



STATE OF THE
SALMON PROGRAM

SPAT Tool Kit: Correlation Analysis

Gottfried Pestal (SOLV) & Michael Barrus

2020-03-31

Abstract

The Salmon Pattern Analysis Toolkit (SPAT) is being developed as part of broader set of analytical tools under DFO's State of the Salmon Program. This report describes the toolkit structure and the first interactive application, which focuses on time series correlation analyses. The work was funded by the Southern Endowment Fund (SEF) of the Pacific Salmon Commission (PSC) under Project Code SF-2019-FRP-26, and through in-kind support provided by Fisheries and Oceans Canada (DFO).

Table of Contents

1	Project Overview	5
1.1	The <i>Salmon Pattern Analysis Toolkit (SPAT)</i>	5
1.2	2019/2020 SEF Project	5
2	User Surveys	6
2.1	Why Interact with End-Users?	6
2.2	Completed Surveys	6
2.2.1	Establishing the scope	6
2.2.2	Preliminary Design	7
2.2.3	Show & Tell Sessions	7
2.2.4	In-Depth User Testing	7
2.3	Summary of Feedback and User Observations	8
2.4	References	8
3	SPAT-Corr App	9
3.1	Introduction	9
3.2	Launching the <i>SPAT-Corr</i> App	9
3.2.1	Online	9
3.2.2	Local Launching	9
3.3	App Layout	10
3.4	App Components	11
3.4.1	Load Data	11
3.4.2	Explore Groups	12
3.4.3	Explore Pairs	14
3.4.4	Correlation Matrix	19
3.5	References	23
4	Worked Example - Package	24
4.1	Introduction	24
4.1.1	Functions	24
4.1.2	Data Sets	24
4.2	Install & Explore	25
4.3	Worked Example	25
5	Sample Data Set: SPATData_EnvCov	27
5.1	Introduction	27
5.2	Data Structure	27
5.3	Variables	28
5.3.1	Environmental Covariates	28
5.3.2	Fraser Sockeye Productivity Variables	28

6	Feature Requests.....	29
6.1.1	Workflow.....	29
6.1.2	Load Data.....	29
6.1.3	Explore Groups Task.....	30
6.1.4	Explore Pairs Task.....	30
6.1.5	Correlation Matrix Task.....	30
6.1.6	New Task: Explore Individual Series.....	30
6.1.7	Other.....	30

1 Project Overview

1.1 The *Salmon Pattern Analysis Toolkit (SPAT)*

The overall goal is to build a tool kit for guided exploration of salmon population data.

Beyond the scope of the current project, the long-term plan includes the development of a generalized flow chart of data exploration steps, completion of R-Software functions for each data exploration step, and then connecting these functions in a single user-friendly Shiny R app.

The short-term priority is to build a few simpler standalone Shiny R apps for individual tasks (e.g. exploring pairwise correlations) and to develop versatile and robust package functions for each of the analytical steps in those tasks.

1.2 2019/2020 SEF Project

The first phase of development was funded by the Southern Endowment Fund (SEF) of the Pacific Salmon Commission (PSC) under Project Code SF-2019-FRP-26 and supported by in-kind support provided by Fisheries and Oceans Canada (DFO) through the State of the Salmon Program.

There are 3 main products from this work, summarized in this report:

- A flexible toolkit structure designed for incremental (modular) development. This was accomplished by setting up an R package for the statistical functions.
- An interactive tool for correlation analysis. This was implemented in Shiny and is available either online or for offline use after installation.
- Feedback and observation from user testing, and resulting revisions of the prototype.

2 User Surveys

2.1 Why Interact with End-Users?

While data visualization is used extensively in salmon research and management, most visualizations are designed by individuals to illustrate a specific phenomenon or dataset. In these cases, the visualization designer can rely on their expertise to decide if the visualization achieves its relatively singular goal (i.e. “Does this make my point more clear?”). In contrast, the success of SPAT’s visualizations is more difficult for an expert to judge without consulting users. The purpose of the tool is not to illustrate a single point, but to facilitate exploration and understanding of user-selected datasets. Our expertise alone is not a sufficient judge of this user-defined success, and so we need to solicit feedback from end-users throughout the development process.

To guide the design of the tool, we employed a few best-practice, user-focused methodologies. Our principal process was *Design Study Methodology* (DSM). DSM is the gold standard for the creation of data visualization systems. In the words of the original authors, DSM provides a framework “in which visualization researchers analyze a specific real-world problem, design a visualization system that supports solving this problem and validate the design” (Sedlmair et al. 2012). DSM prescribes a series of iterative steps to follow. The first stage is discovery, where the designers consult with experts to learn about the problems they need to solve (described in Section 2.2.1). Following this, we designed the interface (Section 2.2.2), implemented the design in R-Shiny (Chapter 3), then gathered feedback and iterated on the design (Sections 2.2.3 and 2.2.4). In summary, the DSM advocates for clearly defining your users and their goals, making a broad study of the possible visualization solutions, implementing the best solution, and testing that solution with users to evaluate its success and iterate based on those results. Our specific execution of these steps is described below.

2.2 Completed Surveys

2.2.1 Establishing the scope

Learning about the problems facing users is the first step of the DSM process (Lam et al. 2018, Borkin 2011). During this ‘discovery’ phase, the visualization developer consults experts to learn about the problems that visualization can solve for them. To accomplish this, we conducted semi-structured interviews throughout 2018 and early 2019 with approximately 35 salmon experts from multiple agencies and across a variety of roles. These experts were employed by DFO and the US Government, and included staff working in salmon science, management, enhancement, and habitat management. This process enabled us to characterize the data and analysis-related problems resource managers and scientists encounter during their work, what tasks they are required to perform, and what outcomes they hope to achieve. Our user surveys provided guidance on the information to incorporate into the final tool as well as how each participant would like to interact with that information.

Following these structured interviews, we compiled a list of all within-scope tasks described by users. These tasks were grouped by similarity and combined when possible to come up with “task abstractions”. An example of abstracting a task would be to take the specific task of “compare current test fishery numbers to historic data for test fishery numbers on a similar day” and abstract it into “Compare trends over time”. Getting rid of specifics like “test fishery numbers”, “historic data” and “similar day” allows us as visualization designers to think not about the specific case being described, but all cases where the user has to behave similarly, and develop a solution for all those similar cases, rather than each individual case. Abstracting a task allows a designer to move past differences of language or specifics, and think in terms of idea, intentions and needs, which will help the designer consider a broader range of solutions.

From the larger list, we decided to address three of the most common ones. They were as follows:

- 1) Compare salmon trends based on user-defined filters: on spatial and temporal scales, lifehistory traits, or management aggregations;
- 2) Find populations with similar patterns as a population selected by the user;
- 3) Group populations into hierarchical clusters based on synchrony of patterns.

2.2.2 Preliminary Design

With the scope of the tool defined, we conducted a design session with two DFO scientists in order to lay out the basic design of the tool. During this session, we came up with specific questions based on their data that fell into each of the three categories, and then discussed their ideal visualization solutions and workflow for each. We created simple whiteboard prototypes, which outlined the visual idioms they wanted to see in the tool and their preferred workflow. Based on these whiteboard prototypes, we designed a preliminary wireframe for first app that incorporated the design suggestions of expert users and our understanding of the best visual idioms for the given tasks. Ideas from this prototype helped guide the design and interface as the prototype for the correlation analysis app *SPAT-Corr* evolved.

2.2.3 Show & Tell Sessions

Several small-group presentation sessions with facilitated discussions were conducted throughout February and March, 2020, in order to introduce the app to potential end-users and elicit high-level feedback on app features and design.

Presentation sessions reached many salmon experts from diverse specialties.

- *Pacific Salmon Commission*: Fiona Martens, Catherine Michielsens, Pasan Samarasin, Julie Sellars
- *DFO - Chinook Tech Committee Members*: Michael Folkes, Nicholas Komick, Maxine V., Antonio Vélez-Espino, Catarina Wor
- *DFO - State of the Salmon*: Sue Grant, Bronwyn MacDonald, Brigitte Dorner
- *DFO - Basin and Coastal Scale Interactions Program*: Erika Anderson, Lyse Godbout, Jackie King

2.2.4 In-Depth User Testing

The final phase of soliciting feedback on the app prototype involved user experience (UX) testing, where participants actively use the app, rather than watch a presentation. This testing procedure puts the *Spat-Corr* app in front of a potential end user, and we ask them to complete certain tasks with the tool. We watch them perform the tasks in order to understand how real users use the tool and how we can improve it. The purpose of this testing is not to get hard quantitative data, but instead to see the tool in use so that we can understand what dimensions of it need improvement.

