

# Evening Workshop Calibration Assignment

Michelle Lam, Alex Reed, Adelaide Robinson

2023-05-11

## Part 1: R function that codes a metric for performance evaluation

- must be a combination of at least two performance measures
- include some comments that explain ‘why’ this metric

```
# read in Sager data
sager = read.table(here("sager.txt"), header=T)
head(sager)

# add date
sager = sager %>% mutate(date = paste(day,month,year, sep="/"))
sager$date = as.Date(sager$date,"%d/%m/%Y")

#read in the function
source(here("compute_model_accuracy.R"))

#initially testing the model on the sager.txt file
compute_model_accuracy(m =sager$model, o = sager$obs,
                      month = sager$month, day = sager$day,
                      water_year = sager$wy)
```

Our function includes 2 metrics: - correlation: This is the correlation between the modeled and observed max stream flow for December and January. This metric is important in evaluating model performance when we are interested in accuracy of stream flows during spawning season. December and January are based on peak coho spawning months but the user can input different months if interested in a another season.

- rmse: This is the root mean square error between the modeled and observed average stream flows. We transformed this metric so that it falls between 0 and 1 and a higher RMSE indicates a better score. This metric is important in evaluating model performance when we are interested in average stream flow for water management purposes.

The output of the function is a single value that combines correlation and rmse by multiplying the values.

