UNIVERSITÉ PARIS-SUD 11
FACULTÉ DES SCIENCES D'ORSAY

N° d'ordre: 2

MASTERS DISSERTATION

Presented to obtain

THE DEGREE OF MASTERS 2 IN
GEOSCIENCES AT THE UNIVERSITY
PARIS-SACLAY XI

Domaine: Hydrogeology, Hydrology and Soils

by

Jeffrey NORVILLE

Development of a Verification Scoreboard Utility for Hydrological Forecasts

Presented 18/09/2016 before the Committee:

Mme. Dr.	Véronique DURAND	
Mme. Dr.	Christelle MARLIN	
Mme. Dr.	Maria-Helena RAMOS	(Supervisor at IRSTEA)
M. Dr.	Guillaume THIREL	(Co-supervisor at IRSTEA)

Dissertation prepared at
IRSTEA - Centre d'Antony
UR HBAN Equipe Hydrologie
1, rue Pierre Gilles de Gennes CS 10030
92 160 Antony
France

Abstract

This thesis presents the prototype of a verification scoreboard developed for the evaluation of hydrological forecasts as well as its design considerations and the workflow undertaken to achieve it.

Based on a review of verification scores for hydrological forecasts and scoreboard interfaces, I created a utility that combines a score database and a web-based display to facilitate visualization and interpretation of forecast quality. Initial tests on reforecast data are presented to illustrate the way scores are displayed by the utility developed.

Forecast verification is the process of assessing the quality of a forecast, and the performance of a forecast system is measured by the quality of the forecasts it delivers. 'Quality' measures how well a forecast does compared against a corresponding observation of what actually occurred, or some good estimate of the true outcome.

In practical terms, hydrological forecast verification is also called "forecast evaluation". In meteorology, the "value of a prediction" specifically refers to how a forecast helps the user make better decisions. The quality of a forecast can be assessed through a wide range of metrics, or verification scores. The verification scoreboard utility is important, as it supports comparisons of performance of different forecasting systems.

The aim of this report is to describe the development of a scoreboard to satisfy the needs of users from the European research project IMPREX (www.imprex.eu), of which IRSTEA is a partner. The tools involved in creating the scoreboard are also addressed in this report, as the project continues outside of the scope of my thesis and a considerable effort was made to build a sustainable, maintainable, and potentially open source product.

Keywords : Forecast verification, Seasonal forecasts, Hydrology, Meteorology, R, Shiny, postgresql.

Développement d'un tableau de bord pour l'évaluation de prévisions hydrologiques

Résumé

Cette thèse présente le prototype d'un *Verification Scoreboard* développé pour l'évaluation de prévisions hydrologiques, ainsi que les considérations de conception et le travail entrepris pour y parvenir.

Sur la base d'une révision bibliographique scores de vérification utilisés pour les prévisions hydrologiques et des interfaces de *scoreboards existents*, je créé un utilitaire qui combine une base de données de scores et une interface pour faciliter la visualisation et l'interprétation de la qualité des prévisions. Des essais initiaux sur des données de prévisions sont présentés pour illustrer la façon dont les scores sont affichées par l'utilitaire développée.

La vérification des prévisions est le processus d'évaluation de la qualité des prévisions, et la performance d'un système de prévision est mesurée par la qualité de la prévision émise. La "qualité" des prévisions est définie par comparaison à une observation correspondante, ou une bonne estimation de celli-ci.

En termes pratiques, la vérification des prévisions hydrologiques est également appelé "évaluation des prévisions". En météorologie, la "valeur d'une prévision" se réfère spécifiquement à la façon dont une prévision permet à l'utilisateur de prendre de meilleures décisions. La qualité d'une prévision peut être évaluée à l'aide d'un grand nombre de critères numériques (ou de scores de vérification). Dans le but de comparer la performance de différents systèmes de prévision, il est donc important de disposer d'un scoreboard.

Le but de ce rapport est de décrire le développement d'un scoreboard, entrepris pour satisfaire les utilisateurs du projet de recherche Européen IMPREX (www.imprex.eu), duquel IRSTEA est partenaire.

Les outils nécessaires à la création du *scoreboard* sont également discutés dans ce rapport, étant donné que le projet continue au délà de ma thèse et un effort considérable a été fait pour construire un produit durable, facile à maintenir, et dont le code source pourra être mis à disposition de la communauté.

Mots-clefs : Météorologie, prévisions saisonnières, R, Shiny, postgresql.

Acknowledgments

This thesis has it's origins in the support of many people. My mentors, peers and friends at IRSTEA – specifically my advisors, Maria-Helena Ramos and Guillaume Thirel, as well as the whole HYDRO team managed by Charles Perrin; R guru Olivier Delaigue; Louise Crochemore, whose data got this thing rolling; and "les autres stagiaires" who helped push along my French, learn a little R, and remember those long-expired uni ultimate frisbee skills.

The team at ECMWF was also very supportive, and I have Louise Arnal to thank for the data received from the EFAS system. Florian Pappenberger and Paul Smith helped conceive the scoreboard and particularly the PostgreSQL database, and there exists a fantastic opportunity to tie their SOAP web services with the scoreboard. Unfortunately the SOS database wasn't online in time to be used for this project.

SMHI and Ilias Pechlivanidis supplied the majority of the scores we're using in the current testing database. Thank you Ilias!

My participation in this Master's program is based on my French studies which began in 2015. U-PSUD FLE (français langue étrangère) professors Laurence Peyraud, Roselyn Debrick, and Annie Reavley were particularly supportive, challenging, and motivating. Though my personal expectations have remained relatively low, theirs were high, which pushed me to progress in a short year.

Hydrogeology professors Véronique Durand and Christelle Marlin also have my eternal thanks; Véronique was the first person to meet with me, introduced me to the FLE program, and encourage me down this path.

Finally, it's my wife Elena, and our boys Oliver and Sebastian, who I've asked the most of during my continuing education and graduate project, and without their support and understanding I simply wouldn't be where I am.

Contents

Glossary

EFAS European Flood Awareness System (EFAS), developed and tested at the Joint Research Centre (JRC) of the European Commission, in collaboration with national hydrological and meteorological services, European civil protection agencies through the Emergency Response and Coordination Centre (ERCC), and other research institutes. It provides pan-European overview maps of flood probabilities up to 15 days in advance, and detailed forecasts at stations where the national services are providing real-time data. More than 30 hydrological services and civil protection services in Europe are part of the EFAS network. From: `http://www.ecmwf.int/en/research/projects/efas`. 9, 11

Forecast period The validity period of a forecast. For example, long-range forecasts may be valid for a 90-day period or a season. 8

Forecast quality How well a forecast compares against a corresponding observation of what actually occurred, or some good estimate of the true outcome. 8

Forecast skill The relative accuracy of the forecast over some reference forecast. Score: a quantitative measure of forecast quality. 9, 10, 15

Forecast value How a forecast helps the user to make a better decision. 9

ggplot2 ggplot2 is an R package for data visualization. Created by Hadley Wickham in 2005, ggplot2 is an implementation of Leland Wilkinson's *Grammar of Graphics* —- a general scheme for data visualization which breaks up graphs into semantic components such as scales and layers. One of the strengths of ggplot2 is how well it shares it's "grammer" with Wickam's other popular R libraries, notably **dplyr**. A weakness: its implementation of grammar costs in performance, and it's slightly slower than lattice or base plot. Regardless, since 2005 ggplot2 has grown in use to become one of the most popular R packages. 38

Hindcast A forecast which is run over a historical period (often 30 or more years) with available observation data; provides verification statistics for forecast tools. 8

IDE Integrated Development Environment, a piece of software that facilitates interaction with a programming language. According to (**wiki:ide**), "An IDE normally consists of a source code editor, build automation tools and a debugger". 21

IMPREX **IM**proving **PR**edictions and management of hydrological **EX**tremes. Funding received from the European Union Horizon 2020 Research and Innovation Program under Grant Agreement N° 641811. (`http://www.imprex.eu`). 9

LaTeX LaTeX is a document preparation system used for the communication and publication of scientific documents. LaTeX is free software and is distributed under the LaTeX Project Public License. https://www.latex-project.org//. 21

Lead time The period of time between the issue time of the forecast and the beginning of the forecast validity period. Long-range forecasts based on all data up to the beginning of the forecast validity period are said to be of lead zero. The period of

time between the issue time and the beginning of the validity period will categorize the lead. For example, a Winter seasonal forecast issued at the end of the preceding Summer season is said to be of one season lead. A seasonal forecast issued one month before the beginning of the validity period is said to be of one month lead. Also, Forecast lead time. 8

Scoreboard User Any person (partner of the IMPREX project or not) who wishes to visualize the quality of a forecast or a forecast system investigated in one of the case-studies of the IMPREX project (under the condition that the score of this forecast or a forecast system is made available by a score data provider). 9, 10, 40

Scoreboard Utility A graphical interface, connected to a score database, to support comparisons between numerical scores of different forecasts or forecast systems. Score data provider: a partner of the IMPREX project who contributes data to the scoreboard. 10

Source Control Also "version control" or "revision control", this is a method of controlling of changes to computer programs or documents. According to (**wiki:source.control**), "Revision control manages changes to a set of data over time." In practice it allows one user to look at their prior work; it allows teams to see one anothers' edits; and it provides a way to "turn back the development clock" to former, working versions in event of catastrophic changes (ex by an inexperience programmer). 28

System 4 EFAS System 4 refers to the "latest" ECMWF Integrated Forecasting System (IFS), implemented November 1 2011. The System 4 system runs on an improved atmospheric model, more (51) member forecasts, a new ocean component (NEMO), and uses initial conditions defined by NRT NEMOVAR. 43

Part I

Introduction and Some Terms

1 General content and overview of the study

In the field of a meteorology, possibly the most famous example of the need for verification is the 1884 Finley report on his tornado forecasts (Finley, 1884). J.P. Finley was a Sergeant in the US Army Signal Corps, and he published results of several months of tornado observations and 12-hour predictions. The original article was not reviewed for this report, but is partially-reproduced and thoroughly documented in "The Finley Affair: A Signal Event in the History of Forecast Verification" (Murphy, 1996) and by the CAWCR (Joint Working Group on Forecast Verification Research, 2016).

According to Murphy, his predictions for the rare event (tornado/no tornado) were "accurate" over 95% of the time (see Table 1, below). He had predicted 28 tornadoes correctly in 2803 forecasts. Within months of his claims, a paper by G.K. Gilbert pointed out a serious problem with Finley's math: a forecast of 0 tornadoes in 2803 events would have increased his "accuracy" to 98.2%. Several other authors joined in the conversation, and more of the resulting verification conversation is continued under Section 2, including an introduction to numerical skill scores.

Forecasting science exists across many disciplines: hydrology and hydrogeology in my direct experience, meteorology, of course, as well as in the financial markets, politics, and in retail systems. If a person can collect historical data and predict a future outcome by organizing it (modeling) in some way, they have made a forecast, or prediction.

Terminology differs across discipline. Lead time, for instance, can be a confusing topic, yet is fundamental to meteorological forecasts. To borrow examples of "lead time" from other fields, journalists define the period of time between getting a story and submitting it as their lead time; supply-chain manufacturers define lead time as the length of time between a cheeseburger or iPhone being ordered and the minute it's ready for delivery (excluding inventory), literally a measure of time it takes raw materials to turn into a burger or phone.

Lead time in the context of meteorological and hydrologic forecasting, thus in the context of this report, means the distance into the future we are forecasting. We may issue a forecast looking ahead one week or several months, and another forecast will be issued in several days' time. Typically the quality, or "goodness" ((Murphy, 1993)), of a forecast decreases as lead time increases.

Forecast period as a general term defines how long a forecast is designed to be valid. In this application, however, it specifies the time period during which we have pairs of forecasts and observations to apply verification.

A reforecast or Hindcast (sometimes used interchangeably) is a forecast run over some historical period, when the forecast scientist can score their model by paring forecasts with recorded observations. In the case of a hindcast this is typically a long period of record, several months or years.

Consistency is a judgment-based value defined as the relationship between a forecaster's judgment and their forecasts.

A very specific term, Forecast quality is a measure of the fit between forecasts and matching observations.

Forecast value is a measure of the economic benefit of the use of a forecast. If damage or other expenses were avoided by use of the forecast it can be given a numerical value.

These three terms – consistency, quality and value – comprise the "goodness" of a forecast according to Murphy (Jolliffe and Stephenson, 2012), (Murphy, 1993).

Forecast skill is the quantification of the quality of a forecast, usually compared to a reference forecast (Jolliffe and Stephenson, 2012). See Section 2 and Table 1 for examples of calculating the skill of a forecast.

A forecast stakeholder, potentially a Scoreboard User, is anticipated to be a person or entity interested in the use or outcome of a prediction. Stakeholders have different needs and terminology, not only across different disciplines but even within the same discipline, for a different application.