A guided user testing session was conducted with PSC staff on March 13, 2020. Participants included Catherine Michielsens, Fiona Martens and Eric Taylor. Sue Grant and Bronwyn MacDonald of DFO's State of the Salmon program observed remotely.

The session followed a standard UX testing script, with the following key elements:

- A user was shown the tool and asked to give their general impressions of it.
- They were then given a sample task and asked to complete it while thinking aloud.
- Observers record both the user's statements related to usability and their own observations of any usability issues.

Following the session, the observers discuss any usability issues they observed during the testing session and agree on how to prioritize those issues. That priority list is used to guide the redesign of the tool in future iterations.

2.3 Summary of Feedback and User Observations

There was a general consensus among participants that the *SPAT-Corr* app is a useful tool for data exploration and has the potential to support different use types, including rapid data displays (e.g. in a working group meeting) and hypothesis exploration (e.g. streamline the scoping process for detailed analyses).

Participants offered differing perspectives on the many options provided in the app. On one end of the spectrum, concerns were raised about the potential for users to take data out of context or blindly try all the options until they find a correlation, without regard for what's biologically plausible. The opposite perspective was that different hypotheses always come up in various forums anyway, and the app could assist with rapidly debunking spurious relationships (e.g. by showing how correlations have changed over time, or how sensitive they are to data transformations). After discussions, participants generally agreed that the tool should be as flexible as possible but needs to come with clear guidance to non-expert users regarding what to watch out for. This will be addressed through a series of worked examples to be published on the app repository. One worked example is included in Chapter 3.

Participants also identified the need for a detailed manual describing all the statistical analyses and options available for each step (e.g. alternative data transformations, alternative correlation coefficients). This will be addressed in 3 different formats:

- Chapters 3 and 4 of this document provide an overview of the analytical steps.
- The help files for the functions of the *SPAT R Package* include details for expert users.
- General descriptions of the methods for each step will be included in a stand-alone manual, once the functions and app stabilize.

Specific topics that require clarification are:

- different correlation approaches
- clustering in the correlation matrix
- retrospective correlations plots

Active use of the app faced several practical hurdles ranging from input data formatting to app layout (e.g. menus, task sequence) and structure (e.g. carry over of settings between tabs, resetting when moving backwards in the sequence). These were compiled and prioritized for the next phase of development.

User testing really highlighted the amount of work on the app interface that will be necessary to reduce the learning curve for new users. Apps always seem much more straight-forward to the developers than they are for end-users.

Chapter 6 summarizes specific feature requests identified through the user surveys.

2.4 References

Borkin, M. 2011. *Evaluation of Artery Visualizations for Heart Disease Diagnosis*. IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 12, pp. 2479-2488, Dec. 2011.

Lam, H., Tory, M. and Munzner, T. 2018. *Bridging from Goals to Tasks with Design Study Analysis Reports* in IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 435-445, Jan. 2018.

Sedlmair, M., M. Meyer and T. Munzner, "Design Study Methodology: Reflections from the Trenches and the Stacks," in IEEE Transactions on Visualization and Computer Graphics, vol. 18, no. 12, pp. 2431-2440, Dec. 2012

3 SPAT-Corr App

3.1 Introduction

Interactive applications are becoming more widely used in fisheries, as recent software developments make it possible for scientists to build powerful and robust online applications with a minimum of web-specific programming skills. For example, the Shiny extension of the widely used R language automatically handles most of the challenges that used to be major obstacles to implementation (e.g. browser and operating system compatibility, adjusting layouts for device screen size).

However, a survey of currently available apps shows that most fisheries/oceanography/limnology applications still focus on data sharing and visualization, rather than harnessing the full analytical potential of these new tools. For more information, see the inventory of apps at <https://github.com/SOLV-Code/GreyFish/tree/master/RESULTS/AppTaxonomy>

The long-term vision for the *Salmon Pattern Analysis Toolkit (SPAT)* is to build a series of related apps, each specialized for particular type of data and analysis. Based on the priorities identified through our user surveys (Chapter 2), we decided to build an app for correlation analysis of time series data as the first tool in the tool kit. This app links to a series of recent influential papers on productivity covariation (Dorner et al. 2018, Malick and Cox 2016, Peterman and Dorner 2012, Peterman et al. 1998, Pyper et al. 2001, Pyper et al. 2005), as well as the environmental covariates being used for salmon forecasts (Vélez-Espino et al. 2019, DFO 2018, Grant and MacDonald 2013).

The *SPAT-Corr* app allows users to load in their own time series data, compare data series side-by-side, explore correlations over time between 2 series, and cluster series in a correlation matrix. The app comes with a built-in data set for illustration, which includes productivity for 18 stocks of Fraser River Sockeye salmon and various environmental covariates (e.g. river discharge, sea surface temperature). Chapter 5 describes the built-in data set.

The source code for the *SPAT-Corr* app is publicly available in the app repository at <https://github.com/SOLV-Code/SPAT-Apps>. The repository also includes a wiki with up-to-date descriptions of the apps, as well as an issues page where users can report bugs, ask questions, and request additional features.

3.2 Launching the *SPAT-Corr* App

3.2.1 Online

The quickest way to get started with the app is to use the online version in a web browser. No software installation is required, and users don't have to be familiar with R. However, some of the plots will be slower to load with the online version.

The online version of the *SPAT-Corr* is currently available at https://solv-code.shinyapps.io/spat_correlationanalysis/, and mirror sites (e.g. via the PSC) may be set up once the app stabilizes and user numbers increase. The app repository at <https://github.com/SOLV-Code/SPAT-Apps> includes up-to-date links to all available mirror sites.

3.2.2 Local Launching

Running the app locally improves speed and will not require internet access once loaded. However, it requires the internet for downloading the app code.

There are 2 options for running the apps locally. For both you have to first download the app code from the repository at <https://github.com/SOLV-Code/SPAT-Apps>. On the repository site, click on the green "Clone or Download" button and select "Download ZIP". Extract the zip folder and open the subfolder *CorrelationAnalysis*.

RStudio

RStudio is set up to work smoothly with Shiny apps, so the easiest way to run the app locally is to open it from within RStudio. You need to download RStudio from <https://rstudio.com/products/rstudio/download/#download> and install it, then install the *SPATFunctions* package using the code below.

```
install.packages("devtools") # Install the devtools package
library(devtools) # Load the devtools package.
install_github("SOLV-Code/SPATFunctions-Package",
              dependencies = TRUE,
              build_vignettes = FALSE, force = TRUE)
library(SPATFunctions)
```

Then open the file *ui.R*, and click "Run App" to launch the app.

R GUI

The app also runs from within basic R. For this approach you need to download R from <https://www.r-project.org/> and install it. Open R in the root folder of the downloaded repository (i.e. the zip file from earlier), then open and run the launch script *1_LaunchGUI.R*. This installs the *SPATFunctions* package and then launches the app.

3.3 App Layout

The *SPAT-Corr* app is structured as a sequence of 4 main tabs, each corresponding to one step of the correlation analysis workflow:

- *Data Loading*: Load in your own data set, or look at the built-in data set. Note that data may need to be either lined up based on assumed mechanisms. For example, using a Sockeye productivity time series, marine environmental data can be lagged to correspond to ocean entry timing for a particular brood year. This can be done ahead of time, before loading the data into the app, or inside the app. Either way, it is the users responsibility to keep track of how the different time series fit together.
- *Explore Groups*: Look at one or more time series side-by-side, and try out various aggregations (e.g. mean across selected series) to generate new time series for use in the later steps.
- *Explore Pairs*: Select two of the time series, look at them side-by-side, and check how the correlation between the series has changed over time.
- *Correlation Matrix*: Check a summary plot of all the pair-wise correlations in the data set, or select a subset of series.

The rest of this chapter describes the menu options on each tab and illustrates each step with a worked example that carries all the way through.

The example has the following steps:

- *Explore Groups*: Create some new time series based on the input data (e.g. normalized time series of median sea surface temperature across April, May, and June, offset by 2 years).
- *Explore Pairs*: Compare the patterns and correlations of the new time series.
- *Correlation Matrix*: Select the new time series, and a few others, then cluster the series based on correlations.

