

dataSPA Package: A Package for Obtaining Data for DFO Strategic Planning

Jaimie Harbin

The dataSPA package is intended to be used by Department Of Fisheries and Oceans Canada (DFO) employees to obtain data from A) the Project Planning Tool (PPT) API and B) Salary Spread sheets to plot information related to allocation of funding.

Some disclaimers/ assumptions with this package include:

1. For this package to work, you must be on the DFO VPN
2. This code assumes some employees have not changed positions over the course of their career
3. The salary calculations are based off of medians of federal employee pay scales, not the exact salary of individuals

To install the package, you must first be a collaborator of the private repository **dataSPA** hosted by **j-harbin** (Jaimie Harbin). Once you have been granted access, run the following code to install the package:

```
install.packages("devtools")
devtools::install_github("j-harbin/dataSPA", repo = "main")
library(dataSPA)
```

Once the package is installed, the user should follow the following workflow to successfully use the package:

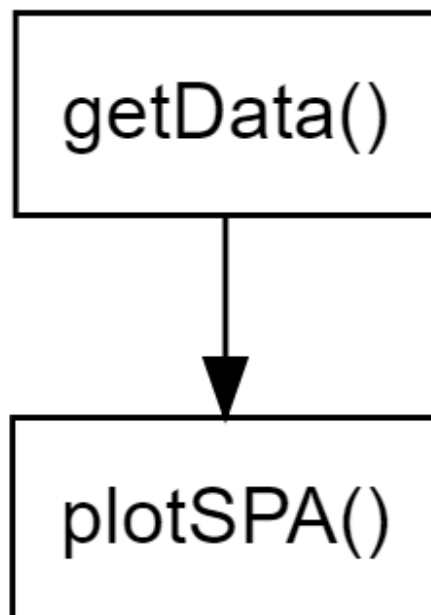


Figure 1: Figure 1: Work flow for the dataSPA package

getData()

If **type** is **om**, this function extracts data from the om-costs project planning tool (PPT) API (<http://dmapps/api/ppt/om-costs/>) and combines it with the fiscal year from the project_years API (<http://dmapps/api/ppt/>). If **type** is **salary**, this function obtains the data from the staff API (<http://dmapps/api/ppt/staff/>) and combines it with information from a Human Resources (HR) spreadsheet.

It should be noted that this function will only work for authorized Department of Fisheries and Oceans Canada (DFO) employees. As shown in the help documentation for this function (`?getData`), a **cookie** is required. A cookie can be obtained by DFO staff by following the steps outlined below:

1. Sign into DMAPps on the internet
2. Right click in DMAPps and click “Inspect Panel”.
3. Click the » arrows next to “Welcome”, “Elements”, “Console”, and “Sources” to locate “Application” and click
4. Click on your csrftoken and sessionid and copy it
5. Assign a variable in R with the following format:

```
cookie <- "csrftoken=YOURTOKEN; sessionid=YOURSESSIONID"
```

This variable **cookie** will then act as your **cookie** argument in the `getData()` function, e.g.:

```
om <- getData(type="om", cookie=cookie)
```

plotSPA()

This function plots specific graphs (determined by the **which** argument) using the data frames returned by `getData()`.

The various plot types are as follows:

- For **which="omBar"**, a bar graph representing amount of money (\$) coded by funding type.
- For **which="omPie"**, a pie chart representing amount of money (\$) coded by funding type
- For **which="omAllocation"** a bar graph representing amount of money (\$)
- For **which="salaryBar"** a bar graph representing amount of salary money (\$) using the median values of a specific job classification. Graph color coded by funding type indicated by the legend.
- For **which="salaryAllocation"** a bar graph representing amount of salary money (\$) using the median values of a specific job classification.
- For **which="weekAllocation"** a bar graph representing time (weeks)
- For **which="indeterminate"** a pie chart representing proportion of indeterminate vs non-Indeterminate employees for project years is plotted

To plot the amount of money invested in O&M per funding type for the Snow crab project, for example, the code is as follows:

```
plotSPA(om, which="omBar")
```