An example of forecasts emphasizing different forecast variables comes from the northern latitudes. Imagine two stakeholders interested in winter temperature forecasts: a power company representative might keep close watch on temperature fluctuations to anticipate changes in electricity demand; however the highway and roadway maintenance sector might only care about which areas will drop below some threshold when their road deicing begins. A nearby airport manager may want a similar temperature-based forecast but with different threshold values and more information on atmospheric profiles. The Forecast value will vary for each of the three user profiles.

1.1 Investigation Questions and Overview of Concept

Across Europe the IMPREX project (IMproving PRedictions and management of hydrological EXtremes) seeks to improve the forecast of extreme hydrological events, namely floods and droughts. Team members include ECMWF's EFAS team; EFAS (European Flood Awareness System) utilizes ensemble forecasts from ECMWF, among others, and ECMWF runs the EFAS Computational center, which runs all that concerns EFAS data processing and models. ECMWF also hosts the web architecture of the EFAS Information System and and supports other EFAS centers who distribute flood warnings, and verify EFAS forecasts.

Communicating forecast verification data and scores is the responsibility of an operational forecasting center. It allows forecast developers to be transparent on the quality of their forecasts and users to gain confidence on the forecasts they are using when making decisions. There are several ways to communicate verification statistics. Commonly, forecast centers will share, via their websites, operational or research activity charts, graphs, numerical tables or summarized information on the quality of past forecasts, or the results of verification studies targeting specific important events. In general, forecast verification displays are accompanied by instructions on how scores are computed and should be interpreted.

Example scoreboards or cards in use to day are introduced in Section 7 (page 21).

Verification scoreboards are commonly used within a forecasting organization to validate their models and help tune future development efforts.

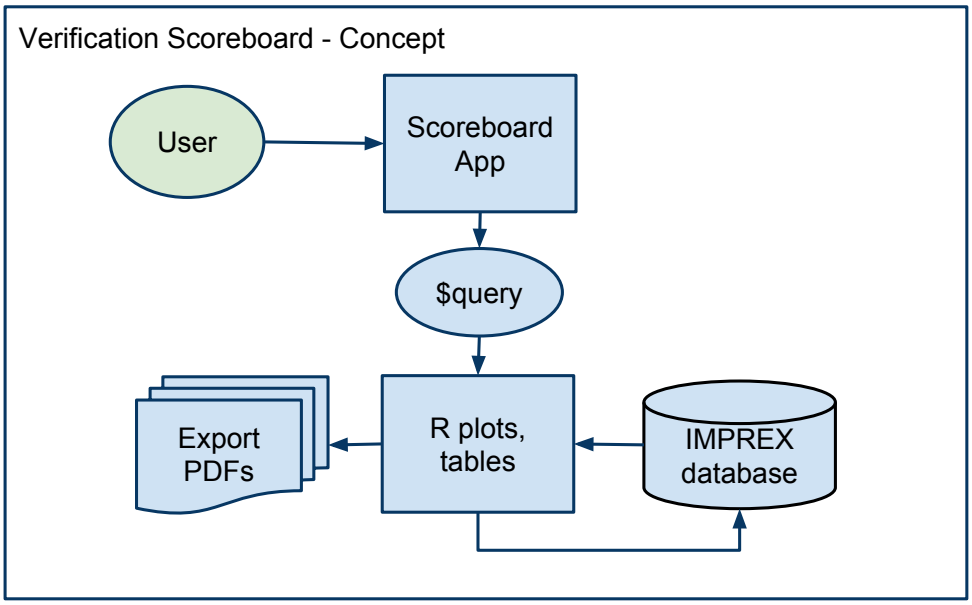

Figure 1 – Concept of a scoreboard.

1.2 Investigation questions

This Masters dissertation project, and our contribution to the IMPREX project, seeks to create a shared back-end database with an open, customizable graphical user interface (GUI) to compare scores between forecast systems.

1.3 Standardizing data input and display

Figure 1 shows a schematic view of the concept of scoreboard that we adopted in this study. A user opens the Scoreboard and start a query, which will allow the scoreboard to display plots and tables based on the score database maintained by the partners of the IMPREX project. All results may be exported as portable document format (PDF) files.

Scoreboard User is the prospective consumer of the GUI. They interact with the visual component of the Scoreboard, the display; they also interact with the more complex component is the "plumbing", or interconnections, underneath.

In order for the scoreboard to display useful and interesting comparison data, users may upload their score data and explore individual or group plots, as well as compare against data other users have loaded. By definition, then, this shared database will be both receiving and displaying data, including standardizing data objects (model identifiers, time and date variables, etc) across models.

Forecast skill is measured and displayed by the Scoreboard Utility.

The main challenges are:

- Comparing comparable scores,

- Comparing comparable locations,
- Avoid misleading users.

In order to create a usable scoreboard we have created a web-based scoreboard which connects to a centralized database. To facilitate the introduction of this utility we have released the scoreboard as a stand-alone tool, which connects either to a locally-managed database or a series of flat files.

There is also the opportunity to install the Scoreboard on a web server, install the database, and connect them to make a high-performance web-based scoreboard. As the data we've used is considered "provisional", and appropriate servers have not yet been identified, this part of the concept is however not included in the work reported here.

1.4 Some Limitations and Exceptions

Forecast verification is a broad field. Over the course of the last few months I touched a tiny part of it, and here I enumerate some aspects of verification I omitted from this study:

1. Spatial visualization of point (coordinate) data in Shiny: automatic data load for coordinates is hard (model systems may use custom coordinates, ex EFAS) and layering in Shiny leaflet library slows the interface,
2. Spatial correlation by shapefile (eg from a GIS): while R can do it, the PostgreSQL requires from extensions (PostGIS) which add complexity; also "expensive" to store shapefiles,
3. We use score data calculated / loaded by IMPREX team members. Creating scores from raw paired prediction / observations (as is done in some verification systems) becomes too complex for this project's scope.

Part II

About Forecasts

2 Verifying Hydrometeorological Forecasts, an Overview

Hydrological and meteorological forecast systems may be verified, checked for performance and quality, by running them on historical parameters – where the answer is already defined. This is known as a hindcast or reforecast. The forecast systems can be run as if "live", that is to say using prior information only, to issue forecasts into the future. A medium-range system producing daily forecasts over the next month might have lead times from 0 (today) to 30.

Resulting datasets are comprised of predictions made on a date and extending to the forecast validity period. These are paired with observations made during the same time period.

The simplest verification might be to plot the two datasets on a timeline and give an eyeball-approximation of whether or not the forecast predicts the observations.

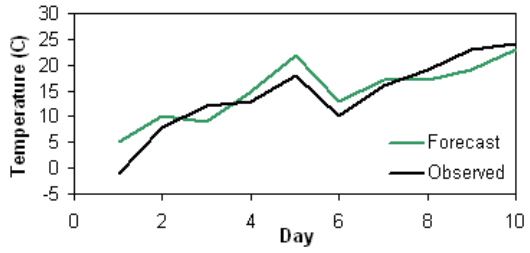

Figure 2 – Timeseries "Eyeball" Verification

Another type of visual verification could be performed on a map, the birds-eye view of a forecasted pattern with the observed weather at the time.

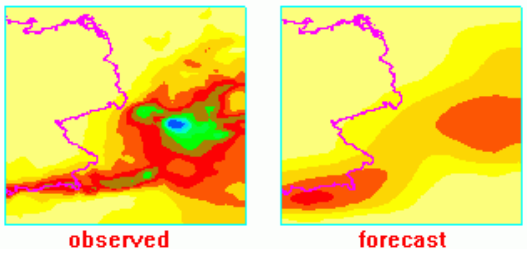

Figure 3 – Spatial "Eyeball" Verification

A more objective comparison may be made using numerical verification; this is simply a calculation done on the difference between the values, which typically degrade as lead time values increase.

3 Motivation to Verify Forecasts

Why verify a forecast? Numerical verification addresses administrative, scientific, and economic drivers. Before we get into that list, let me present a simple arithmetic ques-

tion:

A bat and a ball cost €1.10. The bat costs €1 more than the ball. How much does the ball cost?

This example (borrowed from Kahneman (2011)) is often rapidly, intuitively answered as "ten cents", which is wrong - a simple check shows the trap: if the ball cost ten cents, the bat would be a euro more - €1.10 - giving a total of €1.20.

Intuitively, humans substitute an easier problem for a harder problem. It takes just enough effort to check our conclusion, to evaluate our "gut" response, that we often just don't do it.

Numerical verification – checking the actual value, trends in scores – is the kind of thing computers are particularly good at, and one reason we write so much software. The next sections of this thesis will address:

1. What is a verification score?
2. Which ones work, and why?
3. Different forecasts, different scores
4. Building our scoreboard

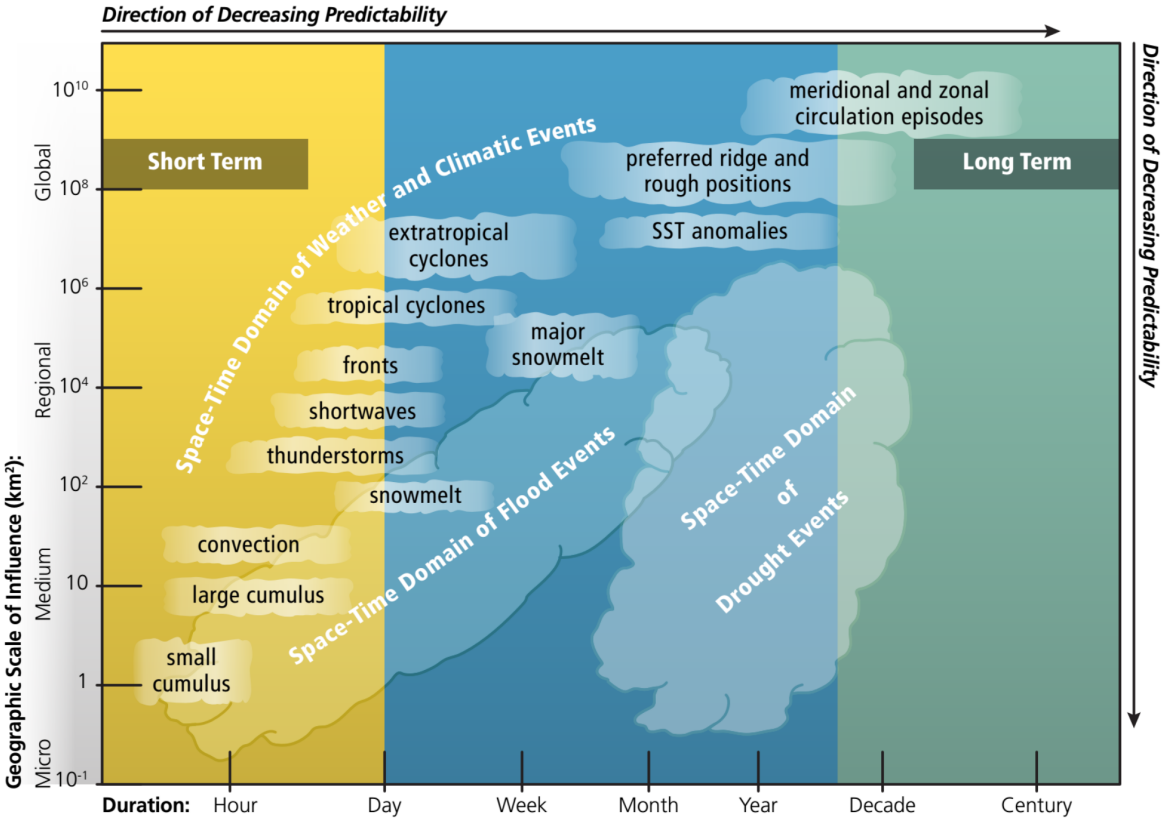

Figure 4 – Predictability and Forecast Event Type (SST: sea surface temperature) (Jones, 2015)

There are complicating factors in modeling and forecasting the physics of our planet. As Figure 4 qualitatively illustrates, our ability to predict phenomena degrades with increasing scale (moving down the left axis), and moving from short term to long term forecasts (left to right). What is striking in this figure is the relative notion of meteorological events; flood, and drought events. The events that affect our societies and businesses aren't in the "easily-predicted" range.

3.1 Administrative motivation

Public agencies typically track their performance since they may be asked to justify funding requests, and it helps to quantify their gains over "competitors".

In "Forecast Verification" M. Jolliffe points out the UK Met Office offers "bounties" or rewards based on model performance quantified with internal scoreboard results. In practice, this can result in forecaster hedging. (Jolliffe and Stephenson, 2012).

A Forecast skill is, as noted in the introduction, the quantitative identifier of prediction quality.