3.4 App Components

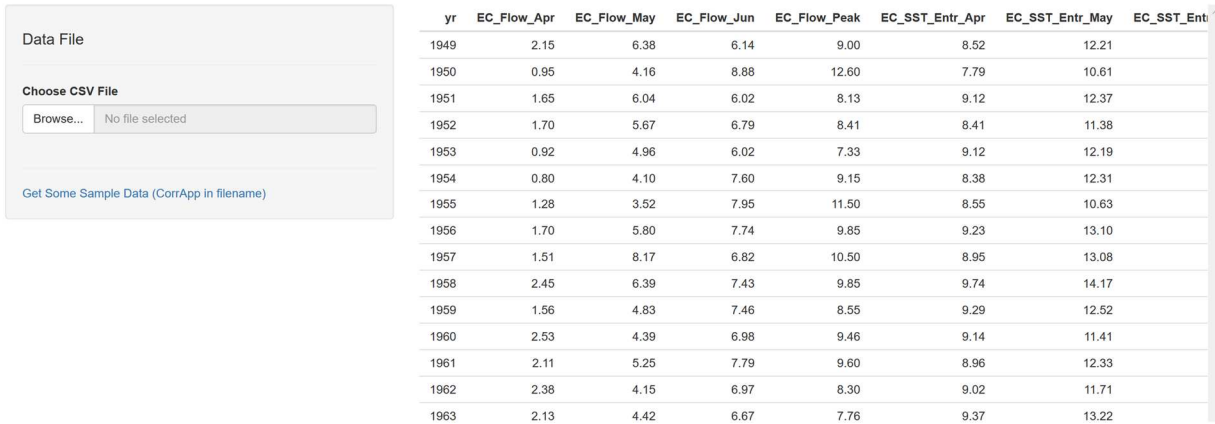
3.4.1 Load Data

The app comes with a built-in data set of Fraser sockeye productivities and environmental covariates. Chapter 5 describes the data set and its required structure. Note that the environmental covariates in this data set are matched to a calendar year, and have not been matched to the productivity series to line up brood years with smolt downstream migration and ocean entry timing by year. The data set has the actual data, and adjustments can be made inside the app, as shown in the worked examples below.

Users can also load in their own data sets, but the structure has to be the same as the built-in data set. Note that some pre-cleaning of your data may be necessary before loading it in. For example, all numeric variables in the input data become elements of the drop-down menus in the app. A data file with many variables (i.e. columns) can make the selections unwieldy, so it may be more efficient to work with subsets of your data.

The tab also includes a link to a dropbox folder with other sample data files, but note that not all of them work with this specific app. Only the ones with *CorrApp* in the filename will work properly.

Data Loading



yr	EC_Flow_Apr	EC_Flow_May	EC_Flow_Jun	EC_Flow_Peak	EC_SST_Entr_Apr	EC_SST_Entr_May	EC_SST_Entr
1949	2.15	6.38	6.14	9.00	8.52	12.21	
1950	0.95	4.16	8.88	12.60	7.79	10.61	
1951	1.65	6.04	6.02	8.13	9.12	12.37	
1952	1.70	5.67	6.79	8.41	8.41	11.38	
1953	0.92	4.96	6.02	7.33	9.12	12.19	
1954	0.80	4.10	7.60	9.15	8.38	12.31	
1955	1.28	3.52	7.95	11.50	8.55	10.63	
1956	1.70	5.80	7.74	9.85	9.23	13.10	
1957	1.51	8.17	6.82	10.50	8.95	13.08	
1958	2.45	6.39	7.43	9.85	9.74	14.17	
1959	1.56	4.83	7.46	8.55	9.29	12.52	
1960	2.53	4.39	6.98	9.46	9.14	11.41	
1961	2.11	5.25	7.79	9.60	8.96	12.33	
1962	2.38	4.15	6.97	8.30	9.02	11.71	
1963	2.13	4.42	6.67	7.76	9.37	13.22	

Figure 1: Data Loading Tab

3.4.2 Explore Groups

The *Explore Groups* task has 4 identical tabs (Group 1 to Group 4), where users can select 1 or more time series from the input data (*Numerical Variables* dropdown), modify the series (transform, offset), and then create an aggregate index across the series. On the fifth tab users can download the data set with the added time series.

The following transformations are currently available:

- *log*: Take the natural log of each series. This is useful for skewed data (i.e. a long tail in the histogram), because it “squishes down” the large values. 0 or negative values will result in NA.
- *z-score*: Rescales the axis without changing the pattern. This is useful when you want to combine series of different magnitude (e.g. looking for a underlying shared pattern across environmental variables with different units).
- *percent rank*: Assigns 0 to the smallest value, 1 to the largest value, and rescales the remaining values based on their percentile in the distribution. You can think of this as roughly like a combination of the log transform and the z-score.

Other transformations have been requested during user testing (Chapter 2) and will be added during the next round of development.

The offset slider shifts the year axis left or right. This does not affect how the series selected on this tab line up against each other, but it will change how the aggregate index generated from this group will line up with other series. This is useful if assumptions about underlying mechanisms include a time lag. For example, if you are checking whether sea surface temperature (SST) during ocean entry is linked to salmon productivity, then you would offset the SST series by two years for most Fraser sockeye stocks, such that SST in 2002 is lined up with the 2000 brood year.

By selecting one of the options on the *aggregate index* dropdown, users can overlay a new time series based on some summary across the selected series. This is useful when checking for underlying shared patterns. For example, salmon productivity can be highly variable from year to year, but similar stocks may share a common underlying long-term pattern. Similarly, monthly sea surface temperature measurements can be noisy, so correlations with salmon productivity may be more clearly pronounced if you use the median across several months.

The text box at the bottom of the menu allows user to add a custom label for the aggregate index.

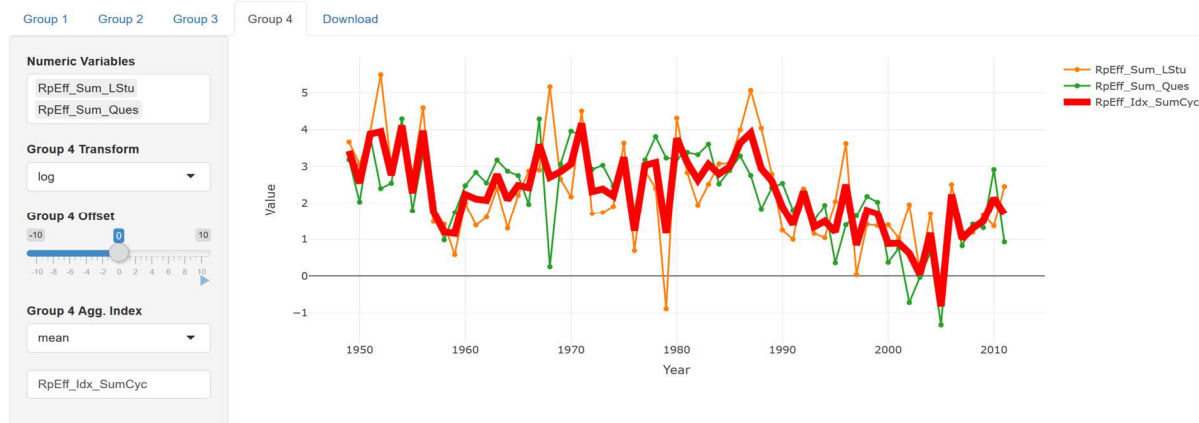


Figure 2: *Explore Groups* Tab - Productivity Example

Figure 2 shows the log-transformed time series of productivity (recruits per effective female) for the 2 cyclic stocks in the Summer management unit (Late Stuart, Quesnel), with the mean productivity overlaid. The mean productivity index is labelled as *RpEFF_Idx_SumCyc*, consistent with the naming conventions suggested in Chapter 5.

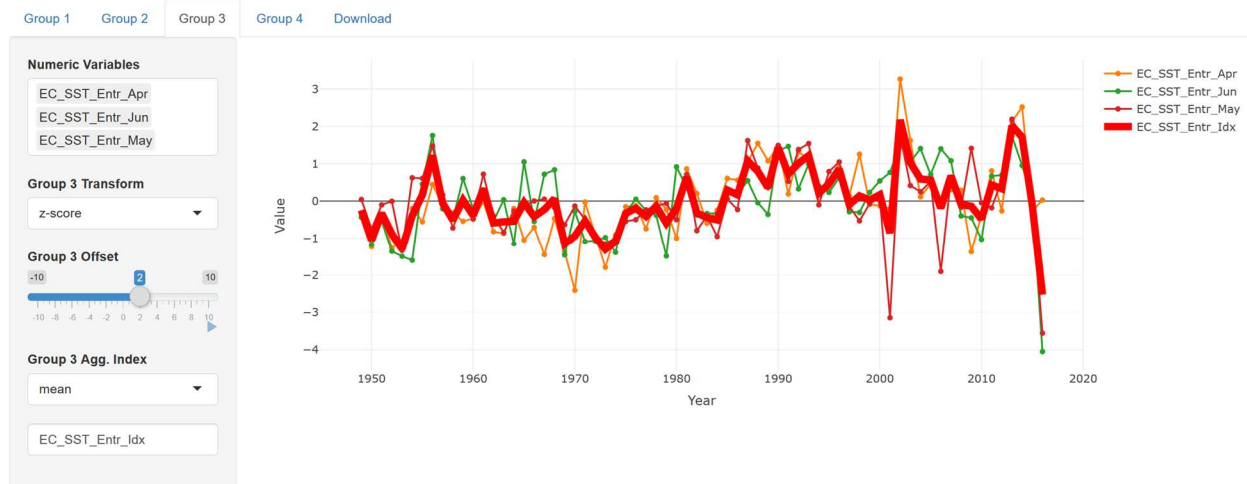


Figure 3: Explore Groups Tab - SST Example

Figure 3 shows the normalized (z-score) time series of sea surface temperature (SST) at Entrance Island for April, May, and June, with the mean normalized SST. The mean SST index is labelled as *EC_SST_Entr_Idx*, consistent with the naming conventions suggested in Chapter 5.

