NHH

HOME EXAMINATION BAN400

Fall, 2023

Start: December 18. 2023 09:00

End: December 20. 2023 14:00

THE HOME EXAMINATION SHOULD BE SUBMITTED IN WISEFLOW

You can find information on how to submit your paper here:

`https://www.nhh.no/en/for-students/examinations/home-exams-and-assignments/`

Your candidate number will be announced on StudentWeb. The candidate number should be noted on all pages (not your name or student number). In case of group examinations, the candidate numbers of all group members should be noted.

SUPPLEMENTARY REGULATIONS FOR HOME EXAMINATIONS

You can find supplementary regulations under the headline "Regulations":

`https://www.nhh.no/en/for-students/regulations/`

Find more information under chapter 4.0 in the Supplementary provisions to the regulations for fulltime study programmes

Number of pages, including front page: 6

Number of attachments: 2 (BAN400-H23-template.Rmd, BACEDATA.csv)

About the home exam

In order to complete BAN400, you must pass this individual home exam. Please take note of the following items:

- You should submit a single `.zip` file on WiseFlow, and the `.zip`-file should contain:
  - One `.qmd`-file with your answer to the questions.
  - One `.html`-file, compiled from the `.rmd`-file that you submit.
  - The data file, `BACEDATA.csv`.
- The `.qmd`-file should be based on the template-file `BAN400-H23-template.qmd`.
- The examiner should be able to reproduce the `.html`-file by compiling the `.qmd`-file without any modifications. State clearly if you make use of packages that must be installed before compilation.
- Note that this is an *individual* exam, and any cooperation is *not* permitted.
- The deadline for submission in WiseFlow is a hard constraint. Make sure you submit in time.
- The grader is mostly interested in your code. However, we recommend also writing a few sentences to each question, explaining your results and answers to the questions. This helps the examiner in assessing your submission.
- If you do not manage to complete some assignments, you may score partial points by submitting a partial solution, with a comment on how your solution fails or is incomplete.
- You will likely need to use functions or packages *not* explicitly covered in BAN400. Understanding documentation and independently resolving programming issues are part of the learning outcomes of the course.
- **The use of generative AI is not forbidden during this exam. Note the following:**
  - Reference all code in your answer that you have not written yourself, for example by a direct link to online sources or to the specific AI model that generated the code.
  - There is a generative AI statement included in the template `.qmd`-file. Do not edit or remove this section in your answer. By submitting the exam answer, you agree with the content of this statement.
  - You are required to include a short personal statement regarding your use of generative AI at the end of the exam. See **Assignment 7**.

Cherry-picking regressions

In the "Many Models" assignment, we worked with a data set covering 139 countries and used ten variables to explain the GDP per capita growth rate. You will analyze a similar data set in this home exam, but we have even more explanatory variables this time. The data is available in the file `BACEDATA.csv`. See Table A1 in the paper linked in the footnote[1] for descriptions of all the variables in the data set.

With 67 potential explanatory variables, we can construct in total 2^{67} different regression models by combining different explanatory variables (even more if we allow transformations and interaction terms). This number is enormous. If we could estimate 1000 models every second, it would take almost 5000 million years to estimate all of them. Thus, any researcher analyzing this data set must be selective in which models to estimate. This freedom of choice, however, could lead to cherry-picking. By "cherry-picking", we mean that the researcher only reports results that support his or her hypotheses. This process could lead to confirmation bias, where the researcher will continue exploring the data set until it yields "reasonable", "desirable", or "remarkable" results, either by conscious choice (cheating) or by ignorance of the scientific process. In either case, we should always

[1] `https://dx.doi.org/10.2139/ssrn.2901346`

carefully assess empirical research against theory and other sources to safeguard against the potential pitfalls of cherry-picking and uphold the standards of genuine scientific exploration and discovery.

In this exam we will systematically assess the risk of cherry-picking. We will do this by checking if we can support **any** conclusion that we want by simply finding the "correct" regression model.