3.2 Scientific motivation

Finding errors or ill-conditioned components in a model, promoting or changing post-processing values, and/or noting when a change in a complex system had unintended influences are a few reasons scientists measure their model performance.

For Finley, he was simply overconfident in the value of his predictions; an objective verification process, or skill score comparison, would have probably highlighted to the interested observer that his predictive model wasn't worth promoting.

Today, national weather centers share the results of their systems together; try to make comparisons of skill extended over different lead times; but it's difficult to agree on and to define a baseline for comparison.

3.3 Economic motivation

Different models may be conditioned for different answers. Consider the fictitious "northern latitudes" example from the introduction: the power company, interested in winter temperature fluctuations, while the highway sector waits on predictions of minimum temperature.

Models considered for use by the power company should be "scored" on their skill in predicting temperature fluctuations, or how well observations followed the model. On the other hand, the roadway authority would select the model with higher skill reported on detecting a specific temperature threshold.

In other sectors, a reservoir manager might be more interested in forecasts helping her understand peak demand for water usage from the reservoir. Disaster preparedness representatives for a downstream municipality would be interested in the forecast better-skilled at predicting the next flood.

These forecasts are generally conditioned on different data and variables, but a conversation about the models is outside the scope of this report. What is important is to understand that an objective skill score may be available for each sector, based on different data.

4 Some Forecast Verification Scores

Forecasters and practitioners often recommend using more than one score to better assess the attributes of a forecast such as reliability, resolution, discrimination and sharpness: " any set of forecasts can then be ranked as best, second best, ... worst, according to a chosen score, though the ranking need not be the same for different choices of score."

Although verification scores themselves are typically run by the forecasting center generating the forecast, there are commonly agreed-on metrics in verification.

The CAWCR (Collaboration for Australian Weather and Climate Research) maintains a website (Joint Working Group on Forecast Verification Research, 2016) describing many aspects and motivations to verify forecasts by a world-wide list of contributors, including a long list of well-known National Weather Centers and authors.

Some of the scores introduced on the CAWCR site are used in the scoreboard and database developed in this study, and introduced next: Skill scores, Brier score, RMSE, and CRPS.

4.1 Skill Scores

A skill score can be based on any of the following (Brier, CRPS, RMSE, etc) scores across different forecast systems, and is therefor one calculation we included in our Verification Scoreboard.

The user specifies a fixed reference value for each forecast; then the score is calculated as a ratio of the different between each forecast and the reference value.

$$skillscore = \frac{score_{forecast} - score_{reference}}{score_{perfect.forecast} - score_{reference}} \tag{1}$$

As noted in the introduction, an early and classic forecast validation conversation was begun with the publication of Finley's tornadoes (Murphy, 1996). Here's a recreation of Finley's contingency table according to Murphy:

		Observed		
		tornado	no tornado	**total**
Forecast	tornado	28	72	**100**
	no tornado	23	2680	**2703**
	total	**51**	**2752**	**2803**

Table 1 – Tornado Forecasts (1884 Finley)

As Finley argued, his forecasts were correct $(28 + 2680)/2803 = 96.6\%$ of the time. However, if one assumed *no tornado* – an 'unskilful' forecast, or reference forecast – the result is $2752/2803 = 98.1\%$, an improvement on Finley's accuracy.

In calculating skill scores, different reference forecasts may be used. A common one is the average expected forecast, generally sampled at random over a large probabilistic forecast dataset to obtain a representative sample. The Finley example is a deterministic one, so the reference forecast is always "no forecast".

Other baselines used include "persistence", mostly used in short-term forecasts; this basically says "the weather this hour will be the same as the last hour". Skill greater than zero means the forecast predicted change when observed. Finally, "climatology" is a common reference used, where the mean value of the variable over time (ex 30 year period of observations) becomes the baseline. (Jolliffe and Stephenson, 2012)

4.2 RPS Score

The Ranked Probability Score, a widely-used measure of probabilistic forecasts, is a squared quantification of the cumulative density function (CDF) of a forecast compared with the CDF of observations for the same / similar zone.

This score is related to the Brier and CRPS scores, described below, and based on similar construction. A fundamental problem with the RPS, however, appears as bias when it's used on ensemble systems with smaller numbers of members (eg less than 40 (Weigel, Liniger, and Appenzeller, 2007)).

4.3 Brier Score, Brier Skill Score

What's today called the Brier Score originally came from (Brier, 1950), called the *sample skill score*, in one of the first probability forecast articles (Murphy and WInkler, 1974). In this paper Murphy and Winkler introduce Brier Score "calibration-refinement factors": reliability, resolution, and uncertainty.

The Brier score is popular because it's been collected for a long time; has been "updated" periodically (Murphy and WInkler, 1974); is straightforward to calculate (uses contingency tables); and is easy to interpret: if a Brier score is close to 0 it's close to perfect; the closer to 1 it gets, the worse it is.

4.4 RMSE Score

Also used in hydrogeological models, the Root Mean Squared Error is the square root of the mean of the squared differences between forecasts and observations.

The RMSE gives more weight on larger errors than smaller errors, which is advantageous if large errors are worse than small ones, A zero represents the perfect score.

4.5 CRPS score

The CRPS (Continuous Ranked Probability Score) introduced in (Hersbach, 2000) is a standard of verification metrics. It shared roots with the RPS, above, but is less influenced by spatial discritization of datapoints. It is one of the most-relied upon values and consistently appears in sample datasets we received.

As with other skill scores, the CRPSS (Continuous Ranked Probability Skill Score) is calculated using a reference score and a "new" score; a value of 0 is good, closer to 1 or -1 is not.

Part III

Design of Scoreboard for Inter-Agency Comparison

5 Introduction

A challenge offered through this Masters dissertation was to create a scoreboard utility within the criteria of IMPREX project. The idea was to create a prototype and to test it with data sets generated by the group to evaluate both the function of the scoreboard.

Very little of the project was predefined. In the beginning a significant effort involved the selection of open-source software tools, an open workflow, and the "look and feel" of the scoreboard itself.

To accomplish these tasks I had to improve my knowledge and skills in a number of fields: understand verification of meteorological and hydrologic forecasts, learn the programming language R, evaluated file or database “data warehouse” options, and identify a platform that would also satisfy the goals of IMPREX.

6 Requirements to Meet, Tools

The technology for creating a web-based scoreboard which queries data from a database dates to the late 1990s; today such web-based systems are everywhere. What was unique about this project was finding the correct combination of tools to meet our requirements:

1. Build an evaluation framework to benchmark the performance of hydrological forecasts
2. Accommodate precipitation; temperature; flow discharge (with emphasis on discharges for my project)
3. Prototype a (HTML) **user** interface for visualizing the scorecard, considering multiple choices
4. Prototype a **provider/user** interface for "feeding" the score database
5. Should rely on Open Source tools
6. Best to use tools familiar to IRSTEA team
7. Should provide high-level programming environment
8. Flexible enough to accommodate needs of the (Europe-wide) IMPREX team
9. Maintainable, readable code
10. Test the scoreboard on seasonal reforecasts in France and at the global scale.

Open Source software has a number of advantages over proprietary (closed) software: with many sets of eyes on the code, it often proves to be better thought-out and more robust. Open Source software is often distributed freely, and therefor may have a larger user base than a costly closed alternative. According to the appendix of "Forecast Verification" (Jolliffe and Stephenson, 2012) written by Matthew Pocernich (National Center for Atmospheric Research in Boulder, Colorado, and author of much

of the "verification" package in R), open source tools may be preferable for the research context: these tools often has a larger user base, increasing likelihood that errors will be discovered; the language with the larger user base will likely be understood by more colleagues; and the source code may be evaluated and modified/extended if needed.

After considering Python, Java, and Microsoft tools from Excel and Access to SqlServer, the environment at IRSTEA and the individuals I met at the Reading Weather Center seemed to share an enthusiasm for R (and either RStudio or Tinn-R).

To better assess the requirements and best tools for the development of the scoreboard utility, I met with hydrologists, meteorologists, infrastructure programmers, and others at ECMWA. They are experienced with the PostgreSQL database platform for storing and querying large data sets, and was impressed at the commitment to open source tools.

Eventually a suite of tools formed:

- R, from the R project for Statistical Computing
- RStudio, the popular IDE from the eponymous RStudio group
- PostgreSQL, high-performance open source database
- Rmarkdown, an R package to create documents with R content by RStudio
- Shiny, an R package to create HTML5 web pages based on R content (also by RStudio)

Added to this list and discussed in the Workflow section, below, are the following tools which I used to write this document:

- git, an open source source control tool which integrates nicely with RStudio
- github, a web-based git repository which also functions as a backup
- QGIS, the open source GIS platform which I used to check locations and perform spatial queries
- LaTeX, text-based document preparation system
- Overleaf, an online Document collaboration service based on LaTeX
- Mendeley, reference management system like Endnote or JabRef, but integrates well with Overleaf

7 Existing Scoreboards

While there has not been a scoreboard designed to allow two climate centers to load their own score files, there are a number of different Scoreboards available today. Below are several examples: ECMWF based in Reading, UK; Met Office, based in the UK; NOAA, in the United States; and the KNMI Climate Explorer based in the Netherlands.

NOAA Weather Prediction Center

Some examples of scoreboards can be found from the websites of operational forecasting centers. For instance, Figure 5 shows a screenshot of the webpage on verification statistics from the Weather Prediction Center of the National Weather Service of NOAA in US. The example shows the evolution of the Bias and the Threat scores over the period 1970-2015 for different lead times.

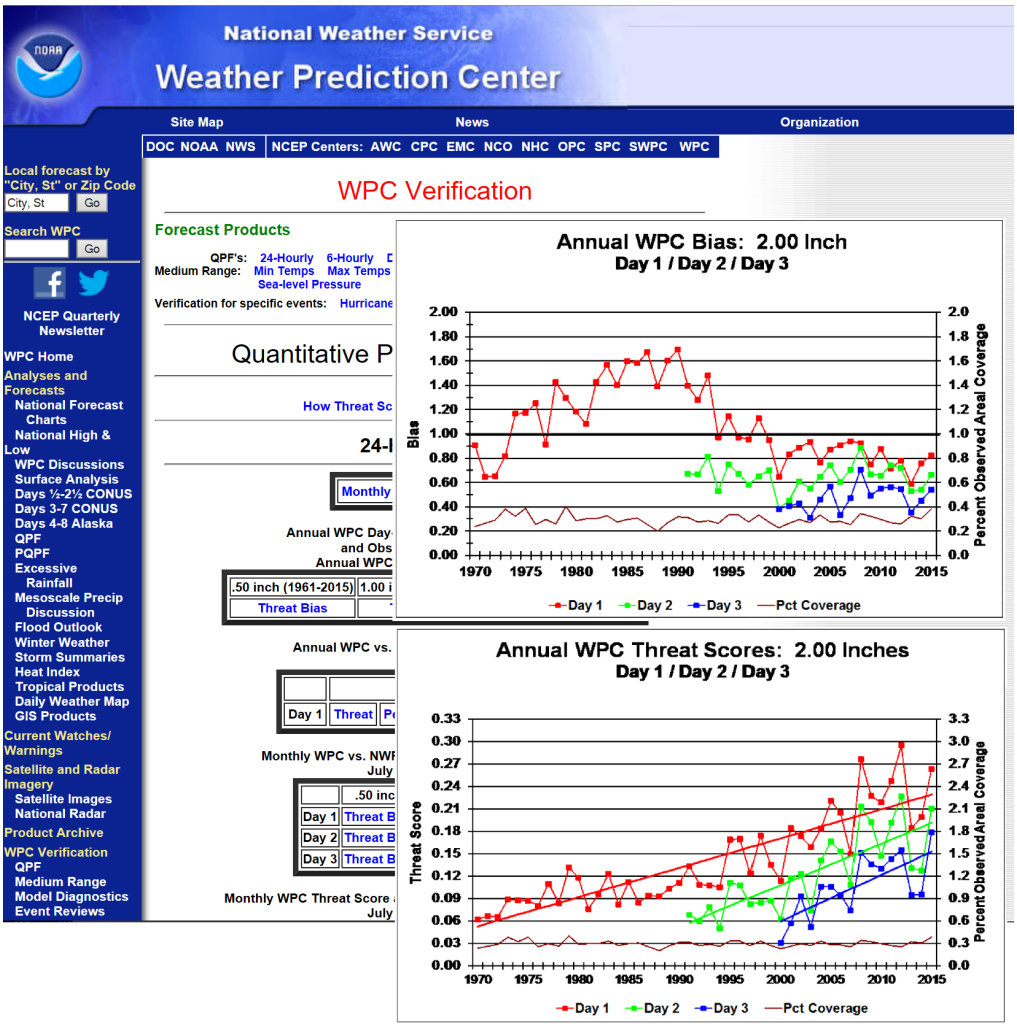

Figure 5 – NOAA's Weather Prediction Center

Also posted by NOAA, Figure 6 shows the verification of Quantitative Precipitation Forecasts (QPF) provided for a single event: Hurricane Sandy, which affected most of the eastern United States (especially the coastal Mid-Atlantic States) autumn 2012. Verification is provided through the comparison of the maps of the accumulated precipitation forecasts and the accumulated observed precipitation over the affected areas.