3.4.3 Explore Pairs

The *Explore Pairs* tab has dropdown menus where users can select 2 time series (*Var 1*, *Var 2*). The menus include all numeric variables from the input data set as well as any time series created on the *Explore Groups* tab.

Each time series can be transformed, with the same options listed in the previous section. The offset slider can be used to shift the second time series relative to the first time series.

The top panel shows a plot of the 2 selected time series. Users can currently select 1 of 3 layout options:

- *single*: Show both time series in a single figure. This default works when the 2 series have the same units and similar values.
- *2 panels*: Show each time series in a panel with its own axis.
- *2 axes*: Show both time series in a single figure, but with a secondary axis for the second variable.

Users have the option to show/hide a correlation plot, which tracks the correlation between the two series over time. The blue line with point markers shows the cumulative correlation (i.e. all data points since the beginning of the time series). The grey line shows the correlation for a moving time window (e.g. sliding 12 yr intervals). The length of the time window can be adjusted with the *window* slider. Horizontal dashed red lines mark positive and negative correlations larger than 0.5, as a visual guide for interpreting the pattern.

Values higher up on the plot show a stronger *positive* correlation (i.e. *larger* values of Variable 2 tend to be associated with *larger* values of Variable 1). Values lower down on the plot show a stronger *negative* correlation (i.e. *larger* values of Variable 2 tend to be associated with *smaller* values of Variable 1). Values around zero indicate no relationship one way or the other.

Each plot also has interactive features (via the *plotly* implementation). The interactive menu shows up when you hover the mouse over the plot. Users can zoom in/out, read off data values, reset the axes, and take a print screen snapshot of the plot.

Other displays and interactive features have been requested during user testing (Chapter 2) and will be considered during the next round of development (e.g. scatter plot, box plot, violin plot).

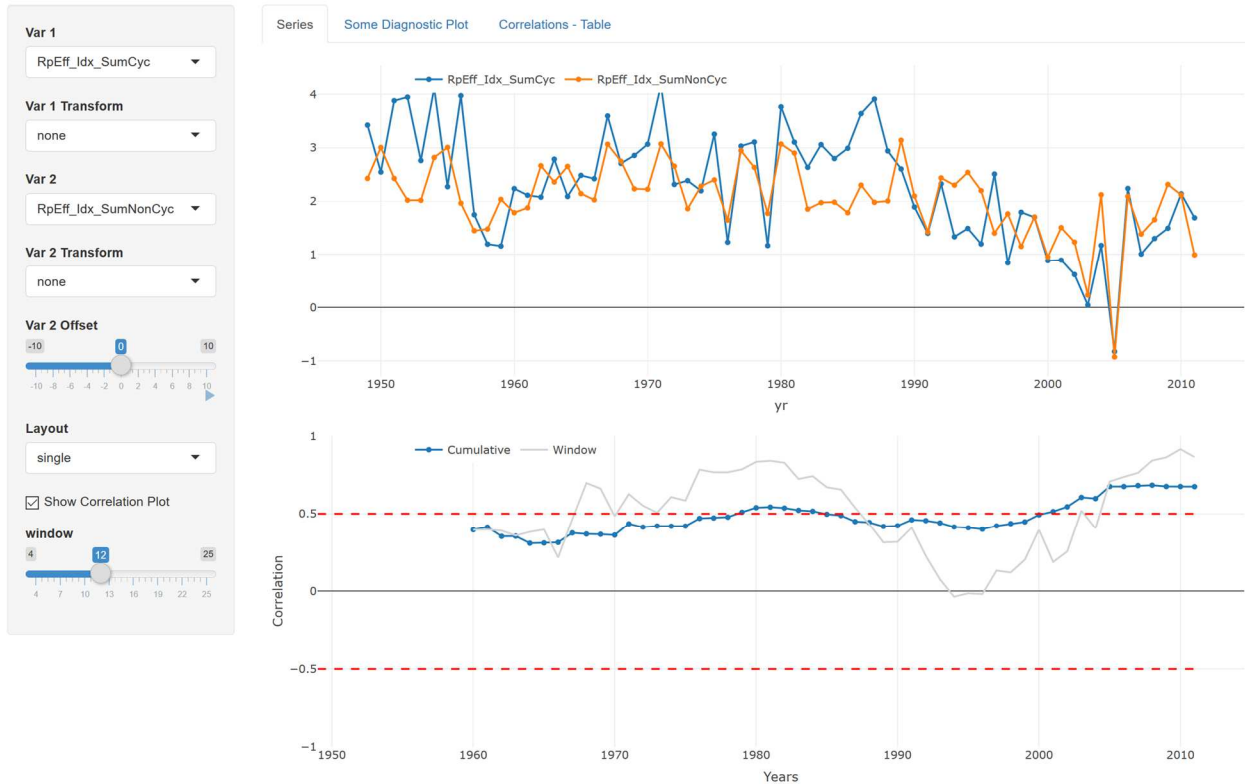


Figure 4: Explore Pairs Tab - Productivity Comparison

Figure 4 shows the two productivity indices generated in the previous step (mean log-transformed recruits per effective female for the cyclic stocks of the Summer management unit compared to the same index for several of the non-cyclic stocks). The comparison is using all the default settings, and the plot layout does not need to be adjusted.

The cumulative correlation (blue line with points) starts out weakly positive, but gradually increases as more data are included. The 3-generation sliding correlation (grey line, window length specified by menu slider) is strongly positive in the late 1970s and early 1980s (e.g. 1980 data point is the correlation over years 1968 to 1980).

The correlation weakens and disappears for time windows that end in the mid-1980s (i.e. grey line around 0), but increases rapidly for more recent data, with the strongest observed positive correlation for the 12-yr time window that ends in 2010.

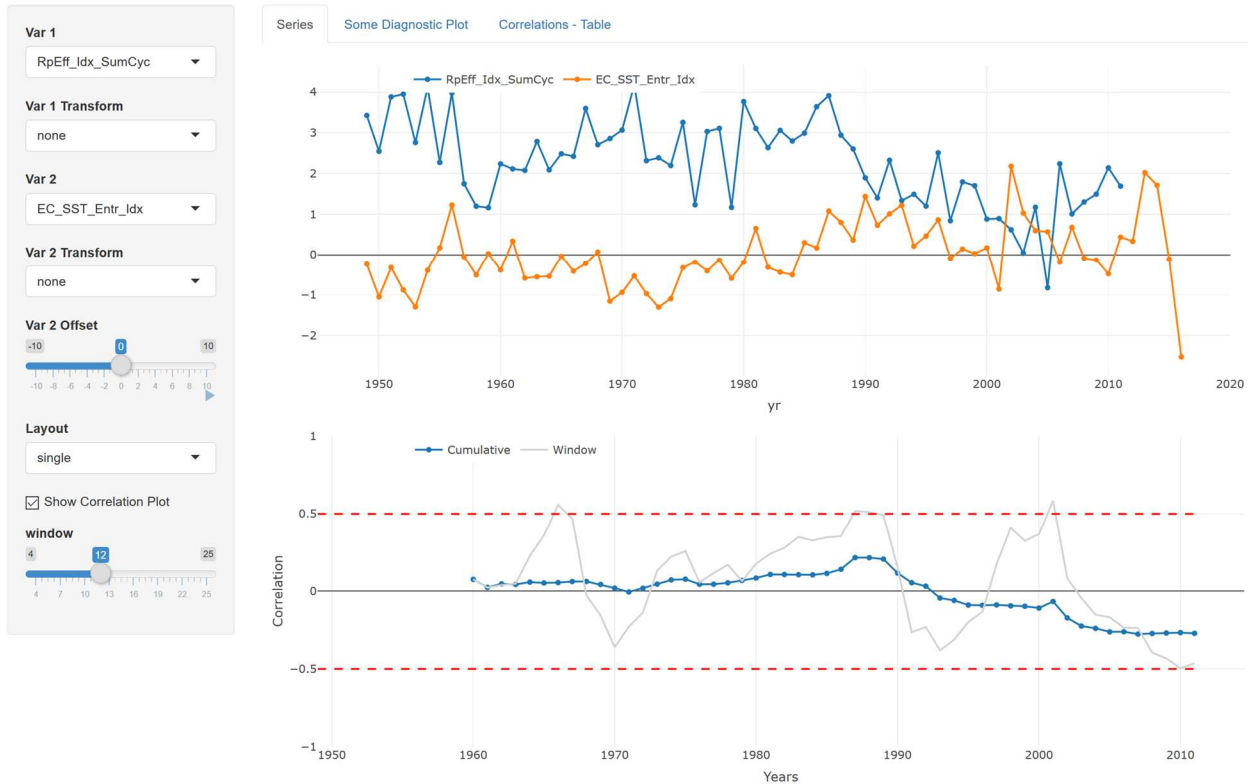


Figure 5: Explore Pairs Tab - Productivity vs. SST

Figure 5 compares one of the productivity indices to one of the environmental covariate indices. Specifically it compares the mean log-transformed recruits per effective female for the cyclic stocks of the Summer management unit to the mean normalized sea surface temperature (SST) at Entrance Island with a 2 year offset (1980 SST matched to 1978 brood year).

