**Event Data TwoRavens**

**Documentation and User Manual**

**Introduction:**

**TwoRavens** is a system of interlocking statistical tools for data exploration, analysis, and meta-analysis. The first to be released is an interface for quantitative analysis, that allows users at all levels of statistical expertise to explore their data, describe their substantive understanding of the data, and appropriately construct statistical models. This integrates with Dataverse [Project](http://dataverse.org)|[GitHub](https://github.com/IQSS/dataverse)) and Zelig ([Project](http://zeligproject.org)|[GitHub](https://github.com/IQSS/Zelig)), through a portable, lightweight, browser-based and gesture-driven interface, allowing users to run statistical models available in Zelig on data archived in Dataverse.

The project page is available at <http://2ra.vn>

Usually, TwoRavens has been created to work with tabular data of smaller scales. But, the problem we try to tack with Event Data TwoRavens is to give the user tools to browse through large datasets, primarily consisting of event coded data. Event Data is data coded from news events, curated through either news websites, or archived print media news reports. The four main components of a typical event data dataset are **Actor**, **Location**, **Date** and **Event type**. Let us explain the four briefly:

**Actors:**

**Location:** The location signifies the place where the event in question occurred. The granularity of the location can be anywhere from the country where the event took place, to the locality of the occurrence of the event.

**Date:** The date on which the event occurred.

**Event Type:**

**Where does TwoRavens come in:** A typical Event Data dateset contains millions of records of coded events, which grow by the day as and when new news reports are being coded.

TwoRavens provides a way to easily select and analyse that data using its familiar GUI architecture, and giving the user control over what the user wants to see, what he/she wants to do with it.

There are potentially three functions TwoRavens aims at implementing in analyzing EventData:

**Subset**: As there are millions of entries in the dataset, we have to provide the user with tools to select and filter data according to their requirement. The User can select data according to a specific source actor, target actor, location, date and event type, to start with. The user can do whathe/she wants with the selected data. He/She can download the data, or use the TwoRavens Architecture to further analyse the data.

**Aggregation**: Even if the user has selected data at the subset stage, the data is still raw, and cannot be analyzed at stage. TwoRavens provides tools to count the events according to a temporal unit(date) or spatial units (actors, locations). The choice of the aggregation unit is completely on the user.

**Analysis**: After the data is aggregated, it is ready to be analysed using time series analysis. The analysis models are implemented into TwoRavens, giving the user the option to customize his/her models and then estimate results.

**The Project**

**Setting up**

To start working on TwoRavens, you need to set it up first. The installation steps are very easy, and you will need very little time to do that.

Requirements:

* Github account, and preferably a github GUI client(For windows) or git installed(for linux or macOS)
* R software, and RStudio. R can be downloaded on <https://www.r-project.org>. RStudio can be downloaded from <https://www.rstudio.com/products/rstudio/download> Note: Rstudio is an IDE for R, so it’s your choice to download it.
* Python 2.7. It can be downloaded from <https://www.python.org/downloads/release/python-2712/>

Note: Linux usually has python installed. Use python -V to see if python is installed. For Windows users, make sure you check the option “add python.exe to system path” during installation.

Additional notes: Windows blocks cross domain requests between client scripts and server even if CORS(cross-origin-resource-sharing) is enabled in the request and response headers. And because the R server serves on a different port than the python server, it is very important to enable cross origin resource sharing. Linux based operating systems and macs do not have this problem. So, it is suggested to set up the project on a virtual ubuntu machine if you have windows as your primary OS. You can download Oracle virtual box from <https://www.virtualbox.org/wiki/Downloads> and ubuntu from [www.ubuntu.com/download](http://www.ubuntu.com/download) .

Once you have all of the above, do the following steps to setup tworavens locally:

1. Open the terminal/command prompt, and type

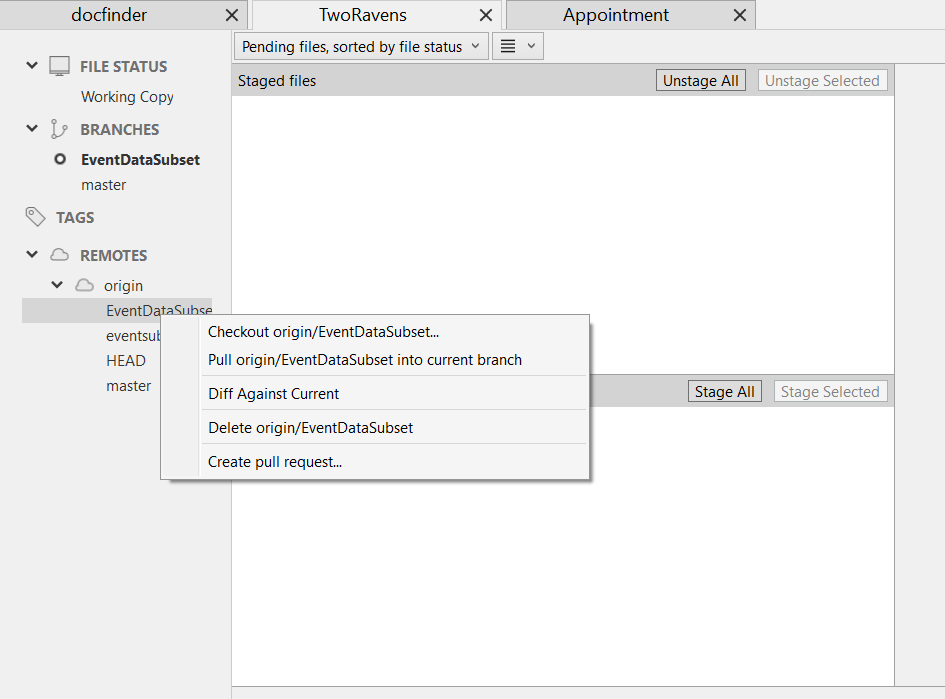
git clone <https://github.com/vjdorazio/TwoRavens.git>

if you are using a github GUI like source tree or smart git,, click on clone/new repository, and enter the above URL when prompted.

1. Once you have cloned TwoRavens, you will need to change the branch from master to EventDataSubset. This branch contains the codebase for Event Data TwoRavens. To do that, in the terminal/command prompt, type

git checkout EventDataSubset.

For GUI clients, look for branches under origin, right-click on the EventDataSubset branch, and select checkout.



1. Open R,either by typing R into the terminal for mac/linux users, or by selecting the R program for windows, and Execute the following with R to install R packages:

install.packages(c("VGAM", "AER", "dplyr", "quantreg", "geepack", "maxLik", "Amelia", "Rook","jsonlite","rjson", "devtools", "DescTools", "Zelig"))

1. After the installations are done, set the working directory to TwoRavens/rook. For example,

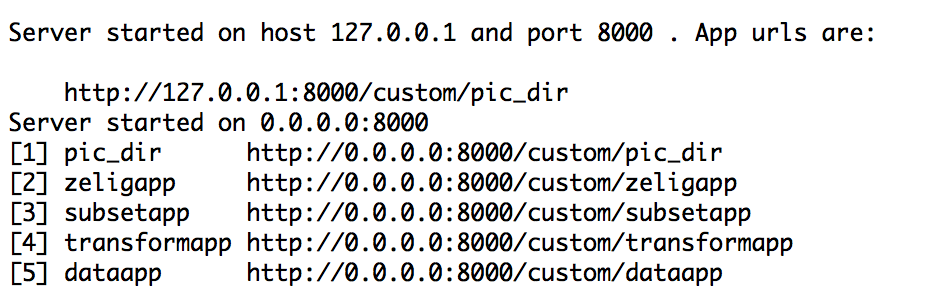
setwd(“/Users/vjdorazio/Desktop/github/TwoRavens/rook”) for linux/mac or

setwd("C:/users/rohit/documents/TwoRavens/rook") for windows.

1. To run the R server, type

source(“rooksource.R”)

This will give a large output, as it will install some more dependent packages when you run this for the first time. The final output should look like this:



Note: Try doing this in the console provided by R, and not Rstudio. R studio uses the default 8080 port for its own purposes sometimes, so it might throw an error that the port is already in use. You can use RStudio for editing the R scripts, but try running this on the R console.

1. Once this server is up and running, launch the terminal/command prompt, and navigate to the TwoRavens directory. For example,

For linux/mac users: cd Desktop/github/TwoRavens

For windows users: cd C:/users/rohit/documents/TwoRavens/rook

After navigating there, you will need to start a server, which we will do using python. Type in the following command:

python -m SimpleHTTPServer 8888 &

This should give an output: Serving HTTP on 0.0.0.0 port 8888 ...

1. Once this is up and running, you are setup! Go to the browser, and type in <http://localhost:8888/gui3.html> to open the TwoRavens Event Data page.