Ethan Hunt

Sean Hasse

Will Stratton

Christian Cervantes

Vince Badali

Software Requirement Specification

1. Introduction
   1. Purpose

The program we create will take in a dataset in a csv file, and optionally a p-value threshold, and will calculate the level of correlation, described as p-value. It will output the p-value. It has a GUI used to control, and display the output. Due to the large quantity of data, this program's job is to do this algorithm as quickly as possible in an effort to make the refined data more readily available.

* 1. Scope

The program will need to be able to handle an input dataset of unique columns (identified by strings) filled with either strings or floating point vectors. The datasets are clean, so it is not in the scope of this project to account for missing values, data with the wrong type, or datasets with the wrong number of rows/columns. This program must be written as a C++ library function so that it may be called by other programs.

* 1. Definitions, acronyms, & abbreviations

Chi-Square: a tool used to find the relationship between categorical variables

Chi-Square test: a non-parametric test that analyzes correlation between categorical variables

p-value: a measurement of correlation between two categories of data. A higher p-value equates to a higher level of correlation between the categories.

* 1. References

<https://www.mathsisfun.com/data/chi-square-test.html> (from the PP)

* 1. Overview

The purpose of our Software Engineering Project is to design, create and implement a program that can run a chi-square analysis algorithm on a given dataset at good speed and with high accuracy. We will be able to accept the desired input of a CSV file and output the proper p values of each column in the GUI. The GUI must be easy enough to understand so that any worker can use it.

1. Overall description
   1. Product perspective
      1. System interfaces

The program will need user interfaces to communicate with the consumer, and the consumer will be running the program on a Virtual Machine. We will not be using any Software interfaces.

* + 1. User interfaces

The user should be able to control the p-value threshold, the file dataset used as input. The GUI will provide many forms of feedback. We will include the number of rows, categorical columns and numerical columns of the input dataset. This will be output to the user only if the values supplied fall within the p-value threshold.

Alongside this, we will include the number of correlated pairs. This number will be output in a separate part of the gui and exists to show how many pairs exist within the p-value threshold.

Also included in our GUI will be the top 10 pairs, ranked by correlation coefficient. This will be done regardless of whether or not all 10 fall within the p-value threshold. If the dataset is not large enough to create 10 pairs, this section will include as many as possible.

Lastly, our output will include the execution time of the program in milliseconds.

* + 1. Hardware interfaces

The only hardware interfaces will be the virtual machine that the consumer will be running our program on.

* + 1. Software interfaces

None.

* + 1. Communications interfaces

None

* + 1. Memory constraints

None

* + 1. Operations

The operations that can be done are:

* Select csv file
* enter acceptable p values
* select “enter” to calculate
* receive p value.
  + 1. Site adaptation requirements
  1. Product functions

Product functions include: taking input datasets via csv file and constraints, performing the Chi-Square function on the set and displaying the output of the most correlated categories that were read.

* 1. User characteristics

The documentation of this function must be easy enough to understand for a programing intern to use it effectively .

* 1. Constraints

Must be able to perform chi-tests efficiently and accurately. GUI must be easy to navigate, and easy enough for the user to be able to work with the GUI. Code must also be highly readable with enough comments to explain what it does. Code must have 100% test coverage. (We could include all of the requirement slides here, but that may be overkill)

* 1. Assumptions and dependencies

We assume that the given datasets are clean with no bad data. We assume the datasets will be given as .csv forms. We assume that all cells in contingency table must have >= 1 frequency, and at least 80% of cells must have >= 5 frequency.

* 1. Apportioning of requirements

None.

1. Specific requirements

Appendixes

* 1. External interfaces

The only external interface that we will have is the GUI. The GUI will be implemented in C#, allowing the user to upload the csv file. The GUI will display all of the output to the user.

* 1. Functional requirements - organized by feature, object, user class, etc
  2. Performance requirements

All functions must have 100% test coverage, except the main() function

R’s chisq.test() and kruskal.test() functions from the standard package will be used to provide the ‘standard answer’ for testing your code’s correctness. The difference shall be within 0.1 for test statistics and 0.001 for p values.

* 1. Logical database requirements

None.

* 1. Design constraints
     1. Standards compliance
  2. Software system attributes

The requirements in this section specify the required reliability, availability, security and maintainability of the software system.

* + 1. Reliability

The software will not crash, instead if a file could not be read the appropriate error message will be displayed. The software should be able to get a correct result from a valid input after running the Chi Square test. If the input is not valid the software should catch that and ask the user for a valid input or display an error message.

* + 1. Availability

The software should be available at all times, meaning it can be accessed any time. The function specifically must be constructed so the larger system can call the Chi-square function at any time.

* + 1. Security

The software does not have any security requirements and so any type of user can use it without any additional privileges.

* + 1. Maintainability

Will be added when complete

* + 1. Portability

The software shall be designed to run on Windows, and Ubuntu. We will write all of the software in C++ , and there will be no platform specific code.

* 1. Organizing the specific requirements
     1. System mode -- or

Will be added when complete

* + 1. User class -- or

Will be added when complete

* + 1. Objects (see right) -- or

Will be added when complete

* + 1. Feature -- or

Will be added when complete

* + 1. Stimulus -- or

Will be added when complete

* + 1. Response -- or

Will be added when complete

* + 1. Functional hierarchy -- or

Will be added when complete

* 1. Additional comments

Some of the sections are left blank, as we continue progression on the software we will go back and fill in some of the missing areas.