The cumulative correlation (blue line with points) is basically zero until the mid-1990s, and in more recent years it becomes weakly negative (i.e. lower productivity tends to be associated with higher SST). The 3-generation sliding correlation (grey line, window length specified by menu slider) switches several times between positive and negative correlations (i.e. indicating opposite directions of the effect of SST in different periods).

Note that the observation that correlations change direction over time is not biologically impossible, but rather may be due to interactions among driving factors (e.g. the effect of air temperature on the salmon productivity in a lake could be interacting with water levels). Though, as with all correlations, spurious relationships are very possible. The main point is that a plot that moves between opposite directions is not automatically wrong but should trigger a careful consideration of potential underlying biological mechanisms.

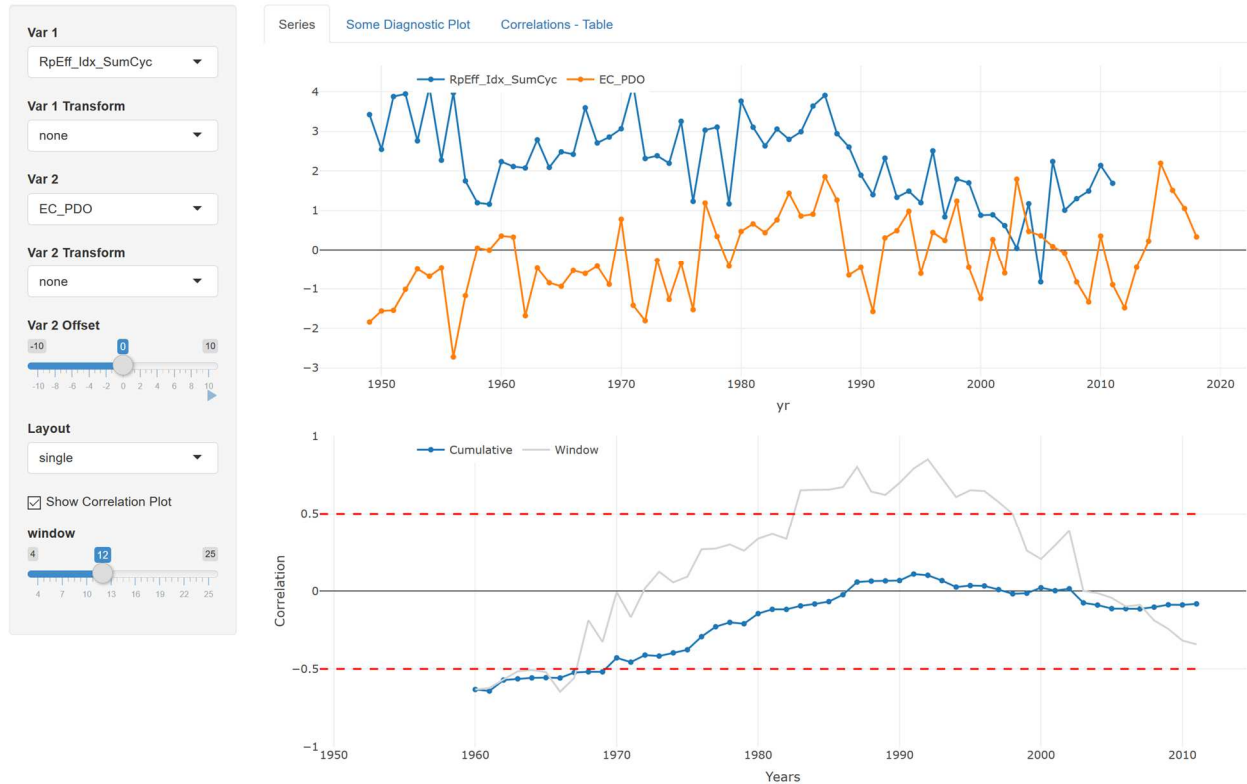


Figure 6: Explore Pairs Tab - Productivity vs. PDO - No Offset

Figure 6 shows the same productivity series as Figure 5, but compares it to a different environmental covariate. Specifically it compares the mean log-transformed recruits per effective female for the cyclic stocks of the Summer management unit to the Pacific Decadal Oscillation (PDO) index.

The cumulative correlation (blue line with points) starts out negative, but gradually weakens and approaches zero in the mid-1980s, and then basically hovers around zero. The 3-generation sliding correlation (grey line, window length specified by menu slider) shows a steep climb from negative correlations in the 1960s (i.e. stronger PDO associated with poorer productivity) to strong positive correlations in the 1980s and 1990s, before dropping again to weak negative correlations for the most recent 12-year sliding windows.

Before reading too much into this pattern, consider the inputs and settings. In this case the PDO time series has not been offset to match up to ocean entry year for the salmon stocks in the productivity index. Figure 7 shows the same two time series, just with the PDO index offset by 2 years (e.g. 2010 PDO matched to 2008 brood year).

The sliding window correlation is still weakly negative in the 1960s, but hovers around zero since then. The pattern from Figure 6 basically disappears with the offset. This means that for these stocks there is little evidence of a link between productivity for a brood year and Entrance Island SST during ocean entry year. However, the PDO could be linked to productivity in the same year indirectly (e.g. PDO years linked to precipitation and winter snowpack). The point is that the correlations can be used to check specific hypotheses, or to explore the patterns and formulate hypotheses, but cannot provide definitive answers one way or the other. There always has to be a reality check based on users' biological understanding of the data they are feeding into the app.

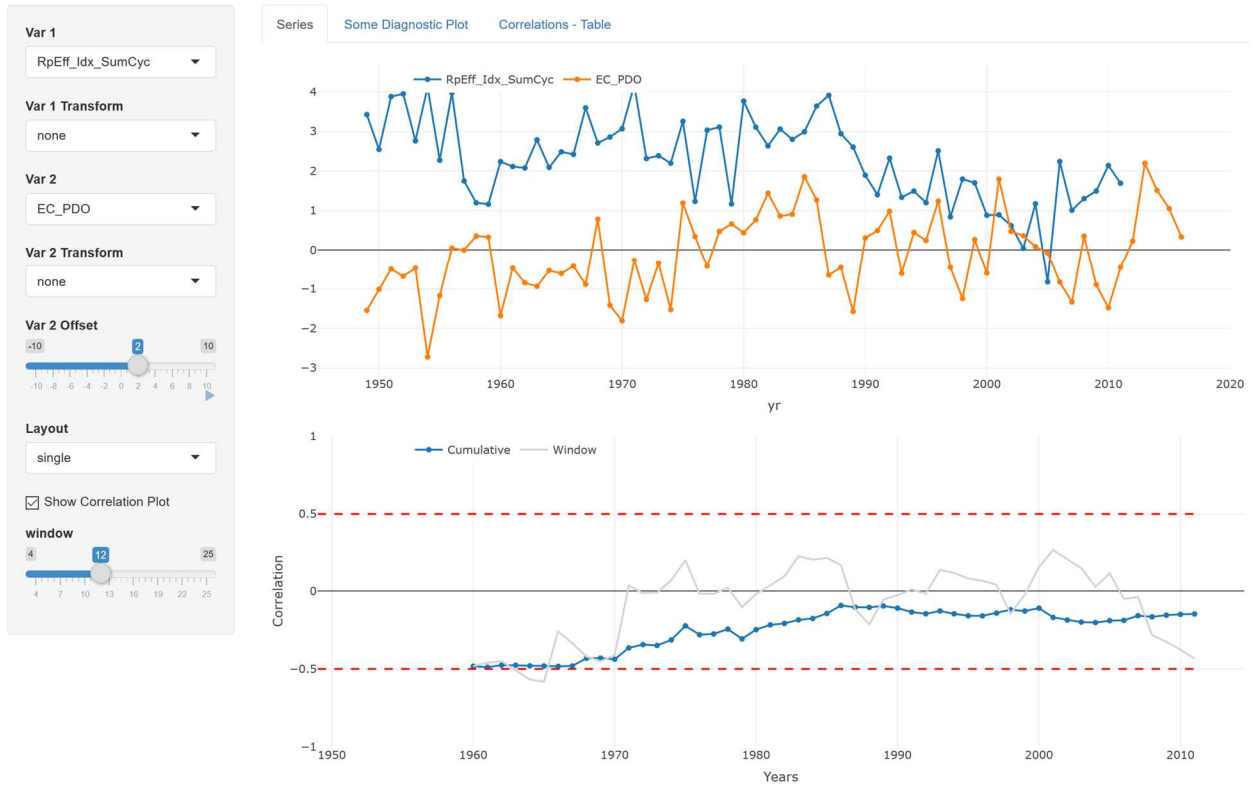


Figure 7: Explore Pairs Tab - Productivity vs. PDO - 2yr Offset

See Figure 6 for explanation.

