

# rnassqs: An R package to access agricultural data via the USDA National Agricultural Statistics Service (USDA-NASS) 'Quick Stats' API

Nicholas A. Potter<sup>1</sup>

<sup>1</sup> Washington State University

DOI: [10.21105/joss.01880](https://doi.org/10.21105/joss.01880)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Kyle Niemeyer](#) ↗

## Reviewers:

- [@kyleniemyer](#)

Submitted: 08 November 2019

Published: 17 November 2019

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

## Summary

The [rnassqs](#) R package provides a simple interface for accessing the United States Department of Agriculture National Agricultural Statistics Service (USDA-NASS) 'Quick Stats' API. The core functionality allows the user to query agricultural data from 'Quick Stats' in a reproducible and automated way. The primary benefit of [rnassqs](#) is that users need not download data through repeated use of the Quick Stats point-and-click interface, which reduces the chance of errors, eliminates the need for repeated manual downloads of new data over time or space, and allows for automated updates of web applications that rely on new releases of USDA-NASS data over time.

[rnassqs](#) manages API authentication by setting a system environmental variable for the duration of the R session. Convenience functions facilitate querying common data. Users can also use [rnassqs](#) to query the list of data parameters and available values for a given parameter (for example, to see the commodities available in a particular county and year). The query requests data as a JSON object and parses that data into a `data.frame` object.

## About USDA-NASS data and 'Quick Stats'

'Quick Stats' is a web interface to access data produced by USDA-NASS. The data comes primarily from the Census of Agriculture, but also includes data from USDA-NASS surveys on a wide range of topics. The Census of Agriculture is conducted every five years in years ending in '2' and '7'. The earliest year available on Quick Stats is 1997. Surveys have different collection periods, but most are collected annually. Some specific data such as the average value of agricultural land and buildings by state is reported as early as 1850.

Aggregate data from the census and surveys is released primarily at the national, state, and county level, though some data may be released for congressional districts, watersheds, and zip codes. It includes a range of data classified under five sectors: Animals & Products, Crops, Demographics, Economics, and Environmental. Examples of data available in these sectors include counts of farms, farm operators, acres of cropland, farm sales, farm expenses, and crop yields to name a few.

## Benefits of [rnassqs](#) over 'Quick Stats'

'Quick Stats' provides a number of selection fields in which the user can select categories of data. Each selection causes the options in other fields to update to reflect available options

available options based on other selections that the user has made. This makes the selection of multiple different variables an at times frustrating process. In addition, data requests are limited to 50,000 records. If a user wants to access more records they must manually subset their data request. For example, requesting crop yields by county for all counties and census years for 1997 to 2017 requires either downloading each state's counties separately or, where possible, downloading all counties for each year separately. The Quick Stats interface works well for quick access or a single use. However, there are several cases in which the 'Quick Stats' interface is not ideal:

- Requests for a combination of measures, years, and geographies that exceed 50,000 records.
- Requests for newly released data that are identical to previous requests.
- Requests that are reproducible.

`rnassqs` addresses each of these issues by making the 'Quick Stats' API accessible with R code. This allows the user to loop over a series of requests to address the first issue, to execute (perhaps automated on a schedule) a data request repeatedly to access new data with the same query to address the second, and to make code available that allows others to reproducibly access the same data to address the third.

For example, there are currently 332,125 records of crop yields in all U.S. counties from 2000 to 2018. Accessing this data through 'Quick Stats' would require manually selecting either a set of years or a set of states to reduce each request to less than 50,000 records and then aggregating that data. With `rnassqs` this can be done with:

```
# Access yields for all counties and all crops
params <- list(sector_desc = "CROPS",
              group_desc = c("FIELD CROPS", "FRUIT & TREE NUTS",
                             "HORTICULTURE", "VEGETABLES"),
              statisticcat_desc = "YIELD",
              agg_level_desc = "COUNTY")

# Get all years from 2000 to 2018 in a list of data.frames
data_list <- lapply(2000:2018, function(yr) {
  params$year <- yr
  rnassqs::nassqs(params, url_only = TRUE)
})

# Aggregate the list of data.frames into a single data.frame
d <- do.call("rbind", data_list)
```

This results in significant time savings, increases the reproducibility of the data, and allows for easy updating of the request when a new year is made available.

## Alternatives to `rnassqs`

USDA-NASS also provides FTP access to text data files<sup>1</sup>. By accessing the data via FTP users can avoid using the selection interface of 'Quick Stats' and avoid limitations on the number of records per request, but do not resolve issues of automated repeated requests or of making data requests reproducible. Other R packages have also been released since the development of `rnassqs` that provide access to Quick Stats data, though somewhat differently, include `usdarnass` and `tidyUSDA`, which uses US Census maps to create maps of Quick Stats data.

<sup>1</sup>Available at: <ftp://ftp.nass.usda.gov/quickstats/>

## Acknowledgements

Thank you to Jonathan Adams, Julia Piaskowski, and Joseph Stachelek for code contributions. A huge thanks to reviewers Adam H Sparks and Neal Richardson for their thoughtful feedback to improve the package and documentation.