Computer Science Tripos

Part II Project Proposal Coversheet

Please fill in Part 1 of this form and attach it to the front of your Project Proposal.

Part 1

Name:	Ria Mundhra	CRSID:	rm2060
College:	Downing	Project Checkers:(Initials)	hg/nk
Title of Project:	Developing a Graphical User Interface for Econometric Analysis		
Date of submission:	15th Oct 2023	Will Human Participants be used?	No
Project Originator:	David Khachaturov		
Project Supervisor:	David Khachaturov		
Directors of Studies:	Dmitrij Szamozvancev		
Special Resource Sponsor:	N/A		
Special Resource Sponsor:	N/A		

Part 2

Project Checkers are to sign and comment in the students comments box on Moodle.

Part 3

For Teaching Admin use only

Date Received:		Admin Signature:	

Developing a Graphical User Interface for Econometric Analysis

Project Description

Econometric analysis plays a vital role in decision-making across various fields, including economics, finance, social sciences, and public policy. Conducting econometric analyses often involves complex data manipulation, statistical modeling, and graph visualization (CFI Team, 2021). Many economists and policy analysts lack extensive programming or software engineering skills, which can be a significant barrier to efficiently conducting econometric research. While econometricians may program in statistical software, often, the code is written in a 'quick and dirty' manner, without long term maintainability in mind, leading to code that is impossible to change or debug (Gentzkow & Shapiro, 2014).

Despite this, most popular econometrics programs are little more than a menu wrapped around a command line. While some point-and-click style programs with Graphical User Interfaces (GUIs) exist – for example, Gauss, SPSS and SAS - they are unpopular because econometricians need to document every data transformation for replicability in empirical research. Point-and-click GUIs make this documentation challenging since they lack a proper way to track every action taken. The naive route most GUIs take is to log every action (Koenker & Zeileis, 2009), however, this leads to messy documentation. In contrast, command-line systems usually rely on a single script to record all steps, making documentation straightforward (Koenker & Zeileis, 2009).

The proposed project aims to develop a novel GUI tailored to the unique needs of econometricians. The GUI will promote:

1. **Modularity:** Since many econometric workflows share common transformation and analysis steps, the GUI should promote reuse, making it easy to save and reload specific analysis units. This would also make debugging easier - one could test and debug each analysis unit separately.
2. **Reproducibility:** Additionally, the GUI should allow users to document and share their pipeline easily, so others can easily reproduce their work.

By providing econometricians with a powerful yet accessible tool, this project seeks to enhance their productivity, promote best practices in data analysis, and ultimately contribute to evidence-based decision-making in economics and related disciplines.

Substance and Structure

Core

To support these requirements, the GUI will have the following core functionality:

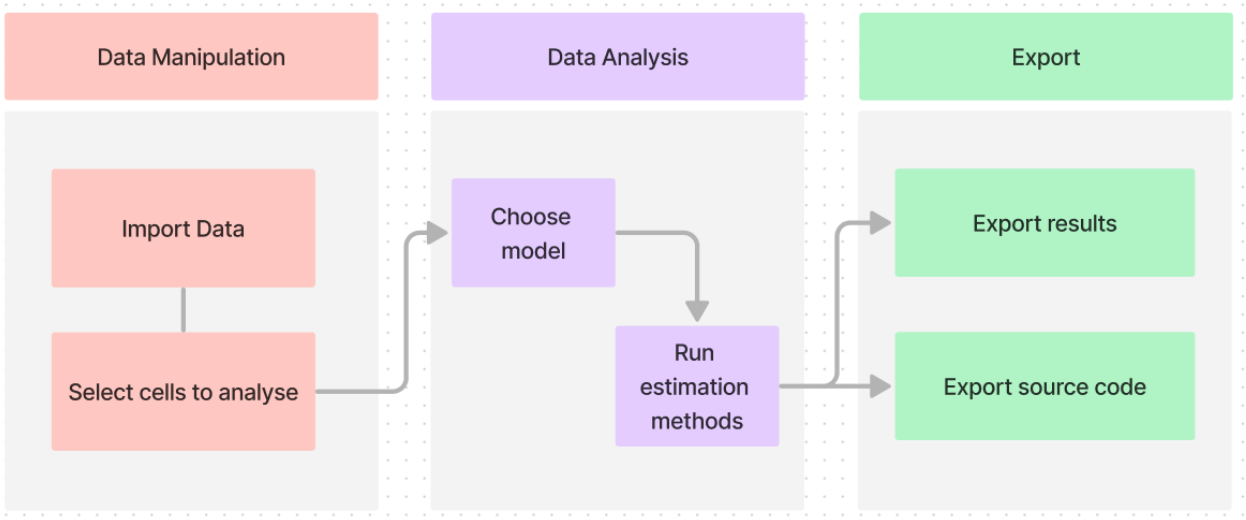
1. **Drag-and-Drop Graph Visualization:** The GUI will feature a drag-and-drop interface for creating, customizing, and analyzing regression models. Users will be able to easily load data from various common formats - csv and .xlsx.