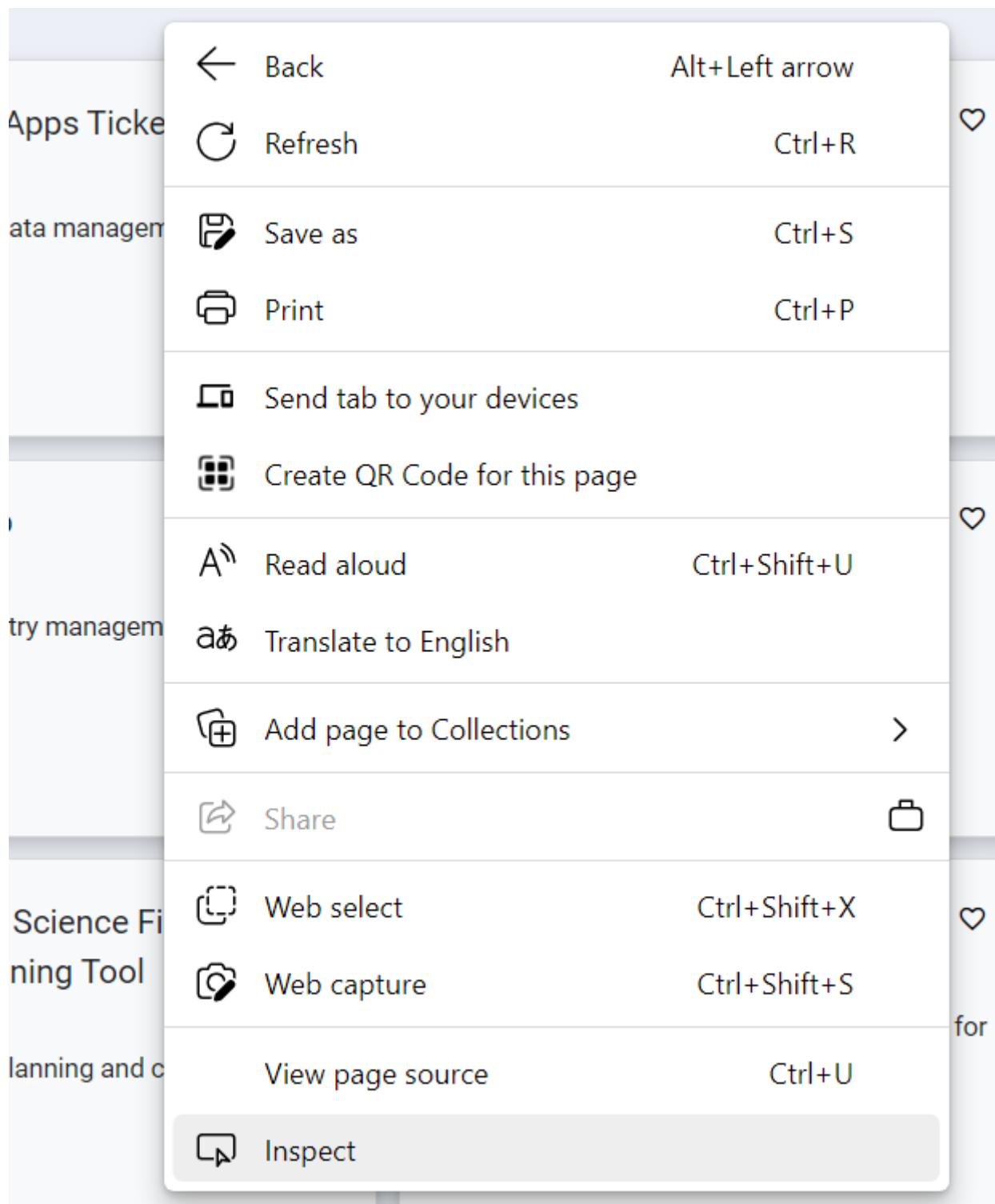


Figure 2: Figure 2: Right click to find “Inspect”