Assignment 1

Read the data into memory. The data starts on line 3. Drop the variables `OBS`, `CODE` and `COUNTRY` in the original data set. Ensure that all variables are encoded as numbers or integers.

Assignment 2

Use `GR6096` (growth of GDP per capita between 1960 and 1996) as response variable in all linear regression models below. Furthermore, we are interested in finding regression models that confirm a particular hypothesis regarding the influence of the variable `SOCIALIST` on the outcome `GR6096`. This means that we need to find out which other explanatory variables we should include in a regression model to achieve this particular goal[2].

Select the variables `LANDAREA, PRIEXP70, EAST, AVELF, ORTH00, POP6560, REVCOUP, OTHFRAC, MINING` and `PROT00` from the data set, in addition to `GR6096` and `SOCIALIST`. This should give you a total of 1024 linear regression models you can try. Estimate all of them, and present a histogram over the estimated coefficients on `SOCIALIST`.

Assignment 3

Researchers typically care for both the estimated coefficients as well as their respective p-values, which indicate whether the estimate is "statistically significantly different from zero". Write a function called `is_variable_significant()` that takes the following arguments:

- `covariates` - a character vector with **names** of variables that should be included in the regression model.
- `response` - the name of the response variable (the left hand side of the regression model).
- `variable_to_assess` - the name of the variable that we want to assess.
- `data` - the data set to use. Use your complete data set as default.

The function should estimate the linear regression model with `response` as response variable, and `variable_to_assess` as well as all the variables in `covariates` as explanatory variables.

The function should return a data frame (a tibble) with one row and the following columns:

- `model_call` - The complete regression equation used (as a string).
- `assessed_variable` - The name of the variable that was assessed.
- `coefficient_estimate` - Coefficient estimate of the assessed variable.
- `p_value` - p-value of the coefficient estimate of the assessed variable.

Ensure that the function passes all tests below (which are included in the template).

[2]That is, we are looking to carry out a classical scientific fraud.

```r
library(assertthat)

tmp <-
  is_variable_significant(
    response = "GR6096",
    variable_to_assess = "SOCIALIST",
    covariates = c("COLONY", "CONFUC")
  )

assert_that(tmp$model_call == "GR6096 ~ COLONY + CONFUC + SOCIALIST",
            msg = "Regression specification is not correct")
assert_that(tmp$assessed_variable == "SOCIALIST",
            msg = "Assess variable is not correct")
assert_that(round(tmp$coefficient_estimate, 8) == -0.01305317,
            msg = "The coefficient estimate of SOCIALIST is not correct")
assert_that(round(tmp$p_value, 9) == 0.004308533,
            msg = "The P-value of SOCIALIST is not correct")

rm(tmp)
```

Assignment 4

Use the function `is_variable_significant()` to re-estimate all models from Assignment 2, and show a list of all regression models with both a positive coefficient estimate for the variable `SOCIALIST` *and* where the p-value for this coefficient estimate is lower than 0.05. Use the library `tictoc` to measure the time it takes to run the code.

Assignment 5

The final assignment of this home exam will require calling the `is_variable_significant()`-function many times. It might therefore be worthwhile to speed up the function.

There is an interesting discussion on Stackoverflow[3] on how to speed up the `lm()`-function, where Dirk Eddelbuettel suggests using the `fastLmPure`-function from the `RcppArmadillo`-package. The main idea behind this function is to do calculations within R-functions using a faster programming language than R, in this case C++.

Create a new version of the `is_variable_significant`-function: `is_variable_significant_cpp()`. In this function, you use the `fastLmPure()`-function from the `RcppArmadillo`-package instead of `lm()`. Does the function return the same values as `is_variable_significant()`[4]?. Redo the calculation from assignment 4, but this time using the `is_variable_significant_cpp()`-function. Compare the two implementations, and comment on the (potential) speed gains.

Ensure that the function passes all tests below (which are included in the template).