3.4.4 Correlation Matrix

The *Correlation Matrix* tab has a main display showing a graphical summary of pairwise correlations between variables. The menu on the left provides options for alternative calculation methods, different layout, and variable selection.

The plot shows each selected variable as a row *and* as a column of a grid, and the cells of the grid contain a visual representation of the correlation.

The *Method* dropdown allows users to choose among 3 types of correlation coefficient:

- *Pearson* is the “typical” correlation calculation, and is generally appropriate for continuous variables that have a roughly normal distribution and a linear relationship
- *Spearman* is a rank-based variation that does not require normally-distributed continuous inputs. The relationship between variables is not assumed to be linear, only monotonic (i.e. strictly increasing, or strictly decreasing). This is generally less powerful in terms of finding correlations, but also requires fewer assumptions about the data.
- *Kendall* is even more robust to underlying data issues (e.g. outliers, small sample sizes).

For all three, the resulting values range from -1 (strong negative correlation) to +1 (strong positive correlation).

The *Ordering* menu currently has two options:

- *original*: variables are shown in alphabetical order
- *clustered*: variables are grouped based on hierarchical clustering. The default clustering is for $n/3$ clusters (e.g. 18 variables -> 6 clusters), but the number can be adjusted with the *Num Groups* input.

The *Plot Type* dropdown currently has 3 options. The default is *circle*, which represents the magnitude of correlation with 2 visual cues: shading and size. Stronger correlations are shown as larger, darker circles. The color indicates the direction of the correlation. Positive correlations, where both variables increase or decrease together, are blue. Negative correlations, where an increase in one variable is associated with a decrease in the other variable, are orange. The *color* option preserves the coloring, but shades the entire cells (i.e. removes the size cue). The *number* option retains the color coding, but displays numeric value of the correlation coefficient.

The *years* slider allows users to select a subset of the data (e.g. “show only the 1980s and 1990s”).

The final element on the menu is the variable selection dropdown. For now, this starts with all the numeric variables in the data set, but other options for the default are being considered based on feedback during user surveys.

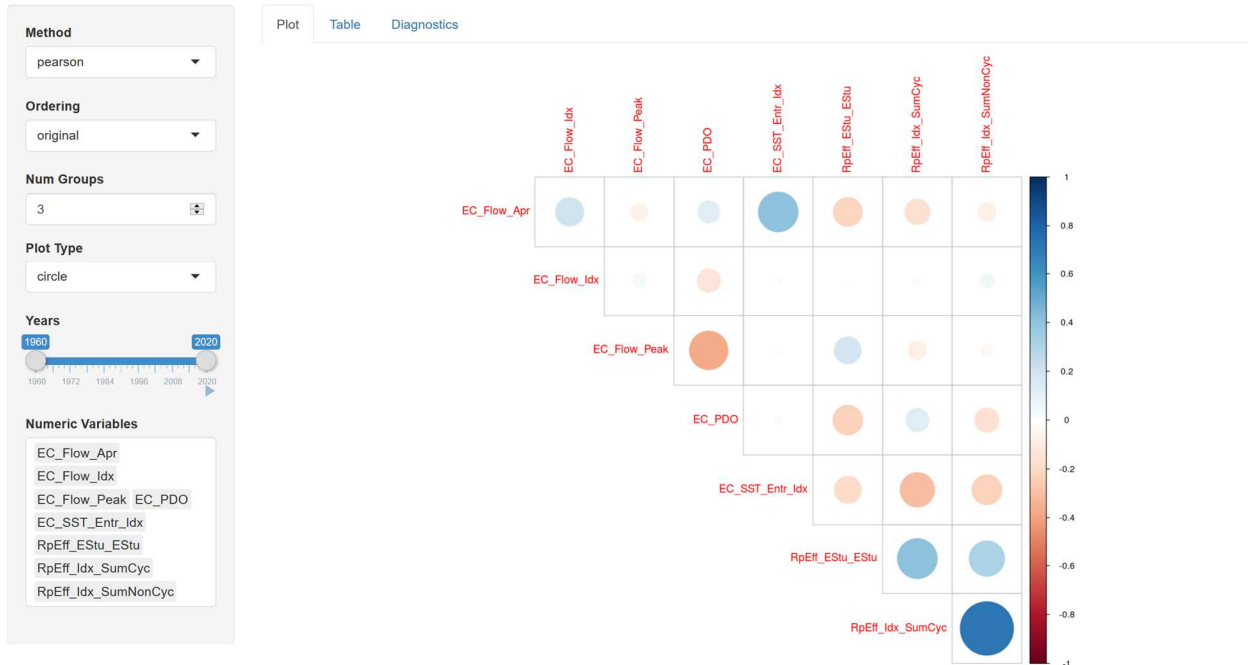


Figure 8: Correlation Matrix - Default

Figure 8 shows the default correlation matrix for 8 time series:

- 3 productivities (RpEff, recruits per effective female spawners)
- 5 environmental covariates: flow, sea surface temperature (SST), and Pacific Decadal Oscillation (PDO)

Note that the indices of environmental covariates (*EC_SST_Entr_Idx* and *EC-Flow_Idx*, generated in the previous steps) are already matched up to the appropriate brood year (i.e. 2 year offset), but that the unmodified environmental data is not (*EC_PDO, EC_Flow_Apr, EC_Flow_Peak*). For now these types of offset issues have to be handled either before the data is loaded or in the *Explore Groups* step. Future interface revisions will provide additional flexibility for offsets and transforms in the *Correlation Matrix* task.

The following correlations are identified in the matrix:

- The 3 productivity series are positively correlated with each other (3 cells on the bottom with blue circles). Of the 3 pairwise comparisons, the correlation is the strongest (i.e. darkest, largest circle) between the cyclic and non-cyclic productivity indices for Summer run stocks. Both are positively correlated with Early Stuart, but the correlation is weaker.
- There is a moderate positive correlation between river flow in April (*EC_Flow-Apr*) and the index of SST at Entrance Island, but the index series is offset by 2 years, so the relationship is likely spurious.
- There is a moderate negative correlation between peak river flow and the pacific decadal oscillation (*EC_PDO*). Both of these series have not been offset, and so the comparison is valid.



Figure 9: Correlation Matrix - Clustered

Figure 9 shows the same correlation matrix as Figure 8, except that variables are sorted into 3 groups based on hierarchical clustering. The 3 productivity series are grouped together (middle box), but the environmental covariates end up in clusters with very little shared overall pattern (little correlation).

Note the offset considerations in Figure 8.



Figure 10: Correlation Matrix - Clustered with Year Subset

Figure 10 shows the same clustered correlation matrix as Figure 9, except that the early years of data were dropped, and only the correlations since 1994 are used. Several of the correlations become more pronounced with the shorter time period, but the chances of spurious relationships also increase. The index of river flows (mean across April, May, and June, see previous section) is now grouped with the 3 productivity series, indicating a potential larger-scale functional relationship in recent years.

Note the offset considerations in Figure 8.