2. **Estimation Methods**: The GUI will integrate estimation methods for regression models - least squares estimation and maximum likelihood estimation. Users will have the ability to specify parameters directly within the interface, streamlining the econometric workflow.

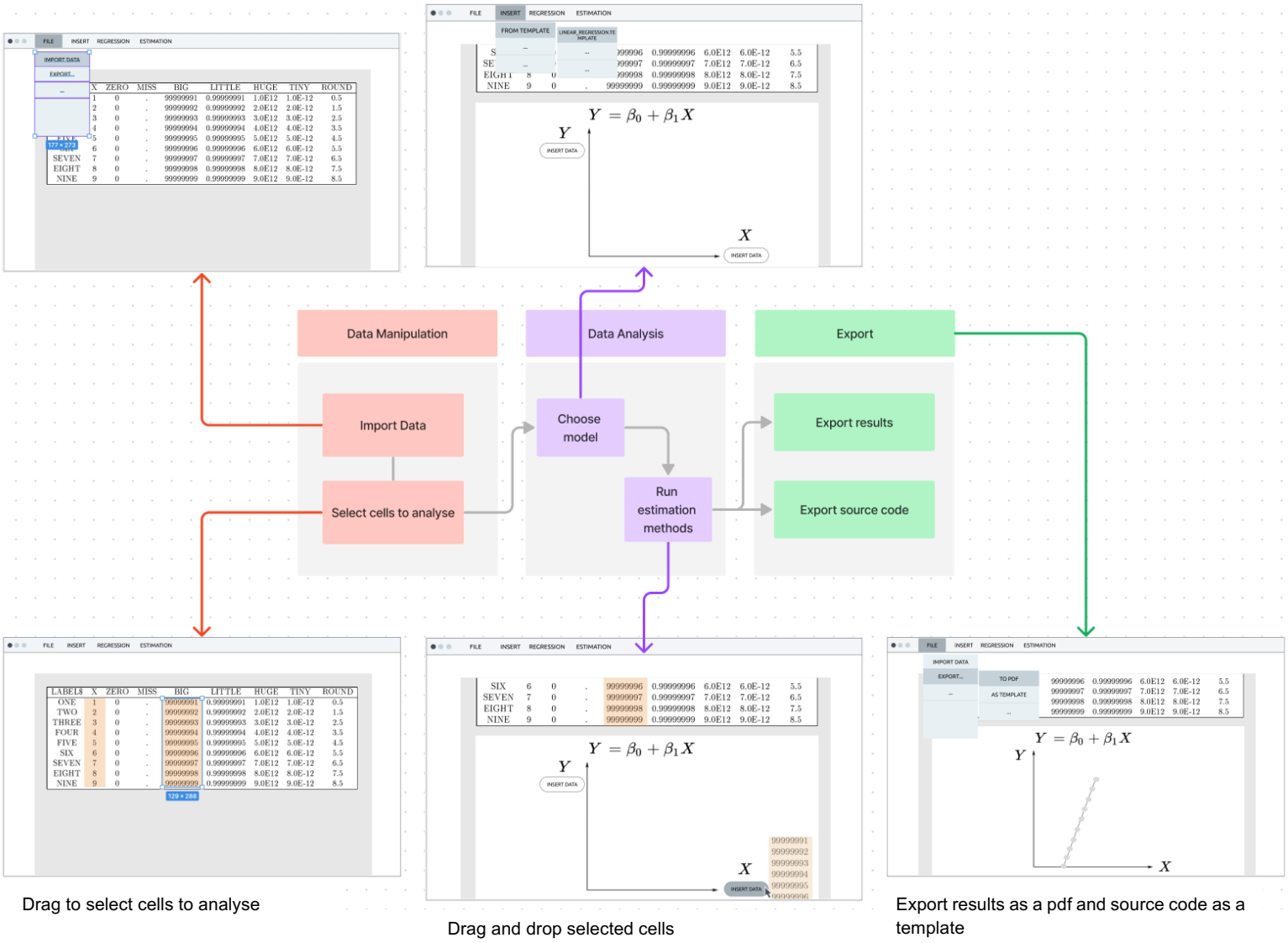
3. **Templates**: The GUI will enable users to save and reuse templates from all or parts of their current pipeline. This feature promotes modularity since template parts act as stand-alone modules, and reproducibility since a template for the full analysis pipeline acts as documentation of the analysis.

