

# A Guide to Empirical Measures

Ao Dong

June 9, 2020

This document is a general guide to the implements of empirical measures.

There are several sections in this guide, with one tool to use in each of them, and usually measures different aspect of a software package. There is no mandatory order to use which tool first.

## 1 git\_stats

### 1.1 Introduction

Name: git\_stats

Source Code: [GitHub repo](#)

### 1.2 User Manual

Official Manual: [GitHub repo](#)

### 1.3 Demo of Installation and Running the Tool

This is the showcase of how to install and run this tool.

The installation steps on your machine may be different from this section. Please refer to Section [1.2](#).

Hardware: a virtual machine with 8 cores and 16 GB RAM

OS: Debian GNU/Linux 9.11

1. Install ruby/gem environment

```
apt-get install ruby ruby-nokogiri ruby-nokogiri-diff ruby-nokogumbo
```

Check the installation:

```
gem --version
```

2. Install the tool

```
sudo gem install git_stats
```

### 3. Prepare the target repo

Make sure the target repo (the repo to be analyzed, not the repo of this tool) is on your machine. In this demo, the target repo is downloaded from a GitHub repo:

```
# change [git path] to the url of your target repo
git clone [git path]
# e.g. git clone https://github.com/nroduit/Weasis.git
```

### 4. Generate analytics

```
# make sure [repo path] is the target repo path
# the [output path] can be anywhere you desire
git_stats generate -p [repo path] -o [output path]
# e.g. git_stats generate -p /home/user/git-stats/Weasis -o
    /home/user/git-stats/Weasis-analytics
```

### 5. View the analytics

View the analytic results by open [output path]/index.html with any browser or other software supporting HTML web page format.



### 6. Download the data

On most of the tabs of this web page, the data can be downloaded for more analytics by clicking the menu button beside the data-range section.

## 2 scc

### 2.1 Introduction

Name: scc

Source Code: [GitHub repo](#)

### 2.2 User Manual

Official Manual: [GitHub repo](#)

### 2.3 Demo of Installation and Running the Tool

This is the showcase of how to install and run this tool.

The installation steps on your machine may be different from this section. Please refer to Section 2.2.

Hardware: a virtual machine with 8 cores and 16 GB RAM

OS: Debian GNU/Linux 9.11

#### 1. Install Golang

Follow the [official instructions](#), or the following demo,  
download the installation package:

```
wget https://dl.google.com/go/go1.14.3.linux-amd64.tar.gz
```

unpack to /usr/local:

```
sudo tar -C /usr/local -xzf go1.14.3.linux-amd64.tar.gz
```

use a text editor to open ~/.profile, e.g.:

```
nano ~/.profile
```

add the following lines to the end of this file:

```
export GOPATH=$HOME/go
export PATH=$PATH:/usr/local/go/bin:$GOPATH/bin
```

save the file, and load the commands into the current shell instance:

```
source ~/.profile
```

check the installation:

```
go version
```

## 2. Install the tool

```
go get -u github.com/boyter/scc/
```

## 3. Prepare the target repo

Make sure the target repo (the repo to be analyzed, not the repo of this tool) is on your machine. In this demo, the target repo is downloaded from a GitHub repo:

```
# change [project path] to your desired folder
cd [project path]
git clone https://github.com/nroduit/Weasis.git
```

## 4. Generate analytics

```
# make sure [repo path] is the target repo path
cd [repo path]
# use scc to generate analytics
scc
```

## 5. View the analytics

The results will be directly shown.

Language	Files	Lines	Blanks	Comments	Code	Complexity
Java	745	129067	17549	13709	97809	18207
Properties File	119	7632	411	1116	6105	0
XML	113	9291	163	272	8856	0
Plain Text	44	10778	50	0	10728	0
Shell	7	609	77	112	420	43
Markdown	6	2205	516	0	1689	0
XML Schema	4	529	53	0	476	0
License	2	288	51	0	237	0
gitignore	2	45	8	8	29	0
YAML	1	19	3	1	15	0
Total	1043	160463	18881	15218	126364	18250
Estimated Cost to Develop \$4,348,103						
Estimated Schedule Effort 26.819132 months						
Estimated People Required 19.204800						