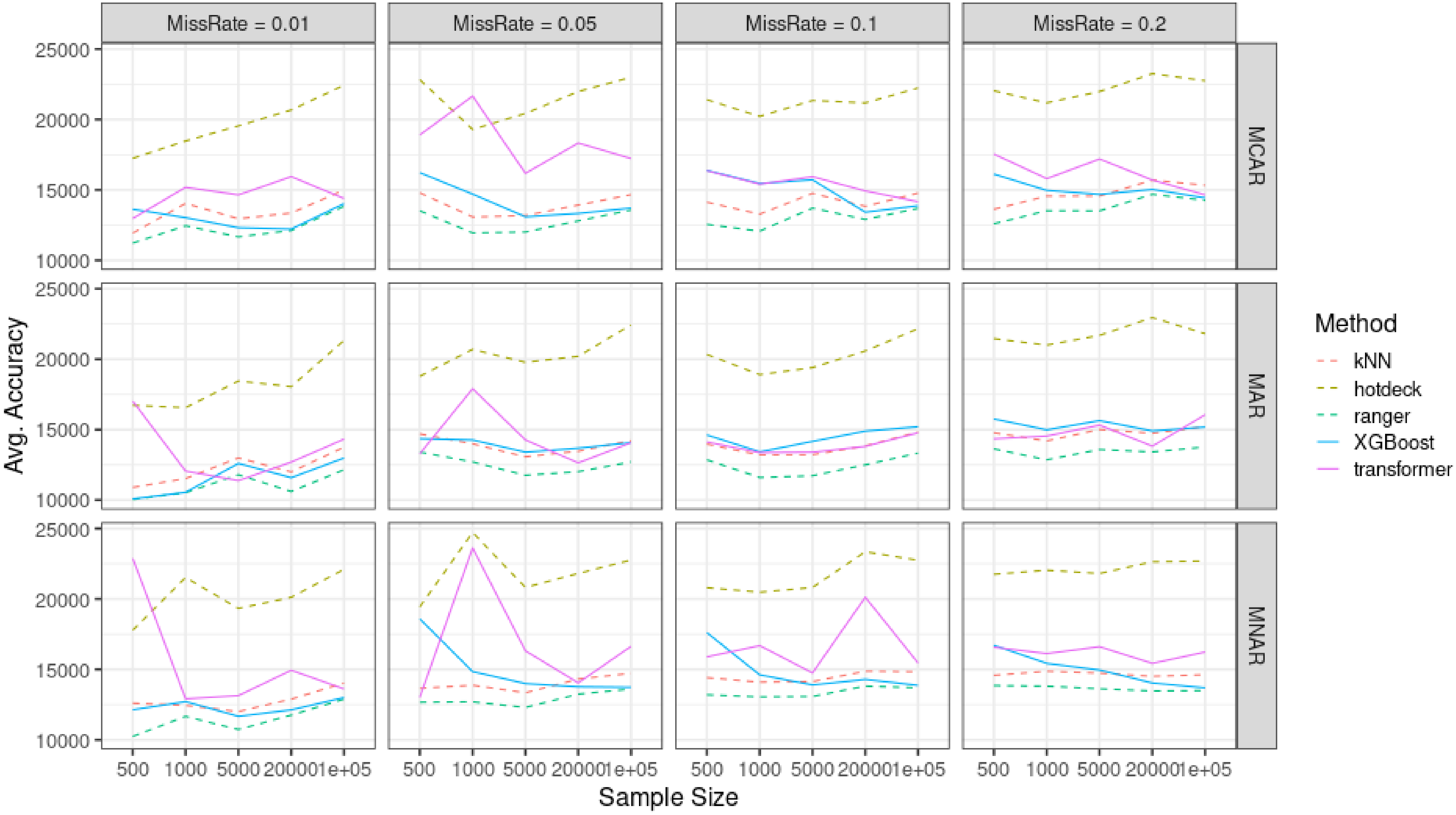
Difference in distribution of classes