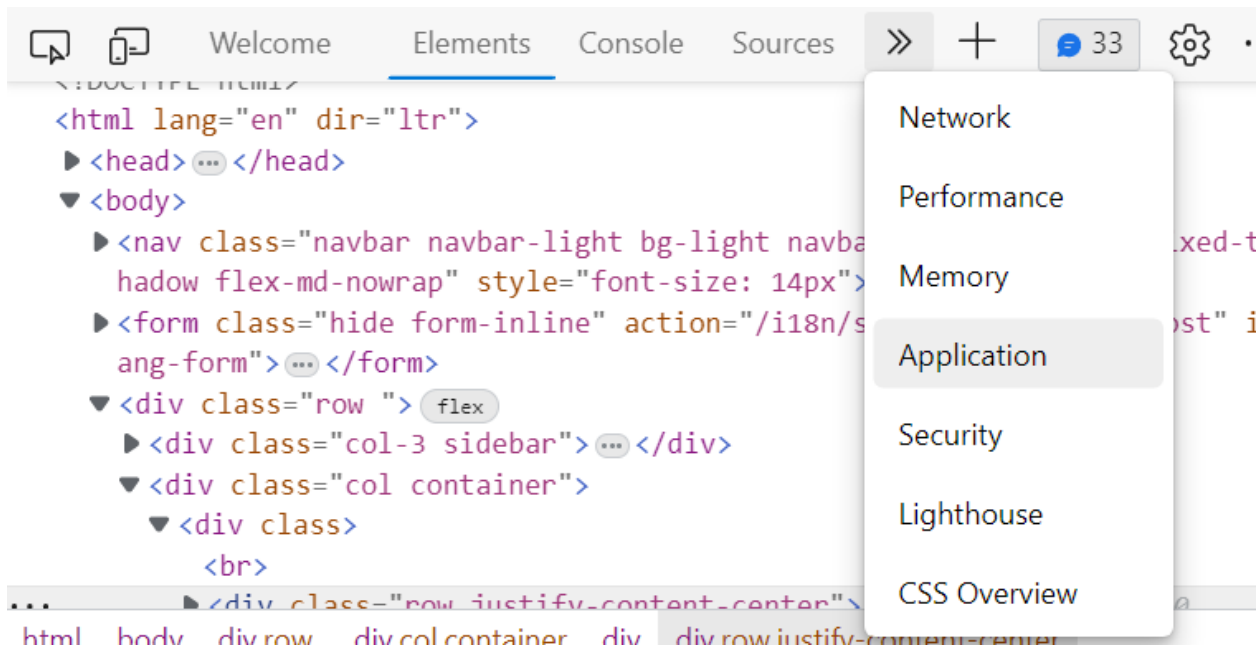


Figure 3: Figure 3: Click » to find “Application”

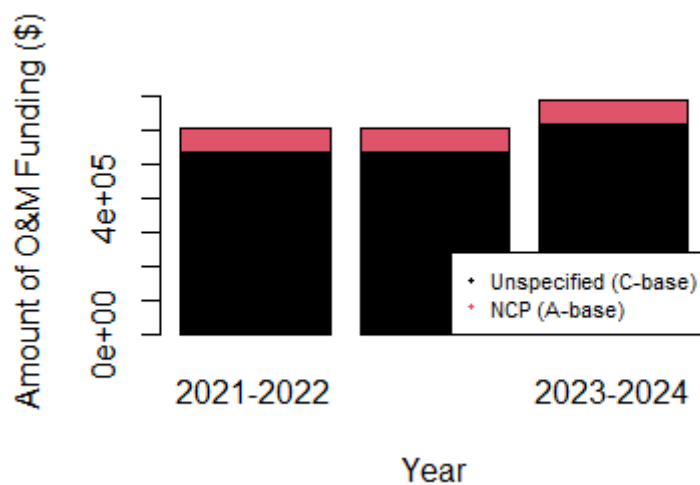


Figure 4: Figure 3 Bar graph representing amount of money (\$) invested in O&M per funding type for project years 2021-2022, 2022-2023 and 2023-2024 for the snow crab project (# 1093). Graph color coded by funding type indicated by the legend.