4. **User-Friendly Documentation**: The GUI will be accompanied by documentation. The documentation will be easily searchable, making it accessible to beginners and advanced users alike (Kipyegen & Korir, 2013). Additionally, commands within the GUI will be designed to be easy to remember, minimizing the learning curve.

Take the following common workflow:



The functionality proposed supports each step in the workflow as follows:



The bulk of the functionality comes into play in the data analysis step, where the ability to use templates, and the ease of selecting data and running regressions and estimation methods saves time spent coding and debugging. Additionally, the templating feature makes exporting source code easy, as all the methods can be bundled up into a single template file.

The GUI will be built using the React.js web app framework, and then ported to both macOS and Windows using the Electron library, which ports web apps to desktop applications.

I will be using statistical JavaScript libraries, such as Simple Statistics to implement the regression and estimation methods, and libraries such as CoreUI, React DnD, and React Paper to build the UI. All of these are freely available on npm and are compatible with a React.js project.

Note that I will not be carrying out user studies, rather, I will be using theoretical techniques, such as heuristic evaluation, to evaluate my GUI.

Extension

Additionally, based on the progress of the core project, I will implement extensions if time permits. Some possible extensions include:

1. Support for more data formats

2. Support for more models and estimation methods

3. Ability to handle larger datasets, up to 10000 rows

4. Smooth switching between GUI and code views for more advanced users who wish to make more granular modifications.

Starting Point

I have considerable experience in front end development, having worked extensively with the MERN stack as a member of the programming specialization of my high school robotics club. I have also had experience with frontend development in a production environment during my internship at the Defence Science and Technology Agency of Singapore. Additionally, I have attended the *Interaction Design* course in Part IA, and the *Further Human-Computer Interaction* course in Part IB.

Additionally, having scored an A in Economics during my Singapore-Cambridge GCE A levels, I have a substantial foundation in the field of economics. I also have a personal interest in applied economics, although I have not delved into econometrics prior to this project.

During my internship at the Defence Science and Technology Agency, I had the chance to learn about and apply data analytics techniques. I am therefore familiar with hypothesis testing and simple linear regression methods. I have also taken the *Data Science* course in Part IB.

Several existing econometric packages and software are freely available to download from the Internet, ranging from low-level Command Line Interfaces (CLIs) to high-level Point and Click GUIs. I will be using these as reference when building my own GUI.

- **Low-level:** R, EViews CLI
- **High-level:** SPSS, SAS, EViews GUI

Moreover, other software tools have tried to use templates to bridge the gap between intuitive use and reproducibility. For example, Gephi, a network visualization and manipulation tool, allows users to save their current work into a session, which encapsulates all transformations on the

data into a single file (Gephi, 2010). This will again serve as a reference when building the functionality for my own GUI.

There are also open-source React.js libraries available to build the data import and export and drag-and drop functionalities. While I will be using components from these libraries, they will be modified to fit the needs of the project.

Success Criteria

A working drag-and-drop GUI for graph visualization that supports:

1. Import and export from .csv and .xlsx data formats.
2. Ability to handle data sets of up to 1000 rows.
3. Visualization of two regression models - simple linear regression and multivariate regression
4. Implementation of least squares (OLS) estimation and maximum likelihood estimation methods
5. Saving all or parts of the current analysis as a template that can be reloaded into the GUI.

Additionally, the GUI will be accompanied by documentation on a Wiki engine for all important functionalities – this includes all regression and estimation functions available to users but excludes any internal functions.

All core functionalities will be unit tested. For example, a unit test for the import/export functionality would import a dataset, make no changes, export it in the same format, and check that the exported file is structurally equivalent to the imported file.

The accuracy of the GUI will be evaluated against example problems from The Wilkinson's Tests (McCullough), as well as any other common workflows identified during informal conversations with econometricians and students studying econometrics.

Additionally, the GUI will be benchmarked against existing software by drawing a comparison across several features, including open-source availability, availability of visual interfaces, ease of exporting source code. I will also use heuristic evaluation to evaluate my GUI.

Project Timeline

The project will be divided into work packages, each with specific objectives, deliverables, and milestones. The approximate timeline for each work package is as follows:

Michaelmas Term

Week 1-3

- Finalize and submit project proposal.
- Conduct background research on existing econometrics software.
- Define user requirements through surveys and interviews.
- Compile a comprehensive list of desired features.