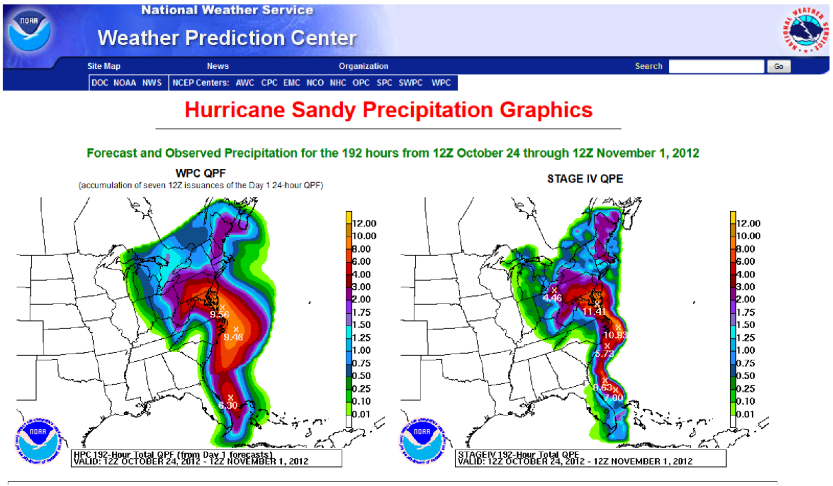

Figure 6 – NOAA's Hurricane Sandy verification page

NOAA's WPC Verification: http://www.wpc.ncep.noaa.gov/html/hpcverif.shtml

Link to NOAA Sandy Precipitation graphics

ECMWF Verification and Charts

The ECMWF has automated and opened to the public a number of verification tools; below the main page are three Brier Score plots for the same dataset over three lead times (4, 6 and 10 days). This nicely-illustrates the trade-off between skill and lead time.

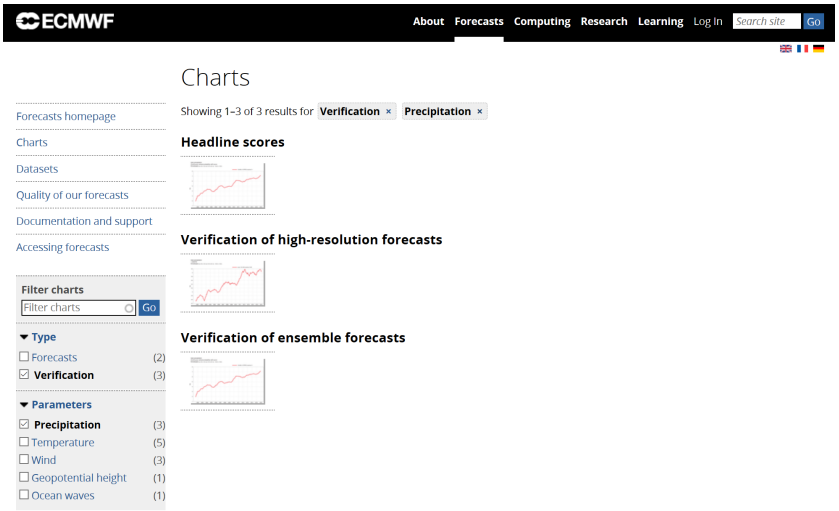

Figure 7 – ECMWF Forecast Verification Products

ECMWF's verification pages are here (select *Type* > *Verification* to limit the page to verification products): http://www.ecmwf.int/en/forecasts/charts/catalogue/

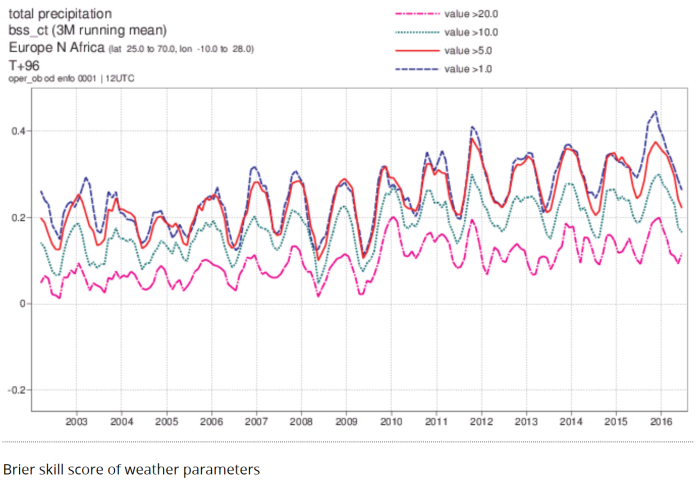

Figure 8 – ECMWF Brier Score on Precipitation; "forecast day" (lead time) = 4

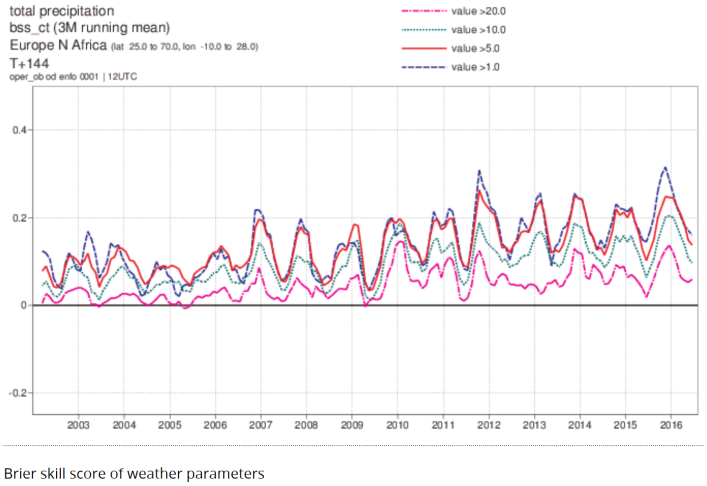

Figure 9 – ECMWF Brier Score on Precipitation; "forecast day" (lead time) = 6

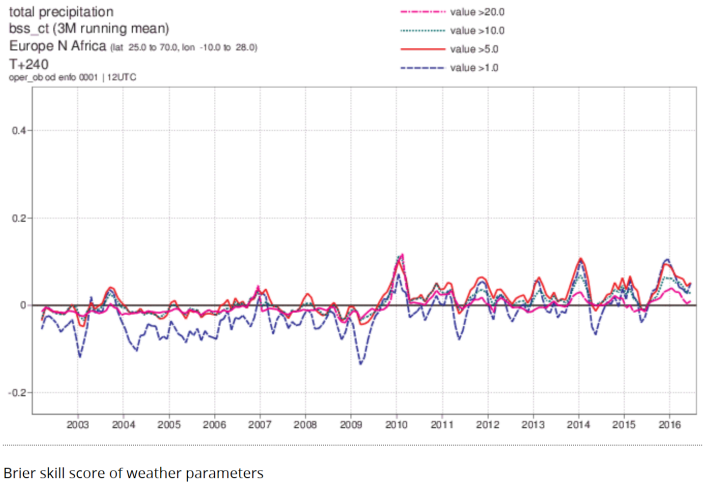

Figure 10 – ECMWF Brier Score on Precipitation; "forecast day" (lead time) = 10

MetOffice

Another example can be found in the website of the UK MetOffice for global long-range predictions. Probabilistic skill maps and plots are available and updated monthly for temperature and rainfall predictions up to six months ahead. Figure 11 illustrates how the user can change the "skill score type" to display a ROC score map or a Reliability diagram.

Other options on the visual display include the variable to display, the geographic area, and the period used for the computation of the scores.

The MetOffice webpage on global long-range model probability skill can be found here: http://www.metoffice.gov.uk/research/climate/seasonal-to-decadal/gpc-outlooks/glob-seas-prob-skill

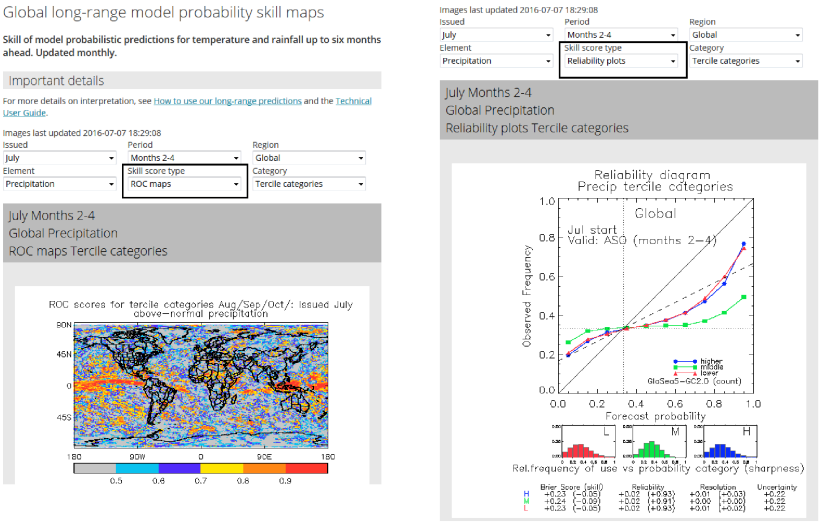

Figure 11 – MetOffice Prediction Center

KNMI Climate Explorer

The KNMI Climate Explorer bridges the database of calculated forecasts, reforecasts and observations, and allows users to add their data. Data formats accepted include NetCDF and flat text files. Although we did not have access to that part of the site, it claims the capability to verify scores spatially (as in the NOAA Sandy page) as well as classically by time series over lead time.

The KNMI Climate Explorer can be reached here; users need to sign in before proceeding the scores: https://climexp.knmi.nl/

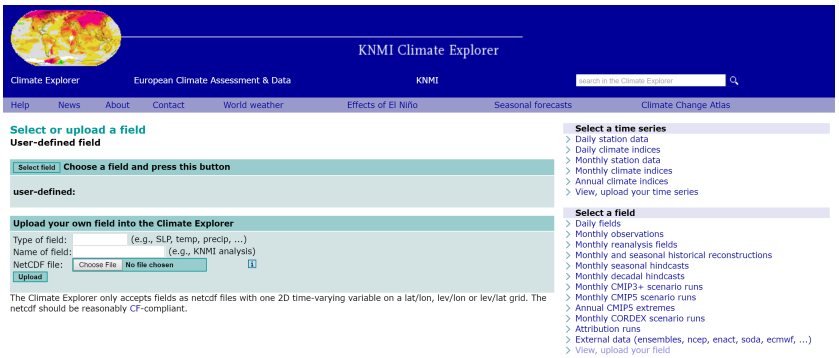

Figure 12 – KNMI's Climate Explorer works with internal forecast / observation files, or directly with NetCDF format

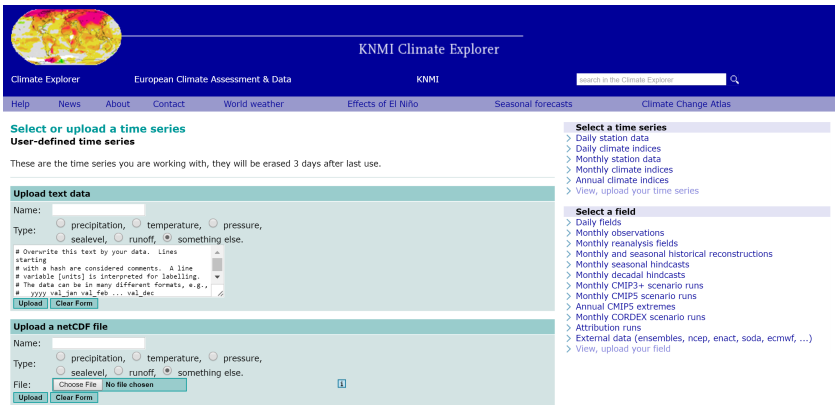

Figure 13 – KNMI's Climate Explorer allows users to upload their own text files

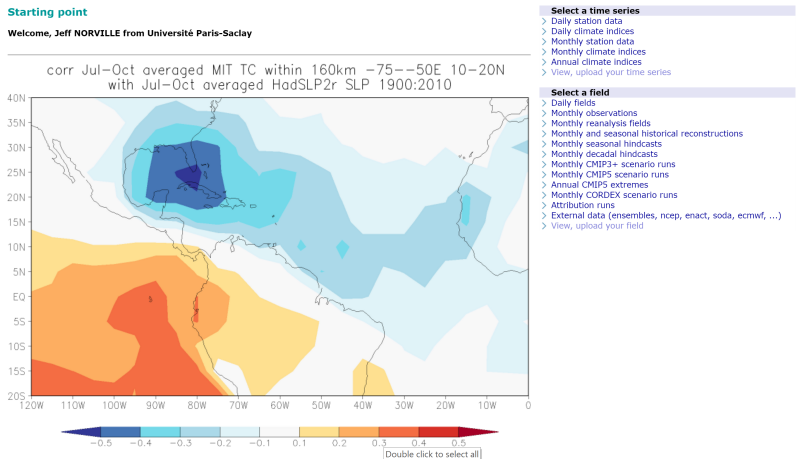

Figure 14 – KNMI's Climate Explorer with map-based verification. Note the page requires users to authenticate themselves

Part IV

Technology Decisions: Workflow and Tools

8 Workflow

While IRSTEA does use code standards, development source control systems, and team development standards for their Fortran and GR team development projects, this project involved non-statistical tools (Shiny package of RStudio, database and data manipulation suites of R libraries). However, as workflow is central to development and software lifecycle and improvements, we specify the processes below.