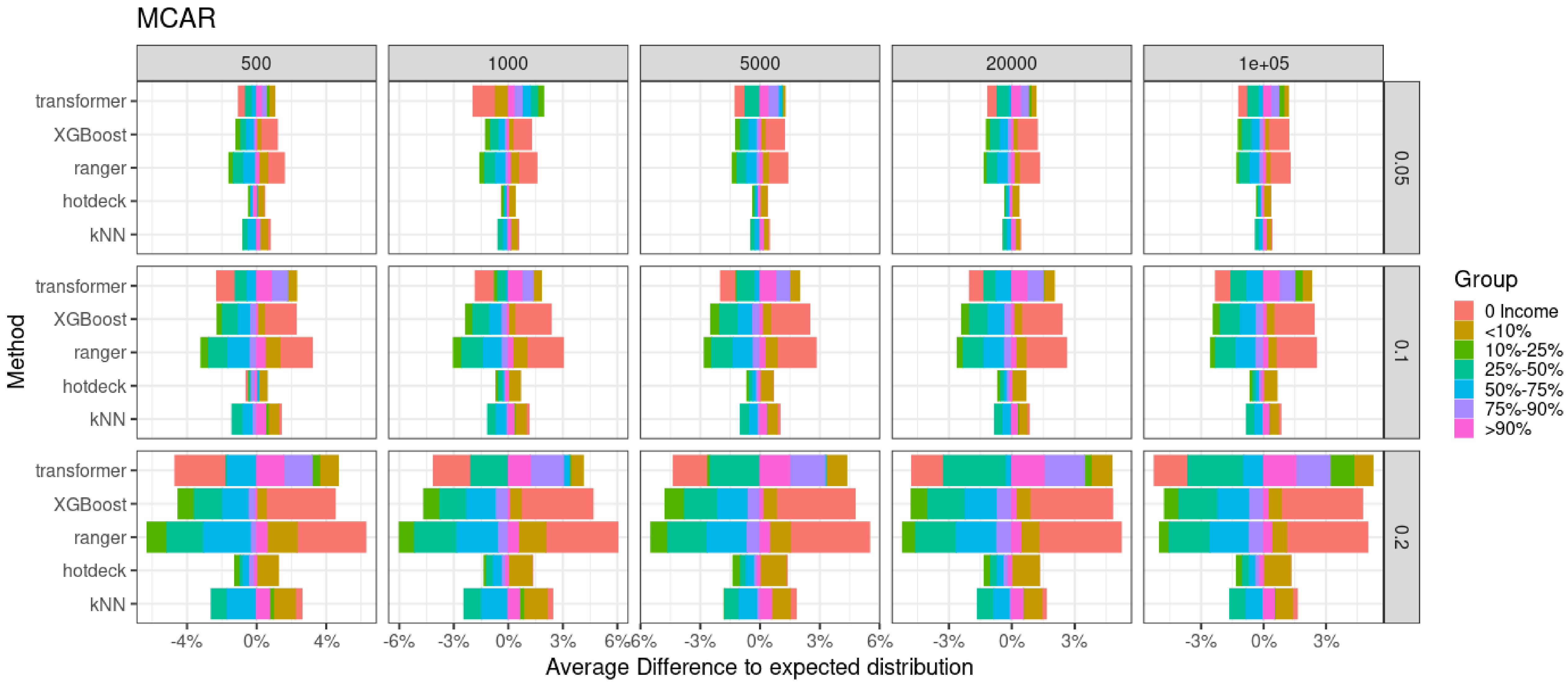
Results - Education
Difference in distribution of classes