Milestone: Completed user requirement documentation and submitted project proposal.

Week 4-5

- Develop initial GUI design mockups and wireframes.
- Seek feedback from potential users and make design adjustments.
- Create a functional prototype.

Milestone: Finalized GUI design prototype

Week 6

- Set up code repository
- Implement data import functionality for supported formats
- Create basic GUI window that is able to read and display imported data
- Test with sample datasets in various formats

Milestone: Successful data import functionality and a well structured code repository.

Week 7-8

- Implement drag-and-drop graph tools.
- Incorporate options for adjusting graph types, axes properties, and labels.
- Test graph customization functionalities with user input.

Milestone: Functional graph customization tools.

Michaelmas Vacation

Week 1-2

- Integrate solver capabilities into the GUI.
- Develop interfaces for specifying solver parameters.
- Test solver integration with econometric models.

Milestone: Integrated solver functionality.

Week 3-4

- Develop modular workflow features, such as saving templates.

Milestone: Functional modular workflow features.

Week 5-6

- Write unit tests based on the Wilkinson's Tests to test accuracy of the GUI.
- Write unit tests for the use of templates.

- Identify and address any bugs or issues.

Milestone: Bug-free GUI

Week 7

- Slack time to tie up any loose ends.

Milestone: All core functionality of GUI is fully implemented and tested

Lent Term

Week 1-2

- Write progress report.
- Create and rehearse presentation.

Milestone: Progress report and presentation submitted.

Week 3-4

- Create user-friendly documentation for the GUI.
- Ensure all commands are easy to remember and use.

Milestone: Completed user documentation.

Week 5-8

- Slack time to tie up any loose ends in the core of the project.
- Work on selected extensions
- Start writing implementation chapter of dissertation.

Milestone: Core of project fully implemented and tested. At least one extension implemented. Implementation chapter of dissertation written.

Lent Vacation

Week 1-2

- Continue working on selected extensions.
- Write unit tests for extensions.
- Write evaluation chapter of dissertation.

Milestone: All extension work complete and tested. Evaluation chapter of dissertation written

Week 3-5

- Write introduction, conclusion and preparation chapters of dissertation.
- Proofread main body of dissertation.

Milestone: Main body of dissertation written and proofread.

Week 6

- Slack time
- Ensure dissertation format follows requirements, including writing the cover sheets, declaration of originality, proforma, etc.

Milestone: Dissertation completed and proofread.

Easter Term

Week 1-2

- Submit dissertation and source code.

Resource Declaration

I intend to use my own computer development work and research. The specifications are as follows:

MacBook Pro 2018, 2.3 GHz Quad-Core Intel Core i5 Processor, 8 GB RAM, 250 GB SSD

To protect against data loss, all files and code will be pushed to GitHub after any major change is made. This also serves as version control. In case my machine fails, I will use the MCS machines.

Additionally, I will be writing my dissertation using a mixture of LaTeX and Microsoft Word on my own computer. I will back up all files to Google Drive after any major change.

I accept full responsibility for this machine and I have made contingency plans to protect myself against hardware and/or software failure.

Bibliography

Gentzkow, M., & Shapiro, J. M. (2014). Code and data for the social sciences: A practitioner's guide. *Journal of Economic Perspectives,, 28*(2), 197-221.

Mackie-Mason, J. K. (1992). Econometric Software: A User's View. *Journal of Economic Perspective, 6*(4), 165-187.

Koenker, R., & Zeileis, A. (2009, 7 3). On reproducible econometric research. *Journal of Applied Economics, 24*(5), 833-847.

CFI Team. (2021, May 11). *Econometrics - Overview, How It Works, Examples.* Retrieved September 17, 2023, from Corporate Finance Institute: https://corporatefinanceinstitute.com/resources/economics/econometrics/

Kipyegen, N. J., & Korir, W. P. (2013, September). Importance of Software Documentation. *International Journal of Computer Science Issues, 10*(5), 223 - 228.

Gephi. (2010, March 5). *Gephi Tutorial Quick Start.* Retrieved September 18, 2023, from Gephi: https://gephi.org/tutorials/gephi-tutorial-quick_start.pdf

Kotwani, P. (2022, May 7). *Create desktop app with React Native.* Retrieved September 20, 2023, from Medium: https://medium.com/@Pankaj.Kotwani/react-native-for-windows-macos-50ac79a9c48e

McCullough, B. D. (n.d.). *Wilkinson's Tests and Econometrics Software.* Washington: Federal Communications Commission.