## Part 2

Apply your performance function to a subset of the Sagehen data set (with multiple simulations) that you want to use for calibration Summarize the performance over the calibration period in 1-2 graphs; you can decide what is useful Record your ‘best’ and ‘worst’ parameter set in this spreadsheet and in your Rmd

```

# multiple results - lets say we've run the model for multiple years,
#each column is streamflow for a different parameter set
msage = read.table(here("sagerm.txt"), header=T)

# keep track of number of simulations (e.g results for each parameter set)
# use as a column names
nsim = ncol(msage)
snames = sprintf("S%d",seq(from=1, to=nsim))
colnames(msage)=snames

# lets say we know the start date from our earlier output
msage$date = sager$date
msage$month = sager$month
msage$year = sager$year
msage$day = sager$day
msage$wy = sager$wy

# lets add observed
msage = left_join(msage, sager[,c("obs","date")], by=c("date"))

head(msage)

# how can we plot all results - lets plot water year 1970 otherwise its hard to see
msagel = msage %>% pivot_longer(cols=!c(date, month, year, day,wy), names_to="run", values_to="flow")

# p1=ggplot(subset(msagel, wy == 1970), aes(as.Date(date), flow, col=run))+geom_line()+theme(legend.pos
# p1

# # lets add observed streamflow
# p1+geom_line(data=subset(sager, wy == 1970), aes(as.Date(date), obs), size=2, col="black", linetype=2)

# subset for split sample calibration
short_msage = subset(msage, wy < 1975)

# use apply to compute for all the data
res = short_msage %>% select(-date, -month, -day, -year, -wy, -obs ) %>%
  map_df(compute_model_accuracy, o=short_msage$obs, month=short_msage$month, day=short_msage$day, water
# note here we use map_df to get a dataframe back

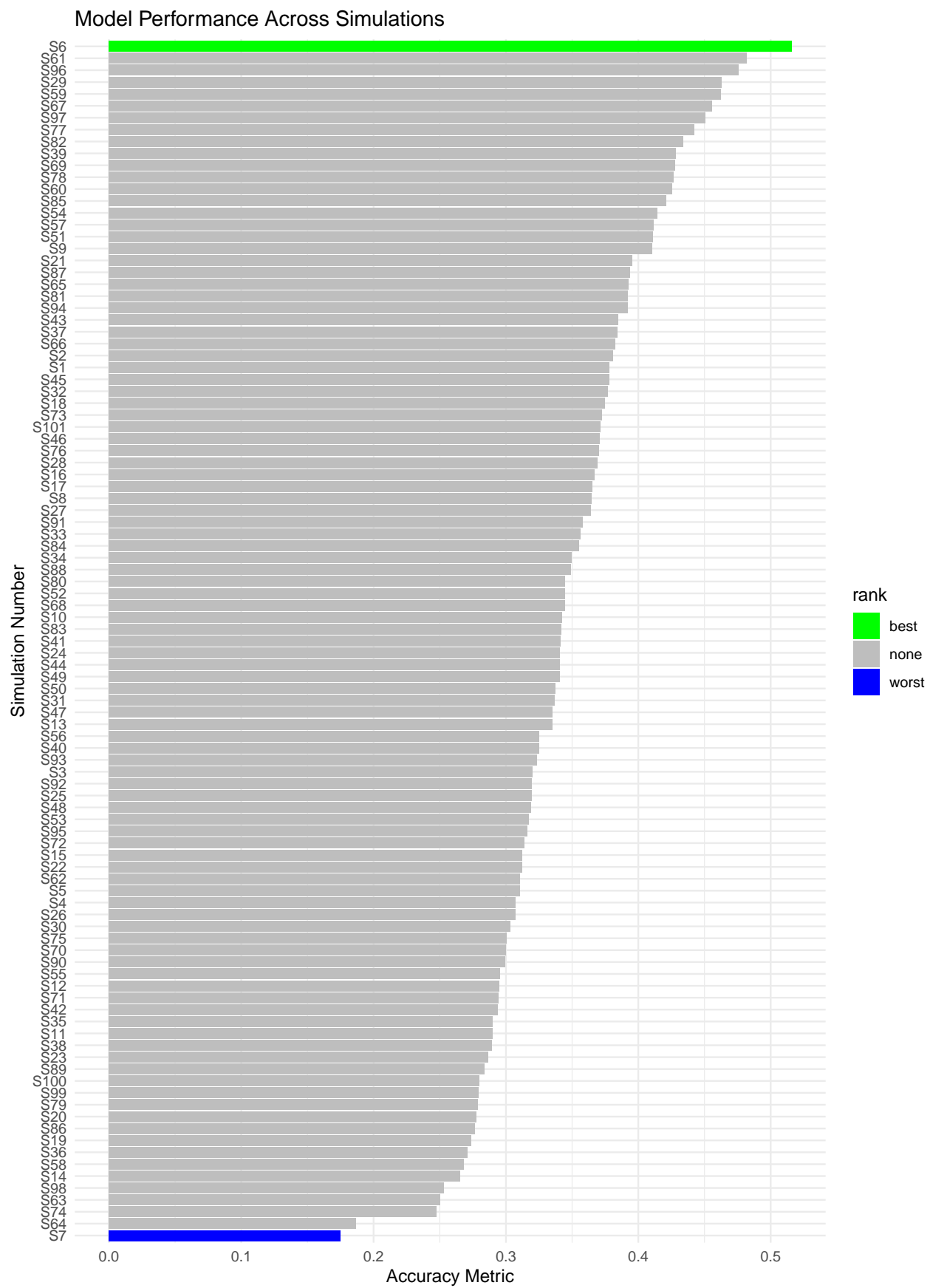
names <- names(res)

# create data frame to store simullation number, metric values, and rank
resl = res %>% pivot_longer(names, names_to="metric", values_to="value") |>
  mutate(rank = case_when(
    value == min(value) ~ "worst",
    value == max(value) ~ "best",
    value != min(value)|max(value) ~"none"))

# graph range of performance measures
ggplot(resl, aes(x = value, y = reorder(metric, value), fill = rank)) +
  geom_col() +
  labs(title = "Model Performance Across Simulations",
       y = "Simulation Number",

```

```
x = "Accuracy Metric") +  
scale_fill_manual(values = c("green", "grey", "blue")) +  
guides(fill = guide_legend(n.dodge = 5)) +  
theme_minimal()
```



```
# store best model performance
best <-resl[which.max(resl$value),]

# store worst model performance
worst <- resl[which.min(resl$value),]
```

```
#print th best and worst
best
```