3.5 References

- DFO. 2018. Pre-season run size forecasts for Fraser River Sockeye (*Oncorhynchus nerka*) salmon in 2018. DFO Can. Sci. Advis. Sec. Sci. Resp. 2018/034.
- Dorner, B., Catalano, M.J., and Peterman, R.M. 2018. Spatial and temporal patterns of covariation in productivity of Chinook salmon populations of the Northeastern Pacific. *Can. J. Fish. Aquat. Sci.* 75 (7): 1082–1095. doi: 10.1139/cjfas-2017-0197.
- Grant, S.C.H. & MacDonald, B.L. 2013. Pre-season run size forecasts for Fraser River Sockeye (*Oncorhynchus nerka*) and Pink (*O. gorbuscha*) salmon in 2013. DFO Can. Sci. Advis. Sec. Res. Doc. 2012/145. vi+42p
- Mallick, M.J., and Cox, S.P. 2016. Regional-scale declines in productivity of pink and chum salmon stocks in Western North America. *PLoS One* 11(1): 1–23. doi: 10.1371/journal.pone.0146009.
- Peterman, R.M., and Dorner, B. 2012. A widespread decrease in productivity of sockeye salmon (*Oncorhynchus nerka*) populations in western North America. *Can. J. Fish. Aquat. Sci.* 69(8): 1255–1260. doi: 10.1139/F2012-063
- Peterman, R.M., Pyper, B.J., Lapointe, M.F., Adkison, M.D., and Walters, C.J. 1998. Patterns of covariation in survival rates of British Columbian and Alaskan sockeye salmon (*Oncorhynchus nerka*) stocks. *Can. J. Fish. Aquat. Sci.* 55(11): 2503–2517. doi: 10.1139/f98-179.
- Pyper, B.J., Mueter, F.J., Peterman, R.M., Blackburn, D.J., and Wood, C.C. 2001. Spatial covariation in survival rates of Northeast Pacific pink salmon (*Oncorhynchus gorbuscha*). *Can. J. Fish. Aquat. Sci.* 58(8): 1501–1515. doi:10.1139/cjfas-58-8-1501.
- Pyper, B.J., Mueter, F.J., and Peterman, R.M. 2005. Across-species comparisons of spatial scales of environmental effects on survival rates of Northeast Pacific salmon. *T. Am. Fish. Soc.* 134(1): 105–119. doi: 10.1577/T04-034.1.
- Vélez-Espino, L.A., Parken, C.K., Clemons, E.R., Peterson, R., Ryding, K., Folkes, M., and Pestal, G. 2019. ForecastR: tools to automate forecasting procedures for salmonid terminal run and escapement. Final Report submitted to the Southern Boundary Restoration and Enhancement Fund, Pacific Salmon Commission, Vancouver BC

4 Worked Example - Package

4.1 Introduction

The main purpose of the package is to make the key statistical operations and their technical documentation easily available, either for use in the interactive app, or for use in R scripts. This chapter describes the key functions and illustrates their use in an R script. The package also includes some built-in data sets.

The source code for the package is publicly available in the package repository at <https://github.com/SOLV-Code/SPATFunctions-Package>. The repository also includes a wiki with up-to-date descriptions of the evolving functions and data sets, as well as an issues page where users can report bugs, ask questions, and request additional features.

4.1.1 Functions

The correlation analysis contained in the *SPAT-Corr* app is built around 3 basic tasks (correlation matrices, pairwise comparisons, and data treatments), and implemented in 6 key functions:

Task	Function	Purpose
Correlation Matrix	<i>calcCorrMatrix()</i>	generates a correlation matrix with various options. For now it is just a wrapper function for <code>cor()</code> function from base R {stats} package. Plan is to build in alternative approaches.
Correlation Matrix	<i>plotCorrMatrix()</i>	generate a plot of a correlation matrix with various options. It is a wrapper function for the <code>corrplot</code> function from the {corrplot} package
Pairwise Comparison	<i>comPair()</i>	compute and plot correlations of 2 series (cumulative and by time window). Wrapper function for the <code>runCor()</code> function from the TTR package
Pairwise Comparison	<i>plotPair()</i>	plot 2 series (various display options)
Data Treatment	<i>shiftSeries()</i>	offset 1 or more variables. Specifically, this shifts each value in the specified column up/down by a specified number of rows. The purpose of this is to line up different variables based on assumed biological mechanisms. For example, to match the productivity from a brood year to the sea surface temperature at ocean entry 2 years later.
Data Treatment	<i>transformData()</i>	modify the values of a variable. Various transformations are available (e.g. log, z-score, percent ranks), and the suite of options is evolving. Check the help file for latest details.

4.1.2 Data Sets

The package includes a dataset of observed productivity (i.e. recruits per effective female spawner, *RpEff*) and various environmental covariates used in the 2019 Forecast for Fraser River Sockeye Salmon (Chapter 5). Note that time series are not lined up based on plausible mechanism. For example, sea surface temp may affect overall productivity during early ocean entry, but *RpEff* values are by brood year. Stock-specific offsets are required to align values properly.

Other data sets will be added to the package over time, as additional types of analyses are developed.

The code examples below illustrated how to access the data set from the R command line. Section 5 lists the variables and illustrates the data structure required for the correlation analysis functions and app.

4.2 Install & Explore

To install this package directly from GitHub, use

```
install.packages("devtools") # Install the devtools package
library(devtools) # Load the devtools package.
install_github("SOLV-Code/SPATFunctions-Package",
              dependencies = TRUE,
              build_vignettes = FALSE)
```

Once the package is installed, you can load it with

```
library(SPATFunctions)
```

Note that you don't need to re-install the package when you open the same R workspace (*.RData*) or project (*.Rproj*), but you need to *load* it with the *library()* command every time you start up a new session.

Once the package is loaded, you can explore the built in data sets and functions as follows

```
library(help="SPATFunctions" ) # see a list of functions

# check the built in data set
?SPATData_EnvCov

# check the help files for the correlation functions

?shiftSeries
?transformData

?plotPair
?comPair

?calcCorrMatrix
?plotCorrMatrix
```

4.3 Worked Example

This example uses the built-in data set of productivity and environmental covariates for Fraser River Sockeye stocks (Chapter 5). The example illustrates a workflow for initial exploration:

- calculate and plot a correlation matrix of all the variables in their original form
- transform and offset some variables, and regenerate the correlation matrix
- look more closely at one pairwise comparison

For each step in this sequence, there is a corresponding function in the package. For function details, see Section 4.1.1.

First, load the library and check the data set.

```
library(SPATFunctions)
head(SPATData_EnvCov)
```

Then, calculate and plot the correlation matrix.

```
# Calculate correlation matrix of raw data (excluding first column with years)
M <- calcCorrMatrix(SPATData_EnvCov[,-1])
head(M$cor.mat)

# plot correlation matrix
```

```
plotCorrMatrix(M$cor.mat) # original order of variables
plotCorrMatrix(M$cor.mat,order="hclust",n.groups=4) # clustered by correlation
```

Now repeat the analysis with modified data. In this example, we shift (offset) the *pdo* variable by 2 years and transform all variables to z-score (i.e. normalize). Note that this is simply an illustration, and not necessarily meaningful for all stocks.

```
data.shifted <- shiftSeries(SPATData_EnvCov, offsets=c(0,0,0,0,0,0,0,0,0,0,0,0,-
2,0,0,rep(0,18)))
data.z <- transformData(data.shifted,type="z-score",
                        cols=names(data.shifted)[names(data.shifted)!="yr"],
                        zero.convert = NA )

# compare the 3 versions
head(SPATData_EnvCov)
head(data.shifted)
head(data.z)

# repeat the correlation analysis with the new data
M.z <- calcCorrMatrix(data.z[,-1])
plotCorrMatrix(M.z$cor.mat) # original order of variables
plotCorrMatrix(M.z$cor.mat,order="clustered",n.groups=4)
```

Based on this initial exploration of all the pairwise correlations, you can then select a few variables for a closer look at how the correlation has changed over time.

```
# pairwise correlations (cumulative and by time window)
vars.test <- c("EC_jflow","EC_pdo")

plotPair(SPATData_EnvCov[,c("yr",vars.test)],layout = "single")
plotPair(data.z[,c("yr",vars.test)],layout = "2panels")

running.corr <- comPair(SPATData_EnvCov[,c("yr",vars.test)],
                        window = 12,plot.type="print")
running.corr

running.corr.z <- comPair(data.z[,c("yr",vars.test)],
                          window = 12,plot.type="print")
running.corr.z
```

5 Sample Data Set: SPATData_EnvCov

5.1 Introduction

The package includes a dataset of observed productivity (i.e. recruits per effective female spawner, RpEff) and various environmental covariates used in the 2019 Forecast for Fraser River Sockeye Salmon.

For a detailed description of the data set, refer to MacDonald, B.L., and Grant, S.C.H. 2012. Pre-season run size forecasts for Fraser River sockeye salmon (*Oncorhynchus nerka*) in 2012. Can. Sci. Advis. Sec. Res. Doc. 2012/011: v + 64 pp.

For the current version of the data set, contact Mike Hawkshaw (DFO - Annacis).

This data set is used as the built-in example for the correlation analysis functions, and the interactive app built around those functions. It also serves as a template for users who want to load in their own data sets.

