

# JQ

[jq](#) is a lightweight and flexible command-line JSON processor. It can make processing and viewing json significantly easier. To make desirable functions easier, this repository has a directory called `jq` that contains a bunch of common queries, that can all be invoked by `jq -f jq/<filename>.jq`. Below is a description of some of them.

## Expand Strategies (`expand_strats.jq`)

This filter takes as input an egta simulation spec file, except in assignment instead of a list of strategies, the input is a dictionary of strategy to count. This makes it much easier to generate simulation spec files that have a large number of one type of agent. An example input looks like:

```
{
  "assignment": {
    "role": {
      "strategy": count,
      ...
    },
    ...
  },
  "configuration": {
    ...
  }
}
```

## Social Welfare (`social_welfare.jq`)

This filter takes as input an observation file (or sequence of them), and outputs the social welfare (sum of all of the player payoffs) of each observation.

## Role Strategy Payoffs (`role_strat_payoffs.jq`)

This filter takes as input an observation and returns an object mapping roles to strategies to average payoff for that role strategy pair. An example output looks like:

```
{
  "role": {
    "strategy": average payoff,
    ...
  },
  ...
}
```

## Merge Payoffs (`merge_payoffs.jq`)

This filter takes as input an array of observation files (or the standard output with the slurp (`-s`) option, and merges them into one observation file with the average payoff for each player. Alternatively, one can group observations together, in which case this can be used somewhat like `numsims` to condense input.