8.1 Revision Control

"Source Control" or revision control is a way of tracking historical versions, changes from others, and testing new insights in an organized, systematic way. **git** and **github** are two system which support source control, but their similar names can be confusing. The program **git** (created by Linus Torvalds in 2005) is installed and runs on your local computer; it doesn't inherently need an internet connection, but creates a repository (or repo) locally. This means your first synchronization with a new git installation essentially recreates your added file(s) entirely; subsequent changes are stored only when you "commit" them to the repo.

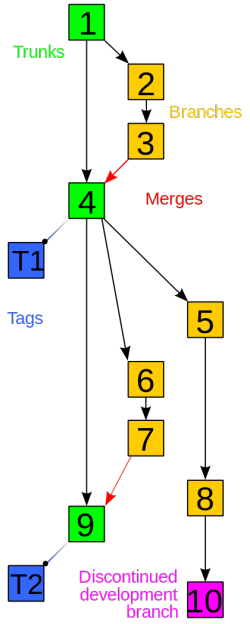

Figure 15 – git Example Workflow

The online service **github** doesn't require installation on your local computer, but, once signed up as a user (free), you can synchronize your local git repository with github and to see and share your source code online.

A critical concept in git and github is the "trunk" and "branch"; I include an introduction here, and refer the interested reader to more thorough and interactive documentation online: https://guides.github.com/introduction/flow/

The goal is to keep the "trunk" of the tree (usually called the master) as the always-clean, always-deploy-able code. When working on a new feature or fixing a bug, pull the latest code locally; create a well-named branch for your task (ex betterplots, below); improve code; commit it, and when it's ready to merge back into production there's any easy workflow below.

See Figure 15 for a visual scheme of a development trunk and individual branches.

The blue boxes in Figure 15 are "tags"; while similar to branches they are essentially dead-ends or archives, not living branches of the code repo.

For
this project I started one github repo (https://github.com/jeffnorville/shinysb1/) in April (Figure 16), then moved to a "cleaner" repo for the deployable project: (https://github.com/jeffnorville/VerifScoreboard)

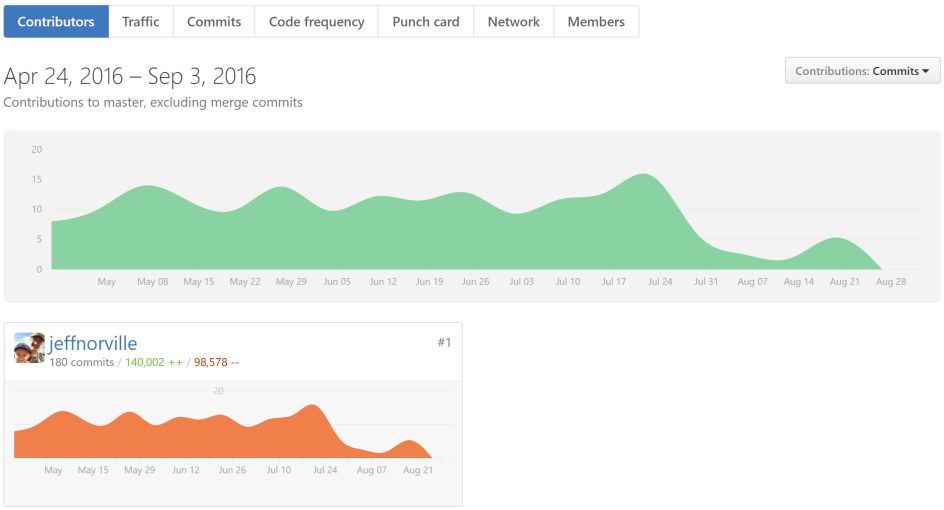

Figure 16 – my github Activity (shinysb1) during development

9 Tools: Shiny by RStudio

The sections below detail our experience using each tool for this project, so I've followed a repetitive format to include: 1) Options; 2) Compromises; 3) Notations (format specifications for example); 4) Lessons Learned (was this the right tool, or was there a better one?).

The most noticeable tool of the project was RStudio's (a popular IDE, integrated development environment) **Shiny**, which facilitates interactive webpage design.

9.1 Options

There are also options other than R for creating a scoreboard. EVS, for example, is written entirely in Java, a language targeting cross-platform users more than statistics. Options for creating a similar webpage / database framework are also many, from commercial partners like Microsoft's SqlServer database with a c# middleware and server pages. However, since R began providing Shiny libraries, there are not many systems designed to create a consistent look and feel across data access services to GUI which are based on one language.

9.2 Compromises

There are languages which feature richer web-based libraries, and better database integration than R; however for a user-base primarily interested in statistics and numerical solvers, R is not a compromise, but the better solution.

But Shiny as a tool is itself a compromise; it is a high-level language (R) and set of libraries which encapsulate difficult JavaScript, HTML and even database / SQL interfaces beneath a "familiar" R interface. The R expert can create dynamic, consistently

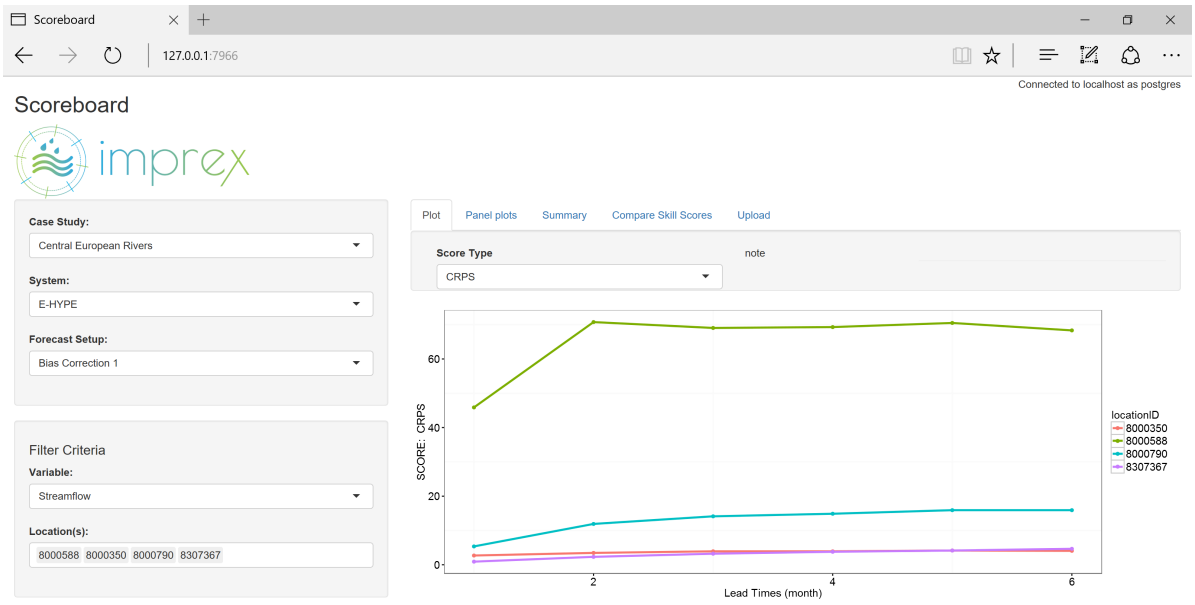

Figure 17 – Scoreboard example

attractive pages based on existing libraries; and deploy them quickly, without dabbling in the other technologies.

In order to create the pages online one also needs to be connected with a Shiny server; RStudio hosts servers which are free at first (www.shinyapps.io), but there is a cost with increasing usage.

It is also possible to host Shiny servers on any server platform, for example on an inexpensive cloud-based service like Amazon Web Services, or any other variety of enterprise web server.

Once the webpage is is active on a Shiny server, of course, it can no longer read data from the local PostgreSQL database instance on one's computer; it reads only from a database server. PostgreSQL runs on a server too, naturally (AWS also provides a data service). According to the Shiny server documentation reviewed, server performance is best if the database instance and the Shiny server are not running on the same machine (ex with the same processors), as the memory- and processor-intensive operations while generating a large or complicated graphic would compete for the same resources on a shared server.

9.3 Notations

Shiny applications are typically split into two executable files: *ui.R* and *server.R*. The overarching goal is to separate the logic of the application from the display – separating form from content.

Otherwise standard R format applies throughout Shiny.

```
require(dplyr)
```

```
require(ggplot2)
toto1 <- filter(toto, locationID %in% basin.list[1:9])
ggplot(toto1, aes(x = leadtimeValue, y = scoreValue)) +
  geom_boxplot() +
  facet_wrap(~ locationID) +
  xlab("Lead Times") + ylab("RMSE Scores")
```

9.4 Lessons Learned

Uploading data from a web interface to a database is universally a hard thing to do. Shiny does this well, but there's a small risk in loading Rdata files from unknown users: the variable names in the Rdata files aren't controlled, and once they are loaded onto the server space they **replace variables on the server application of the same name**.

This behavior is documented but not well-respected in the online Shiny community. A solution exists for a "serialized" data connection – essentially limiting the "Rdata"-like file to one parent entity, and related data objects. This file type is called the RDS filetype, and was our choice in implementing the Data Loader function.

Another problem that remaines to be solved is identifying locations and their proximity. The best alternative to the current selection-box system, which requires a user to select points by a listed name, would of course be to select points from a map. The leaflet R library offers a good interface based on layers (one could overlap different score types as independent layers, selecting groups of points nearby one another, or based on another geographic similarity), but integrating leaflet – and in particular automating the import of geospatial data by anonymous users – remains outside the scope of this project.

10 PostgreSQL

For our current deployment we used PostgreSQL 9.5, as of publication of this document the latest version is 9.6R1.

A database is simply a collection of data; in the classic computer definition it divides the structure of data (tables, queries, schemas and so on) from the data content itself (records, values). An analog database example might be a file cabinet, with the papers contained within the folder in file drawers the records.

However working a database into the source control workflow, even a high-quality system like PostgreSQL, is more complicated; there is no true source control for a database.

For this reason backups or restores are typically done in two steps:

1. Backup / Restore Structure
2. Backup / Restore Content

Occasionally people have stored the "data definition language" for the database structure in git. To simplify distribution and maintenance, I split the structure into a file called *"structureDATE.backup"* and *"contentDATE.backup"*, where the DATE is replaced by the creation date. For now these are stored on my github account, but as they are the largest binary file I will find another system to distribute them.

PostgreSQL is installed with its own GUI called pgAdmin (Figure 18). This utility can create backups and restores directly; the user (with database permissions, ex password) can view the structure directly, and write short data requests to view contents using Structured Query Language (SQL).