5.2 Data Structure

The first column of the data set is **yr**. This is currently a requirement for the data set to work in the *SPAT-Corr* app, but not necessary when using the functions. However, when using the functions, remember to exclude the year variable before calculating the correlation matrix as per the examples in Chapter 4.

The remaining columns can have any labels that work with R data frames (i.e. no special characters like "&" or "/", but "_" works fine). However, it makes working in the app easier if the column labels have a hierarchical system, because the dropdown menus in the app will sort the variables alphabetically. For example, the variables *EC_Flow_Apr*, *EC_Flow_May*, and *EC_Flow_Jun* will show up together in the menus, but *aprflow*, *mayflow*, and *junflow* will be spread among other variables and harder to find and select together. Note that this is not a requirement, but a recommendation for streamlining your workflow.

The variable labels in the sample data set consist of nested identifiers. The first chunk identifies the type of variable as either an environmental covariate (EC) or a stock productivity (RpEff). For environmental covariates, the second chunk specifies the type of variable (e.g. sea surface temperature, flow) and the remaining chunks identify the specific series. For example "*EC_SST_Pine_May*" is an environmental covariate of sea surface temperature at Pine Island in May. For the stock productivity variables, the first chunk specifies the type of variable (for now they are all RpEff), the second chunk identifies the management group (e.g. ESum, Sum), and the third chunk specifies the stock. For example, *RpEff_Sum_LStu* is the productivity in terms of recruits per effective female spawner for Late Stuart in the Summer management unit.

5.3 Variables

5.3.1 Environmental Covariates

Variable	Description
EC_Flow_Apr	Fraser River April flow
EC_Flow_May	Fraser River May flow
EC_Flow_Jun	Fraser River June flow
EC_Flow_Peak	Fraser River Peak discharge
EC_SST_Entr_Apr	April Sea Surface Temp at Entrance Island
EC_SST_Entr_May	May Sea Surface Temp at Entrance Island
EC_SST_Entr_Jun	June Sea Surface Temp at Entrance Island
EC_SST_Pine_Apr	April Sea Surface Temp at Pine Island
EC_SST_Pine_May	May Sea Surface Temp at Pine Island
EC_SST_Pine_Jun	June Sea Surface Temp at Pine Island
EC_SST_Pine_Jul	July Sea Surface Temp at Pine Island
EC_PDO	Pacific Decadal Oscillation

5.3.2 Fraser Sockeye Productivity Variables

Variable	Description
RpEff_EStu_EStu	Observed productivity of Early Stuart sockeye (Recruits/Effective Female)
RpEff_ESum_Bowr	Observed productivity of Bowron sockeye (Recruits/Effective Female)
RpEff_ESum_Fenn	Observed productivity of Fennel sockeye (Recruits/Effective Female)
RpEff_ESum_Gate	Observed productivity of Gates sockeye (Recruits/Effective Female)
RpEff_ESum_Nadi	Observed productivity of Nadina sockeye (Recruits/Effective Female)
RpEff_ESum_Scot	Observed productivity of Scotch sockeye (Recruits/Effective Female)
RpEff_ESum_Sey	Observed productivity of Seymour sockeye (Recruits/Effective Female)
RpEff_ESum_UPit	Observed productivity of Upper Pitt sockeye (Recruits/Effective Female)
RpEff_Sum_Chil	Observed productivity of Chilko sockeye (Recruits/Effective Female)
RpEff_Sum_Harr	Observed productivity of Harrison sockeye (Recruits/Effective Female)
RpEff_Sum_LStu	Observed productivity of Late Stuart sockeye (Recruits/Effective Female)
RpEff_Sum_Ques	Observed productivity of Quesnel sockeye (Recruits/Effective Female)
RpEff_Sum_Raft	Observed productivity of Raft sockeye (Recruits/Effective Female)
RpEff_Sum_Stell	Observed productivity of Stellako sockeye (Recruits/Effective Female)
RpEff_Lat_Birk	Observed productivity of Birkenhead sockeye (Recruits/Effective Female)
RpEff_Lat_LShu	Observed productivity of Late Shuswap sockeye (Recruits/Effective Female)
pEff_Lat_Port	Observed productivity of Portage sockeye (Recruits/Effective Female)
RpEff_Lat_Weav	Observed productivity of Weaver sockeye (Recruits/Effective Female)

6 Feature Requests

This chapter summarizes feedback and requests from interview participants during the user surveys described in Chapter 2.

6.1.1 Workflow

Participants provided many suggestions related to the workflow within the *SPAT-Corr* app, and suggested extensions that would make the app fit more seamlessly into their overall workflow.

High-priority features for next phase of development:

- generate a report with settings and results
- option to re-set tabs to start from scratch in each tab

Features to consider for longer-term development:

- generate output file with the R code that implemented the analysis, including all the input arguments. The idea behind this is the experienced R users could explore the data with the app, then save the code, and build their own more detailed analyses without having to re-do the preparatory steps. This is conceptually straight forward, given that each step in the app is linked to a stand-alone function from the *SPAT R Package*. However, there are practical challenges around setting up a text template where the specific user-selected settings are filled in. The first step will be some feasibility testing of a basic example.
- include a notebook option for collaborative analyses (e.g. Jupyter, RStudio). This could substantially increase future adoption of the app, but likely presents significant programming challenges. More scoping discussion would be required to identify a feasible approach for this.
- offer the ability to link observed correlations to functional relationships (e.g. forecast with environmental covariate). There is no plan to build forecasting capability into this app, but it may be possible to set up an output option that generates data files for importing into the ForecastR app.

6.1.2 Load Data

Participants requested the following data-related features:

- ability to sort data so they can verify it
- ability to filter out NA values
- ability to sort, using pivot tables, removing particular data
- ability to load more than one file

Careful scoping will be required to build some data preparation into the app without making it too complex in terms of the underlying data handling and computations. Note that the app is specifically designed for time series data with "yr" as the first column, and all other variables in additional columns. For now it allows for easy transformation of individual time series and subsetting of time periods. Filtering out NA values is not necessary, but options for how NA values are handled in different parts of the analysis can be built in (e.g. when calculating aggregate indices in the *Explore Groups* task). More complex data clean-up and handling has to be done *before* loading into the app.

We considered more flexibility for data handling during the initial design discussions, when we envisioned a single app for different types of data and different analyses. However, when we started building prototypes we decided to split development into clearly distinct stand-alone apps, each specifically designed for a particular type of data and analysis. As part of the overall *SPAT* family of tools we have identified a conceptual structure for a data handling app that would include the above features, but this is only at the early design stage. The first step in the development is to develop a suite of data conventions that are flexible enough to handle most common types of fishery data, yet

are internally consistent and allow for automated conversions. The current draft of the data conventions is available at <https://github.com/SOLV-Code/SPATFunctions-Package/wiki/1-Data-Conventions>.

A data filtering step should be added on the data loading tab (e.g. exclude some variables or years from all the subsequent steps).

6.1.3 Explore Groups Task

- trouble clearing screen and re-loading; users had to go back and forth between the data tab and explore tab to reset
- clarify how to reset the plot view
- need to clarify offsets and consider limiting the range
- option to plot series in individual panels with zoomed scales
- option to create an aggregated index using weightings (i.e. weighted by another variable)
- additional options for the aggregate index (difference, range)
- option to calculate aggregate index even when 1 or more component series have NA

6.1.4 Explore Pairs Task

- need to clarify alternative layouts, including secondary axis
- move/emphasize option for show/hide retrospective correlation plot
- add option to switch between the retrospective correlation plot and either:
 - a scatter plot with the option for showing a linear fit.
 - 2 boxplots side by side, with the option of showing a violin plot

6.1.5 Correlation Matrix Task

Participants had many questions regarding the handling of offsets and data transformations for individual variables. For now this can be done by generating custom time series with specific offsets on the *Explore Groups* tab, and then selecting those constructed series in the subsequent tasks. We are planning to add more flexibility for this on the *Correlation Matrix* tab, but are still testing alternative implementation options.

In addition, other types of correlation matrix plots should be considered (e.g. pairs plot showing all the scatterplots with linear fits, or all the boxplots. The diagonal could show sparklines or histograms).

6.1.6 New Task: Explore Individual Series

When exploring time series, it would be useful to subset the data e.g. before and after a certain year when there is a shift in the time series, exploring odd and even years, exploring cycle line data, etc. For each of these subset time series, it would be useful to be able to calculate the quartiles and to plot the sub-series as boxplots to be able to compare them.

We will consider either including these features in the *SPAT-Corr* app, or building a stand-alone data exploration app.

6.1.7 Other

- tab with summary statistics for the input data
- display the number of observations in each time series (e.g. put the $n = ?$ in the legend behind the name of the time series)