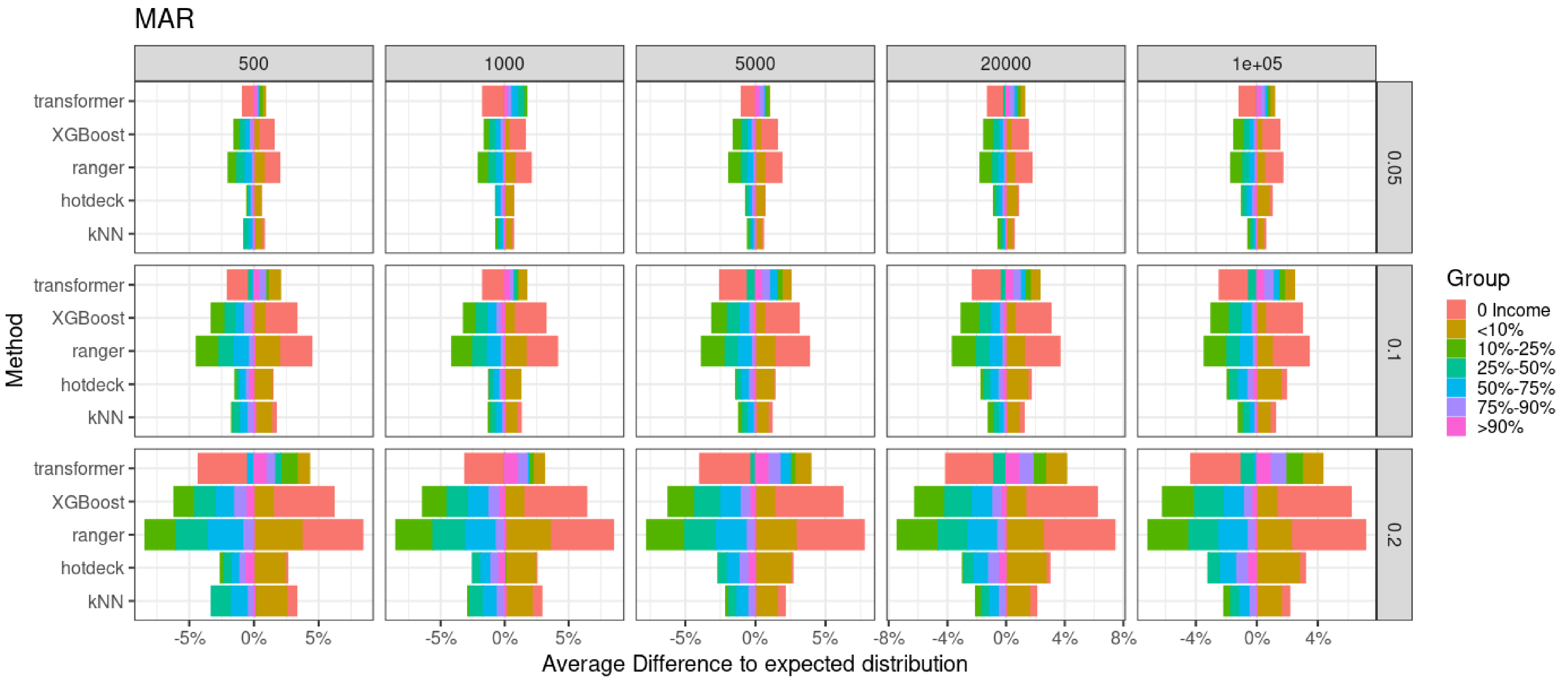
Results - Yearly Income



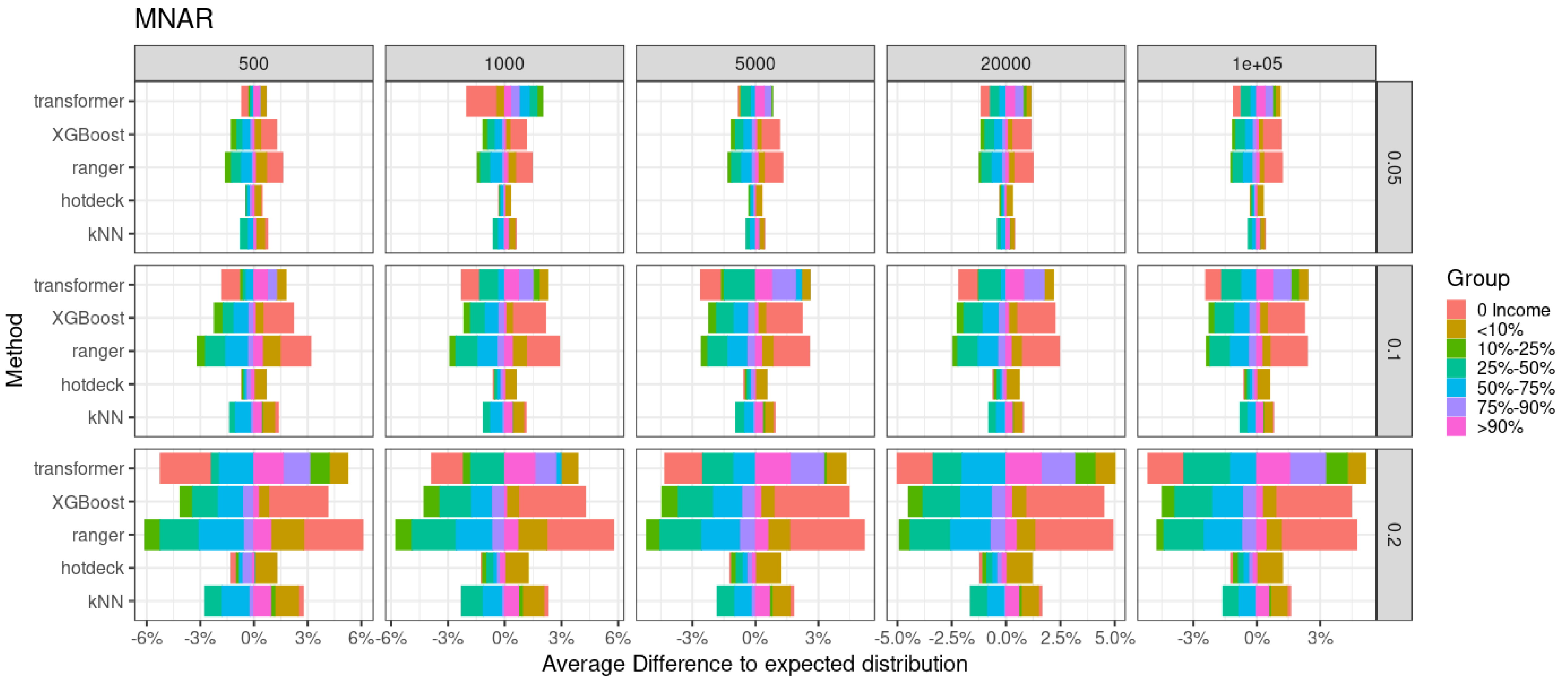
Results - Yearly Income



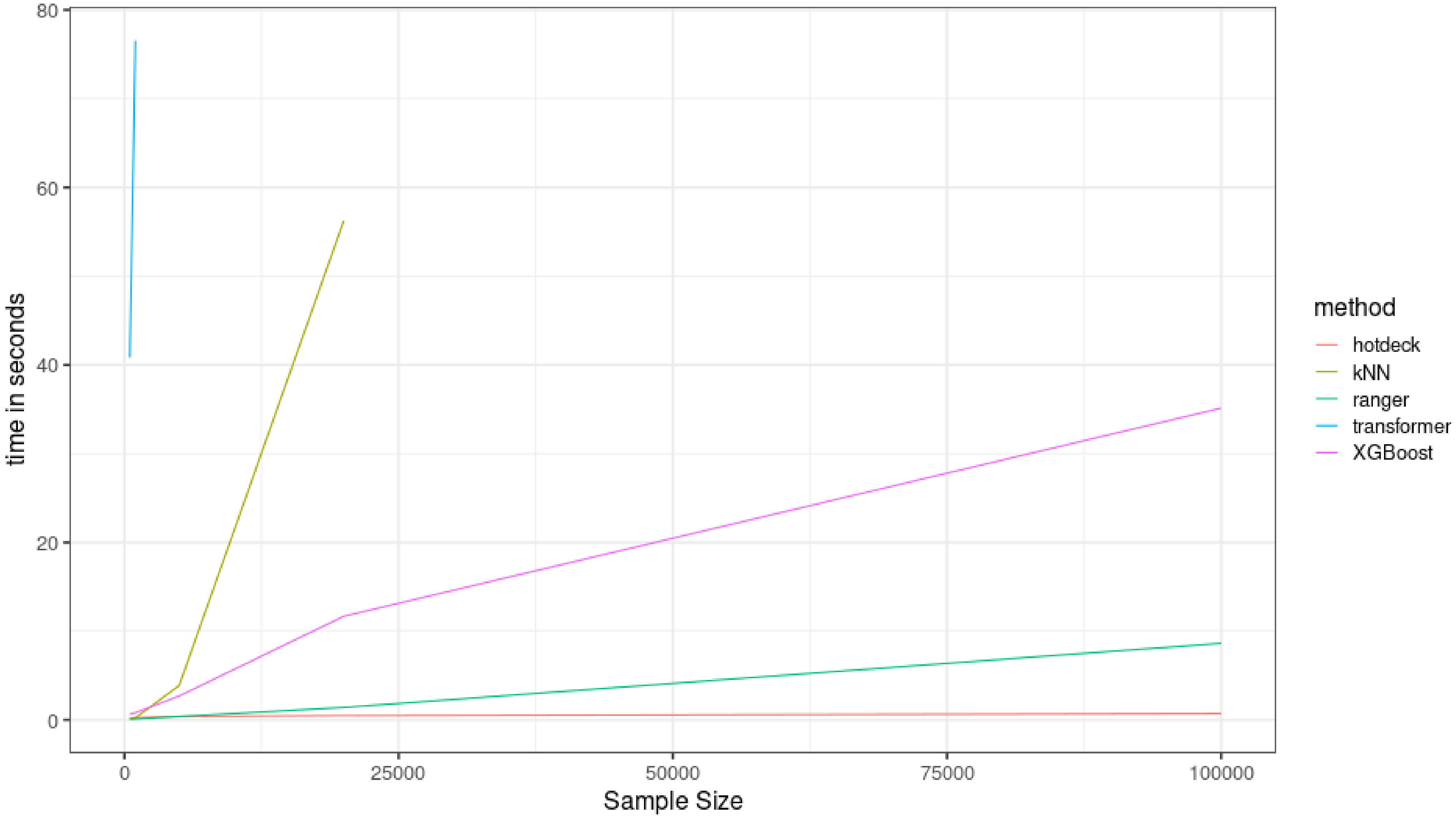
Results - Yearly Income



Results - Yearly Income



Results - Runtime



Conclusions and outlook

- Work in progress and potential to improve
- Trade of between accuracy and biased estimates
- Plan to further develop VIM package
  - harmonise model based imputation methods (and imputation interface in general)
  - Include predictive mean matching
  - Use of pretrained (BERT?) models as starting point for transformerImpute

Rückfragen bitte an
Johannes Gussenbauer
Johannes.Gussenbauer@statistik.gv.at

Any Questions?

STATISTIK AUSTRIA
Guglgasse 13, 1110 Wien