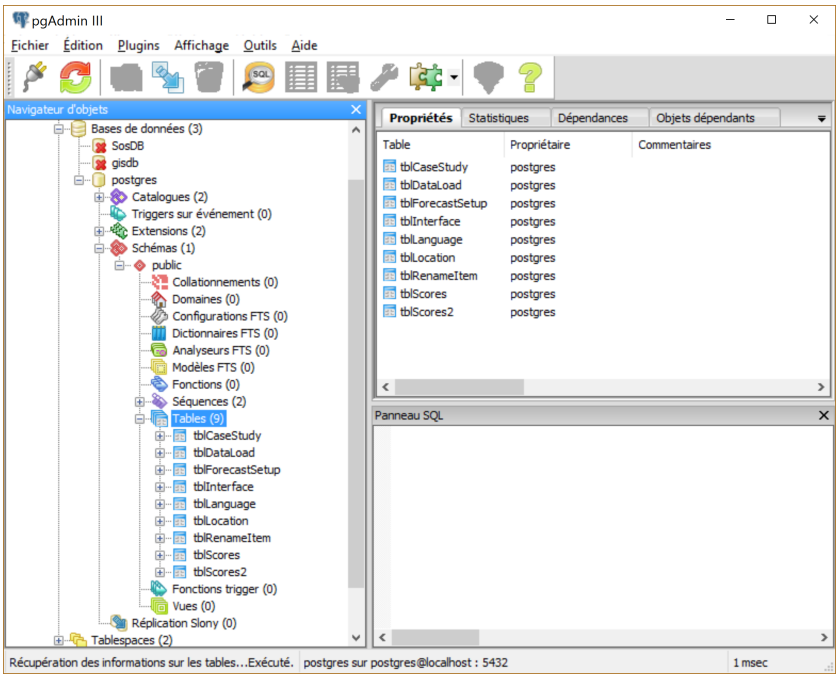

Figure 18 – PostgreSQL pgAdmin GUI interface

10.1 Options

Many alternative databases and file systems exist as an alternative to PostgreSQL:

1. Sun Microsystems MySQL
2. Oracle database
3. Microsoft SqlServer
4. NetCDF data files
5. SOS database (used by ECMWF and implemented recently)

One of the reasons ECMWF and other GIS or data-driver model entities rely heavily on PostgreSQL is it accommodates an open source utility called PostGIS, which integrates spatial data in the database. Similar functionality can be had in Oracle and Microsoft systems, but as exorbitant costs (for an academic institution).

As noted under the list of alternative options to this database, the SOS Database is actually a framework which runs on a combination of PostgreSQL, PostGIS, and several other open-source components (Tomcat, Apache tools). ECMWF recently implemented this to make available their water level forecasts; as of this time their version does not make available observations at all (personal communication Paul Smith, 23 august 2016).

While we hope there are many users of the system, we have anticipated accommodating limited types of data: numeric, date, and integer values, which will be received and processed as score files. We request spatial data (location information as coordinates) but that is optional; currently we do not have a method to upload shapefiles to the database (a potential request from team members).

During imports where our default data schema cannot be matched, or the partner does not use R in-house, we can build a custom import tool. With each successive data import the import method becomes "smarter", adding a mapping for our database and allowing future automated imports. This does not exist today.

10.2 Compromises

Our system needs to account not for time-series data in every case, but also for data on different timescales; for this reason we've included two different "scales" of dates in the database, technically a "no-no" for a database administrator, but a necessary evil to accommodate our potential users.

Currently the database accepts "point based" geographic coordinates in one coordinate reference system, but many users work with shapefiles (polygons). There are add-on modules available for PostgreSQL which work with shapefiles, but we did not implement them for this project.

10.3 Notations

A short example of a database creation script:

```
CREATE TABLE "tblScores"
(
  "row.names" text,
  "locationID" text,
  "scoreValue" double precision,
  "forecastType" text,
  "dateValue" date,
  "datePartValue" numeric, -- valid numeric values: 12 (month), 51 (week)
  "datePartUnit" text, -- valid values: "month", "week"
  "leadtimeValue" integer,
  "leadtimeUnit" text, -- daily weekly monthly
  "scoreNA" boolean, -- if scoreValue == NA then TRUE
  "scoreType" text, -- lookup to other table of forecast types
  "modelVariable" text,
```

```
  "dataPackageGUID" text
)
```

10.4 Lessons Learned

I made a number of errors with PostgreSQL implementation, most of which were due to being out of date with the tools or general complexity.

1. SQL's date type column is the fastest and most-efficient index to use when querying large datasets by actual dates; however, I initially overlooked that the vast majority of our data is NOT timeseries data
2. Amazon Web Service for remote hosting - due to the installation being slightly more complicated than I anticipated, the free AWS tier was quickly exceeded
3. PostgreSQL "series" datatype (automatic sequence) not compatible with R's write table function

As noted, although our first dataset relied heavily on daily values, and due to it's size benefitted from using the "date" datatype in the database, the majrity of our score datafiles do not have daily date values; additionally, using a date field for a month value when that value represents perhaps 30 years of months is incorrect, and misleading to future code maintainers.

We solved this by implementing both a date field and a datePart field, which is paired with datePartUnit. One or the other is mandatory for each record, but they are also exclusive of one another. This means a record will either have a date: "2011-08-04" ; or an entry in datePart ("34") paired with datePartUnit ("week").

For the last point, PostgreSQL "series" datatype remains a sticky issue. We rely on automatically-generated indexes when loading new datafiles into the PostgreSQL tables, which is a standard operation, and I modified the RPostgreSQL library to work with the datatype on local installations. However, updates to the RPostgreSQL library – or installations to new servers, other laptops – do not respect the local modification. To date the maintainer of that part of the source code library has not accepted the proposed change (easiest path).

11 R and RStudio

11.1 Options

For graphical output and statistical models, R seems to be winning the open-source – and even among the proprietary – tool sets in popular circulation today.

Base R and R IDEs are often confused, the IDEs are a developer convenience – the R base installation may be accessed directly from the command line.

Popular IDEs other than RStudio include Tinn-R, RCommander, Eclipse, and Microsoft's Visual Studio.

11.2 Compromises

RStudio is available for Linux and Mac as well as Windows, and it is a stable product. However, it is fair to say using a language designed for statisticians to create a web interface is a strange concept to many outside the R universe!

12 Conclusion on Technology Decisions

The general workflow and technology using R and Shiny make a lot of sense for this workflow. There are many examples of such simple interfaces, hosted both by RStudio and through their forums. The vast majority of the Shiny pages I reviewed, however, use either a static datafile on the server or "scrape" data dynamically from another service or page (weather, elections, etc). There are not as many database-driven Shiny websites today.

While the database interfaces to Shiny seem somewhat under-developed (lack of support for auto-incrementing series, error handling is limited, lack of awareness of database "state"), the database part of this project is too important to disconnect the two.

Another option could be to write a custom set of libraries to speed up communication from the R Shiny scoreboard and the database directly. Because of the nature of R, the authors have opened up their source code to facilitate that approach to programmers who need it.

Part V

Scoreboard Design

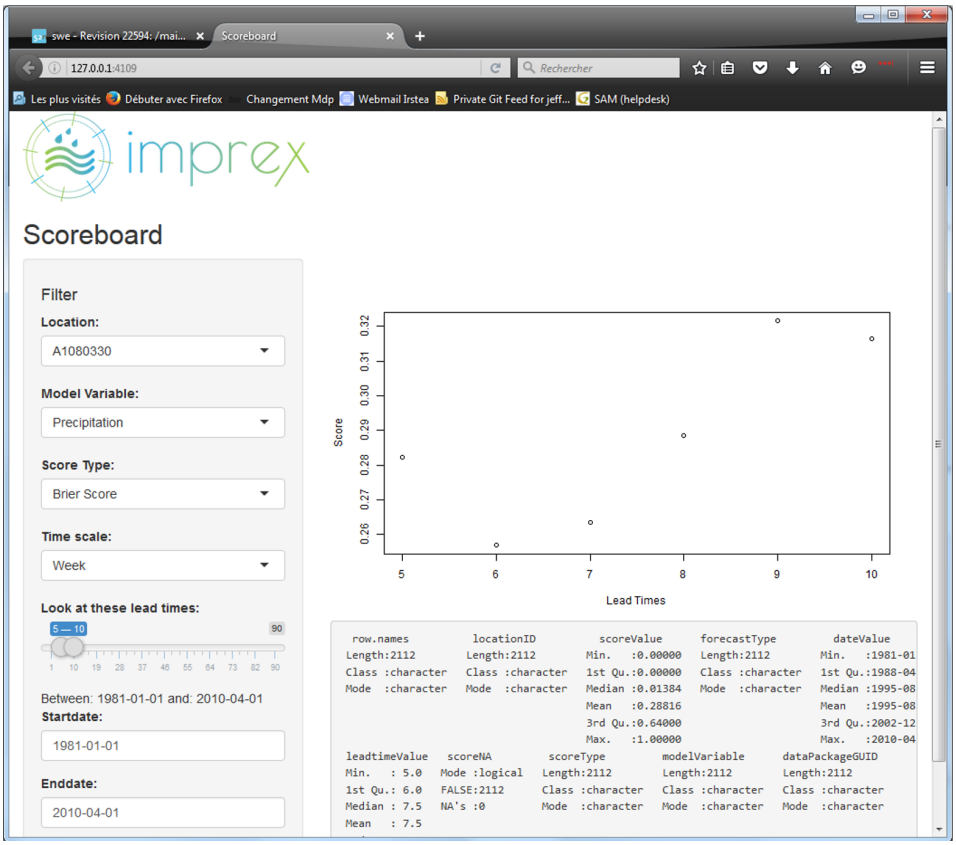

Figure 19 – Scoreboard v1

As noted elsewhere, a "universal" scoreboard was an undefined concept. Here are three iterations using our chosen toolchain (RShiny, postgresql) to build test frameworks.

13 Version 1 - Working With Daily Values

My first "scoreboard" was based on the first data we received – a big data set with daily values over 30 years, lead times to 3 months (90 days), for 16 French catchments. That dataset included four model data types: two different bias corrections for streamflow and precipitation.

Figure 19 shows the navigation bar, left, where the user may change Location; Model Variable; Score Type; Time Scale; then there's a slider to select between one and 90 lead times, and a date range selector in case one wants to tune comparisons of different data sets.

This plot shows one score type over just six lead times, and the standard R "summary" function below the plot was useful to check selections during development.

Figure 20 shows the first scoreboard again but with more lead times selected (1:75). This first version did not accommodate multiple selects (even on Location), but that was due to a "bug" in the Shiny library I was using: I could use 2:n values to search, or one value, but the change between the two was quite awkward. I solved this problem in future scoreboards.

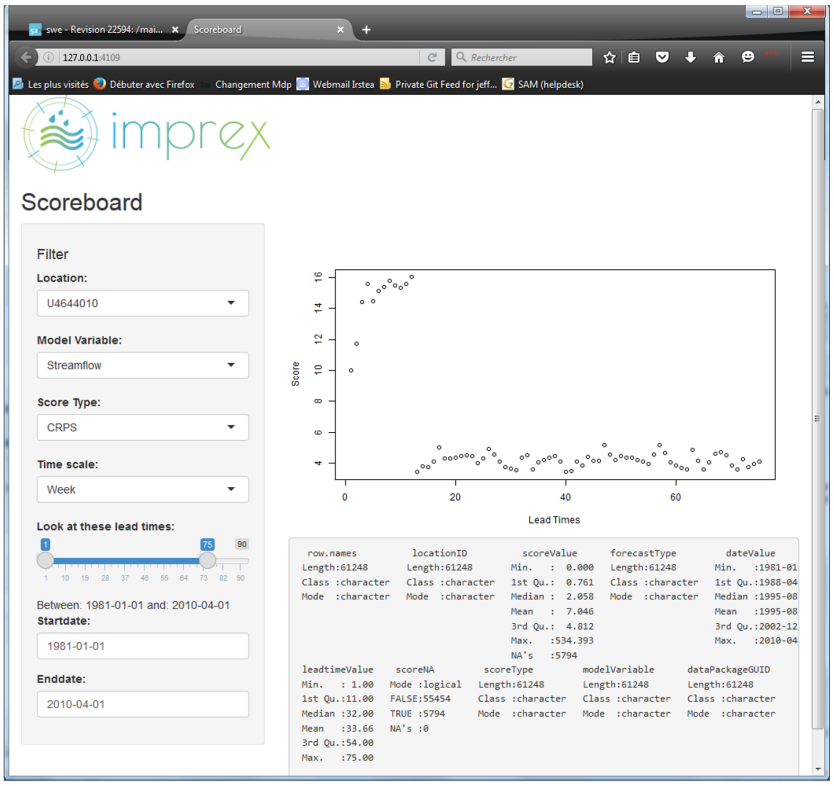

Figure 20 – Scoreboard v1, note leadtimes 1:75 selected by "slider"

14 Version 2 - Displaying Details

The second version incorporated the same controls (since we were using the same first dataset) with nicer graphics, namely the ggplot2 library and default color scheme. One big enhancement based on ggplot: we added confidence intervals on every data point. Another was including a function to return confidence intervals and standard error when running monthly averages on the entire dataset.

The error bars were included because every plotted point in Version 1, above, is a mean monthly value taken on 30 years of daily values compared against lead time. In order to "display" the depth of the data, including an R function to count the values, collect basic stats, and return a dataframe which made the error bars easy to add and modify made sense. In this image the bars are 95% confidence intervals.