[3] `https://stackoverflow.com/questions/49732933/fast-method-to-calculate-p-values-from-lm-fit`

[4] Small numerical deviations may indeed happen here, but the results should be practically the same

```r
library(assertthat)

tmp <-
  is_variable_significant_cpp(
    response = "GR6096",
    variable_to_assess = "SOCIALIST",
    covariates = c("COLONY", "CONFUC")
  )

assert_that(tmp$model_call == "GR6096 ~ COLONY + CONFUC + SOCIALIST",
            msg = "Regression specification is not correct")
assert_that(tmp$assessed_variable == "SOCIALIST",
            msg = "Assess variable is not correct")
assert_that(round(tmp$coefficient_estimate, 8) == -0.01305317,
            msg = "The coefficient estimate of SOCIALIST is not correct")
assert_that(round(tmp$p_value, 9) == 0.004308533,
            msg = "The P-value of SOCIALIST is not correct")

rm(tmp)
```

Assignment 6

So far, the coefficient estimates on the term `SOCIALIST` have been negative. Use the complete data and find *any* model where the coefficient estimate for the variable `SOCIALIST` is both postive *and* has a p-value below 0.05 (which, let us reiterate, is a deceptive strategy for telling whatever story that we want and then back it up with data and statistics).

Given that the set of possible models is so large, you need to write a search algorithm to explore the space of possible models. Feel free, for example, to use a completely random search, generating random model specifications until you find a model that fits the description. The only requirement for the models that you try is to use `GR6096` is the response variable and that `SOCIALIST` is included as the explanatory variable for which we check the sign and statistical significance of the estimated regression coefficient.

You should continue to search for models until one of the conditions below are met:

- You have found a model where the coefficient estimate on `SOCIALIST` is both positive and has a p-value below 0.05.
- You have searched for a reasonable amount of time (e.g. 100 seconds) or checked a reasonable number of model specifications (e.g. 100 000 models).

If you find such a model, stop your search and re-estimate the model with `lm()` and present a summary of the regression results. Comment on the credibility of the results. Use any additional modelling choices as you see fit to solve this assignment. It may be useful for you to know that if the model contains too many explanatory variables, then most coefficient estimates will not be significant.

Assignment 7

Write a short paragraph describing how you have used external sources during this exam, including any generative AI models. You can use this paragraph to point out code snippets that you have not written yourself with a reference to the source.

Assessment of answers

The learning outcomes and general competencies defines the targets for the assessment of the exam. Submissions will be ranked by the point system below. The cutoffs between grades will be determined at grading.

- **Read and understand documentation of packages and functions.**
  - 1 point: The submission uses functions/packages not covered explicitly in BAN400
  - 1 point: The submission solves Assignment 5 using `fastLm`.
- **Use basic data structures (lists, arrays, matrices, vectors and data frames) as appropriate. Combine, merge and reshape data sets in R.**
  - 2 points: The submission passes all assert-tests in Assignments 3.
  - 2 points: The submission passes all assert-tests in Assignments 5.
- **Independently resolve warnings, errors, and other basic programming issues.**
  - 1 point: The examiner successfully reproduces the results from the submission, without needing to editing the code.
- **Use functions, loops, assignments, subsetting and conditionals in an R-script.**
  - 1 point: The submission makes use of map-functions as applicable for iterations
- **Use vectorization, iterations and parallelisation as needed computationally demanding tasks.**
  - 2 points: The submissions allows for use of multiple CPU-cores to speed up the time for compilation of rmd-file.
- **Write documented and standardized, formatted code as part of code development.**
  - 1 point: Applies functions to reduce code repetition and improve readability of code.
  - 1 point: Naming of functions and function arguments improves readability of the code.
- **Use R to program and apply selected prediction and machine learning methods and correctly interpret the output in the relevant context.**
  - 1 points: Successfully solves Assignment 2
  - 2 points: Successfully solves Assignment 4
- **Create and export convincing tables and figures for use in reports and presentations.**
  - 1 point: The figures are visually pleasing, appropriately labeled, and informative of the case questions.
- **Apply R to empirical business and economics problems.**
  - 4 points: Successfully solves Assignment 6, with reasonable and explained modelling choices.
  - 1 point: All questions are answered, with code that balances readability and compactness.