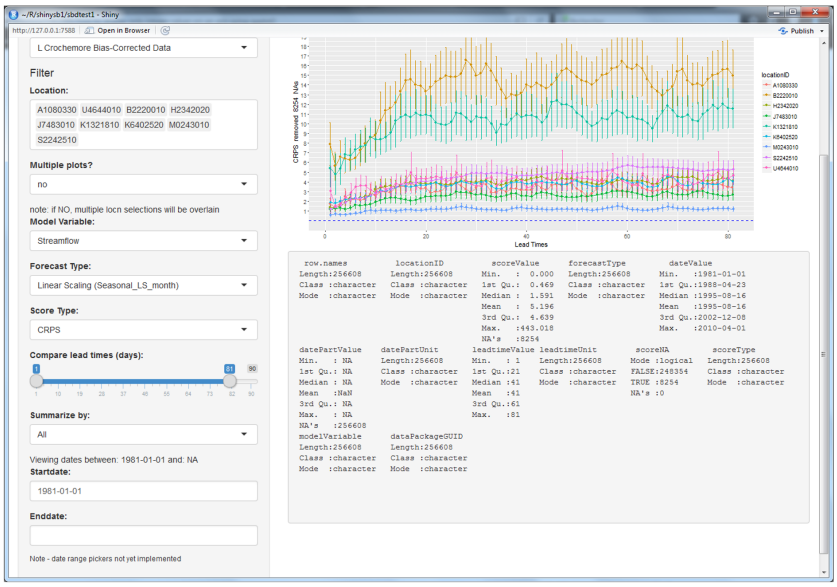

Figure 21 – Scoreboard v2, note color change in plot

However, the error bars were, in fact, less interesting than other comparisons we could perform between locations, between score types.

15 Version 3 - Data Mining

The third (and latest) scoreboard version looks quite different than the first two, partly because we received two new data sets that were quite different from the first.

While the initial dataset had **daily** values including daily projections of lead time, the next two datasets were somewhat more processed. Values were already averaged by month, then sliced across years, creating one value which represented nearly a thousand values in the first dataset. (This varies slightly, as not each dataset started and finished on the same date.)

To illustrate the differences in volumes of data, see Table 2 below.

When using daily datasets (*dataset 1*), more data are stored. Receiving score data averaged to one month from the entire 30 year range (*dataset 2*) means we have fewer data to process before makng the display. In fact, we received double the 3 month lead time noted in(*dataset 2*), for 6 months of lead time (see *dataset 2 extended*), which allows for deeper comparisons.

Combining the two resolutions of data in the display required some changes to the way the database schema handled date values, but in this version the resolution of data is nearly transparent to the user (the more compact datasets response a bit more quickly).

While we set out to define our own data standards, we want to encourage team members to send us their score data without too much rework. We changed assumptions that team members would send us daily values, as 1) it does not add much to the scoreboard product, and 2) it will "fill up" the database very rapidly to import daily

	Interval	Years	Values per Year	Lead Times	Total
dataset 1	daily	30	365	90	985500
dataset 2	month avg	30	12	3	1080
dataset 2 extended	month avg	30	12	6	2160

Table 2 – Size Comparison by Dataset Resolution

values.

As of the writing of this document I have not yet aggregated the original dataset to monthly values to match the other two.

Table 3 is our outline for creation of the three main "plot" panels of the Verification Scoreboard:

- Plot (see Figure 24)
- Panel Plot (see Figure 25)
- Compare Skill Scores Plot (see Figure 27)

The Scoreboard User may choose a Forecast System, a Setup, a single Model Variable, but any number of Locations.

More complex is the decision matrix under Compare Skill Scores tab: the user is creating their own skill score based on the chosen score as a "reference"; they are limited to choosing either two Systems and one Setup, or one System allowing two Setups. This programming is taken care of in Shiny.

	System	Forecast Setup (System)	Locations	Variable	Score
Plot	1	1	n	1	1
Panel Plots	1	1	n	1	n
Skill Comparison Plots	1	2	n	1	n
	2	1	all		

Table 3 – Choices for User Selections in latest Scoreboard

Additionally, if there are more than twelve Locations selected, the line plot changes to a series of box-whisker plots. See below, Scoreboard Testing, for an introduction to the Skill Comparison Plots.

Part VI

Scoreboard Testing

16 Score Data Used for Testing

We received three data deliverables from two different weather centers, ultimately loading one from each center to the IMPREX database for testing purposes: SMHI and ECMWF.

16.1 SMHI Score Data

Swedish Meteorological and Hydrological Institute of Sweden (SMHI) sent us score data calculated from their E-HYPE model. E-HYPE is a Pan-European hydrological model for seasonal streamflow forecasts that runs over 35000 sub-basins (median resolution=215 km^2) across all of Europe.

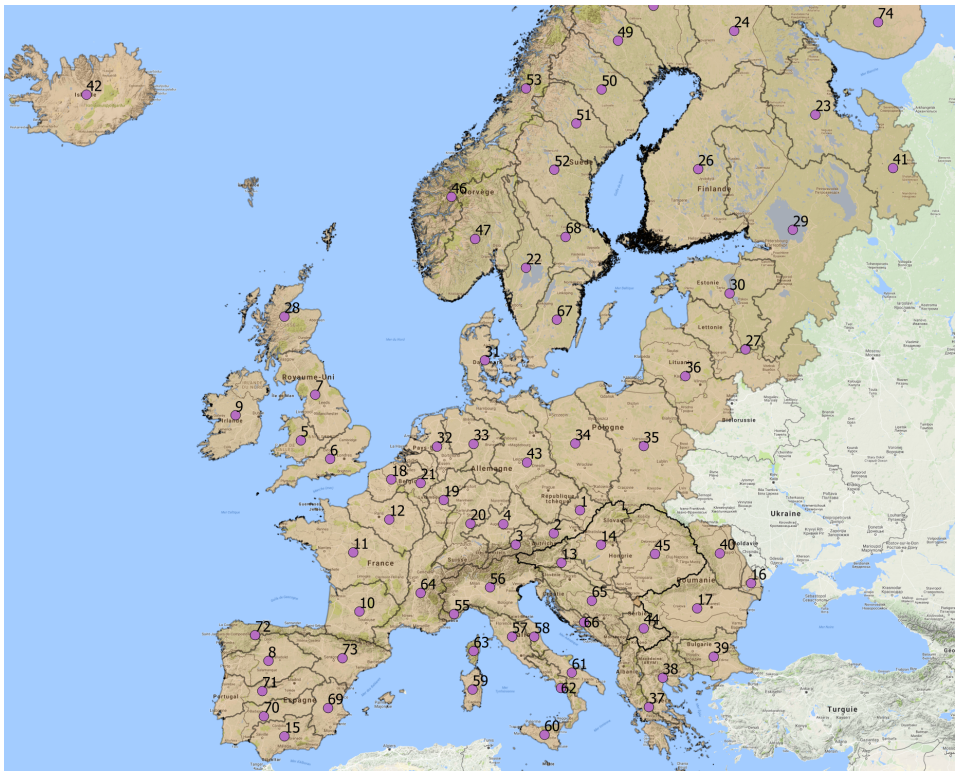

Figure 22 – E-HYPE Shapefile Centroid Data

Daily streamflow forecasts were obtained using the System 4 seasonal precipitation forecasts from ECMWF as input to the E-HYPE hydrological model. The streamflow forecasts were verified by SMHI with a variety of numerical scores, including Brier score, correlation, CRPS, RMSE, and related skill scores.

The E-HYPE score data were provided in RData file format. It contained scores and skill scores for 825 stations in Europe. The file was about 50 MB and contained scores for each month of the year, 6 lead months and each station (i.e., 12*6*825 data points per score).

16.2 ECMWF Score Data

System 4 EFAS data was provided by the European Centre for Medium-Range Weather Forecasts (ECMWF). Score data come from the evaluation of seasonal streamflow forecasts issued by EFAS (European Flood Awareness System).

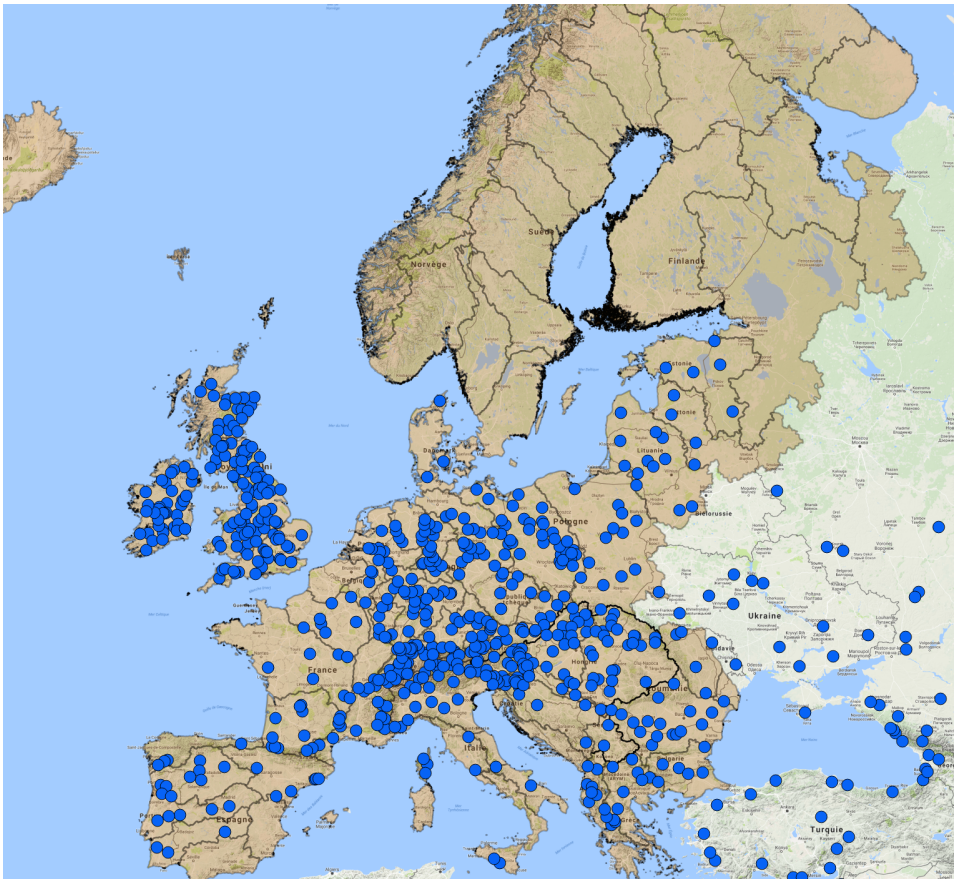

Figure 23 – ECMWF Point Data

Daily streamflow forecasts were obtained using the System 4 seasonal precipitation forecasts from ECMWF as input to the daily LISFLOOD hydrological model, following its set up in Europe for the EFAS project. LISFLOOD is a GIS-based, distributed hydrological rainfall-runoff-routing model. It is run for all of Europe on a 5x5 km grid. Streamflow is simulated on a pixel basis.

The streamflow forecasts were verified by ECMWF and we have collected CRPS values for the score database. The score data were provided in text file format for 74 basins over Europe. The scores are average scores over all the years of data, for each month for which the forecast is made and each month of lead time (up to 7 months). We have received score values for each basin (not for a station as in the E-HYPE data). As it was explained to us, these scores come from the quality evaluation of average monthly discharge values over each basin. Daily discharges were first spatially averaged over each pixel inside the basin area, and then temporally averaged over a month, before proceeding to the forecast verification. Shape files were also provided for the geographical location of the basins.

Note that the EFAS LISFLOOD model uses a custom global projection; in order to

reproject EFAS points into WGS84, the standard system for our scoreboard, a conversion was necessary.

Additionally, in order to compare locations between these two datasets, I used a QGIS to find nearest neighbor points, and renamed the EFAS points to share the same name as their nearest-neighbor EFAS locations. This issue would be better resolved in R, which has a number of geospatial libraries available to either allow actual spatial queries in Shiny, or to create some pilot points which share similar-enough geographical information as to be useful comparing forecast systems.

16.3 Testing Results: Screen Captures

In this section we walk through several use-cases of the Verification Scoreboard.

The navigation features are consistent throughout the application, with minor exceptions. The user begins by selecting a Case Study, all nine of which are defined by the IMPREX project. Central European Rivers is our default choice and where most of our test data is located.

The user is then presented with all possible Systems, or Forecast Systems; in the test database we can chose from E-HYPE and EFAS SYS4. Depending on that selection, the next field "Forecast Setup" is populated with valid selection choices. Finally, the selection box "Model Variable" is populated from prior selections, and the user can typically chose between precipitation, Temperature, and Streamflow. During testing only Streamflow data have been loaded.

Now the interface can create a simple plot once the user selects one or more locations. The user may also change the current Score Type above the plot; in this version all Score Types are shown each time, rather than a selection of valid values from the database.

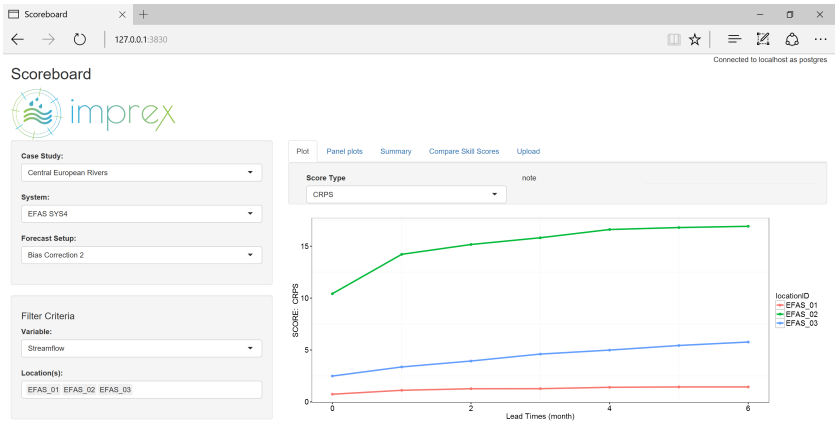

Figure 24 – Scoreboard Plot function

A small legend appears to the right of the plot to indicate which selected location matches each curve. As with all figures in this scoreboard, the X axis is always the lead time of the forecast scores. As shown in this screenshot, and logically, the CRPS scores increase over lead times; note that a higher CRPS score indicates an increase in

the (square of the) difference between observed and predicted values, so a lower value is better. Typically a forecast of today or the next month will be better than one for five or six months from now.

The user may switch directly to the Panel Plot next, which will display the same locations but change the default scores to be: CRPS Skill Score; Brier Skill Score; and RMSE Skill Score. This default behavior is because the only way to hold all variables equal to directly compare scores is to compare skill scores. Other scores are weighted on flow volume, or the size of the basin, or other factor related to a specific location.

Since the database did not have skill scores for these values, I added CRPS in order to plot figures. In stead of the plots being overlain the user sees sequential plots.

In the event there are multiple locations AND multiple scores, the matrix of plots grows in the X and Y direction, respectively.

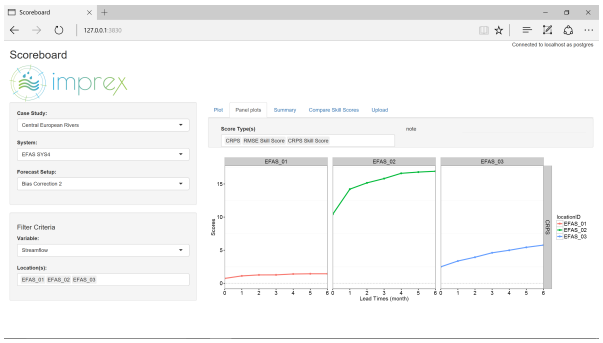

Figure 25 – Scoreboard Panel Plot function

Next, the "Summary" tab generates a small R-style summary of data currently selected.

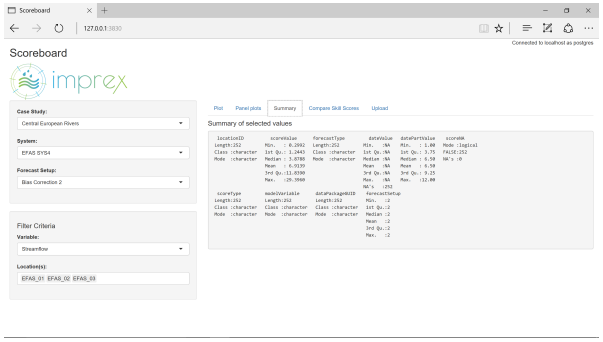

Figure 26 – Scoreboard Summary function

Finally, the tab titled "Compare Skill Scores" is next, and this allows the user to customize their own skill score.

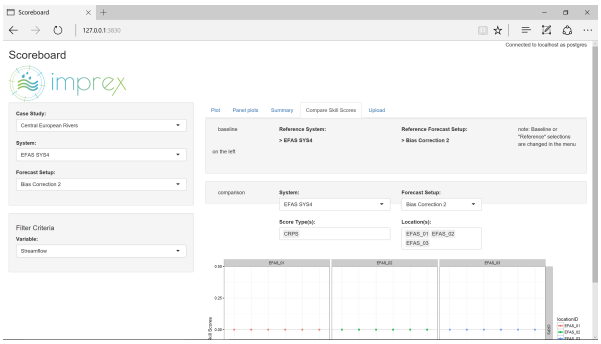

Figure 27 – Scoreboard Compare Skill Score Plots

The upper zone titled “baseline” mirrors the user’s selections on the lefthand bar, in this case EFAS SYS4 and Bias Correction 2. This becomes the “reference forecast”.

To make a comparison the user selects an available system under “comparison”; it IS a valid choice to copy the “baseline” selections, but note that you are comparing the same System / Setup to itself; the plot will be zeros right across.

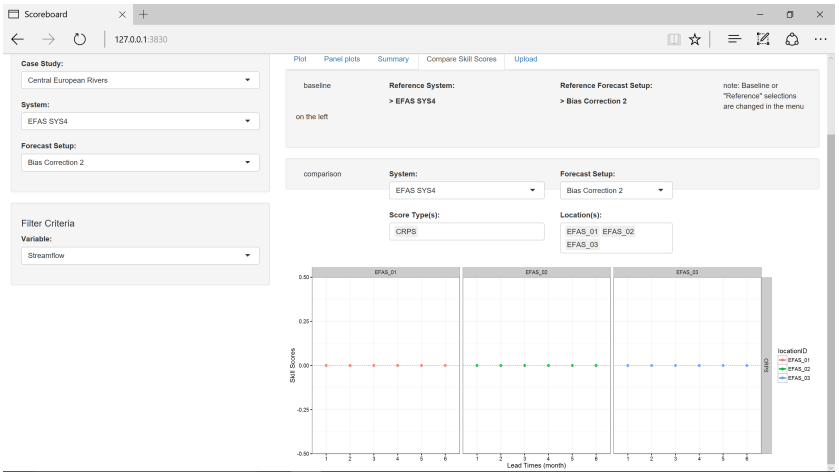

Figure 28 – Scoreboard Compare Skill Score Plots, lower part

If you chose a different System and Score, a new selection box becomes available: all possible Scores for that combination. On selecting a score (always CRPS for the test case), a new skill score plot will be made.

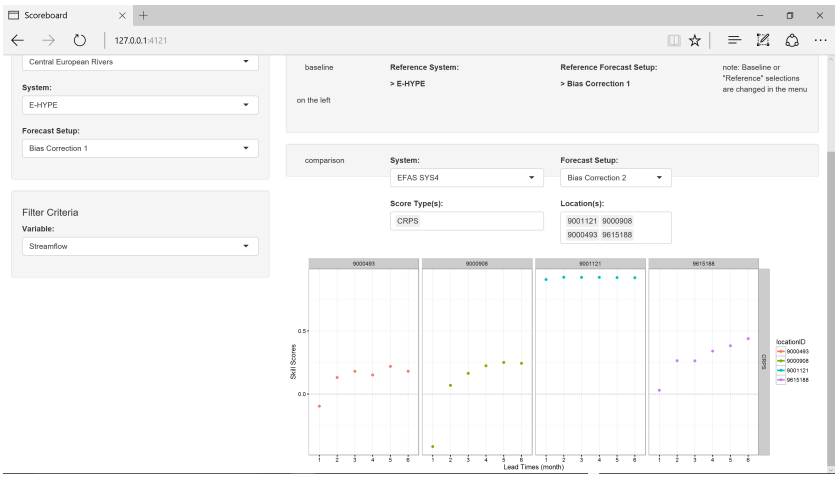

Figure 29 – Scoreboard Compare Skill Score Plots - Reference, Compare Forecasts shown

A value of zero (0) means the two forecast systems were perfectly aligned, or had identical skill. The other extreme on this scale is one, the maximum score; minus one (-1) is also valid, and the two indicated lack of skill. In the example here (Figure 29), the reference system is E-HYPE; it's being compared with EFAS SYS4. The first plot (9000493) shows that at a lead time 1 EHYPE does a bit better than EFAS; however, from LT = 2 through 6 EHYPE loses while EFAS manages increasing lead times better.

That is generally repeated with 9000908 and 9615188. Location 9001121 shows EFAS consistently better.

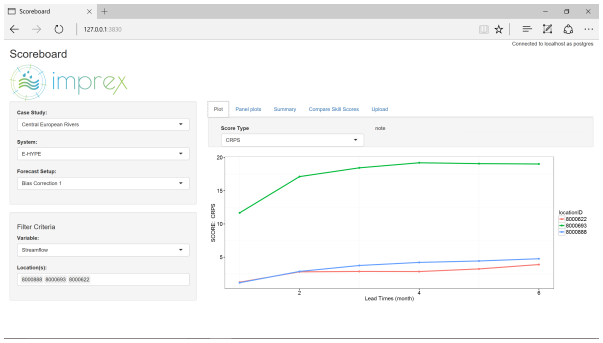

Figure 30 – Scoreboard Plot - changed System, Setup automatically adjusts

Figure 31 – Scoreboard Panel Plots

16.4 Conclusion on the Design of the Scoreboard

In order to evaluate scoreboard performance and features we evaluated the scoreboard installed on one computer. At the time of testing we did not use remote servers for the database nor the web pages.

Overall the scoreboard meets our design goals, and provides three graphical plotting interfaces: Plot, Panel Plot, and Compare Skill Scores. The user can navigate between the interfaces, changing criteria and exploring how models diverge in skill over time and according to different forecast setups (ex Bias Correction, post processing, etc).

As we tested on one local computer, moving the system to a hosted solution (PostgreSQL server, Shiny server) infrastructure may increase latency within the application. If this is the case, adding a "progress bar" to let the user know when a slow query is taking place may improve responses to the interface. Additional tuning of the software may be investigated.

Part VII

General Conclusions and Way Forward

Forecasting hydrology and meteorology events involves complex systems of data collection, physical and/or empirical modeling. This project targeted only an understanding of the verification process – evaluating predicted values against an observation or confident value – and the numerical scores that modelers may produce. Finally, the project succeeded in defined and constructing a useful and interesting utility that had not existed before.

There are clear opportunities to improve the scoreboard as it exists today. The reporting system (outputting graphics and PDF files of queries created) is not complete, as the creators of Shiny have change their libraries; a more robust system of saving markdown PDFs is available. Our upload system works on well-formatted RDS files, but is inflexible about reporting errors.

As noted in the report, currently we select a "location" from a list; with increasing numbers of users, both data submitters and evaluaters, the location identifiers will become impossible to evaluate. Implementing a graphical system for point selection is important.

Finally, with increasing numbers of users we will begin hearing more community feedback, and ideas from those with expertise in forecast verification – and a little distance from the tool itself – for improvements and changes.

References

References

Brier, Glenn W (1950). "Verification of forecasts expressed in terms of probability". In: *Monthly weather review* 78.1, pp. 1–3.

Finley, Jno P (1884). "Tornado predictions". In: *American Meteorological Journal* 1, pp. 85–88.

Hersbach, Hans (2000). "Decomposition of the continuous ranked probability score for ensemble prediction systems". In: *Weather and Forecasting* 15.5, pp. 559–570.

Joint Working Group on Forecast Verification Research (2016). *Forecast Verification - Issues, Methods, and FAQ - cawcr*. [Online; accessed 6-September-2016].

Jolliffe, Ian T. and David B. Stephenson (2012). *Forecast Verification*. Ed. by I. T. Jolliffe. 2nd ed. Wiley, p. 292. ISBN: 9781119960003.

Jones, Jeanine (2015). "Workshop on Improving Sub-Seasonal and Seasonal Precipitation Forecasting for Drought Preparedness". In: *Proceedings of the Western States Water Council/California Department of Water Resources*. Ed. by Tony Willardson and Cheryl Redding. Jennifer Segur, Graphic Services Branch, California Department of Water Resources, pp. 1–24.

Kahneman, Daniel (2011). *Thinking, fast and slow*. Macmillan.

Murphy, Allan H (1993). "What is a good forecast? An essay on the nature of goodness in weather forecasting". In: *Weather and forecasting* 8.2, pp. 281–293.

— (1996). "The Finley affair: A signal event in the history of forecast verification". In: *Weather and Forecasting* 11.1, pp. 3–20.

Murphy, Allan H and Robert L WInkler (1974). "Probability Forecasts: A Survey of National Weather Service Forecasters". In: *Bulletin American Meteorological Society* 55.12, pp. 1449–1452.

Weigel, Andreas P, Mark A Liniger, and Christof Appenzeller (2007). "The discrete Brier and ranked probability skill scores". In: *Monthly Weather Review* 135.1, pp. 118–124.