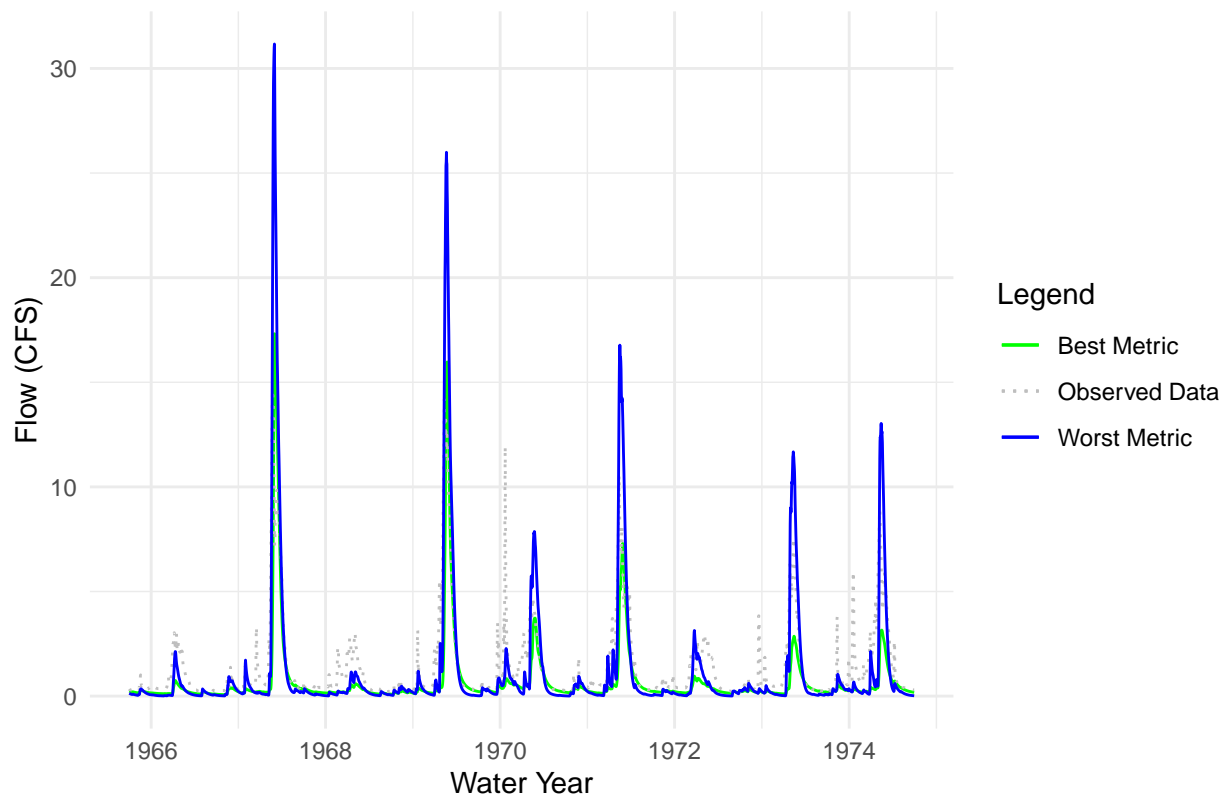
```
## # A tibble: 1 x 3
##   metric value rank
##   <chr>   <dbl> <chr>
## 1 S6      0.516 best
```

```
worst
```

```
## # A tibble: 1 x 3
##   metric value rank
##   <chr>   <dbl> <chr>
## 1 S7      0.175 worst
```

```
# create plot to compare best and worst model performance against the observed for the calibration period
ggplot(short_msage, aes(date)) +
  geom_line(aes(y = short_msage[, best$metric], color = "Best Metric", linetype = "Best Metric")) +
  geom_line(aes(y = obs, color = "Observed Data", linetype = "Observed Data")) +
  geom_line(aes(y = short_msage[, worst$metric], color = "Worst Metric", linetype = "Worst Metric")) +
  scale_color_manual(name = "Legend",
                     values = c("Best Metric" = "green", "Observed Data" = "grey", "Worst Metric" = "blue")) +
  scale_linetype_manual(name = "Legend",
                       values = c("Best Metric" = "solid", "Observed Data" = "dotted", "Worst Metric" = "dashed")) +
  labs(title = "Comparison of Best and Worst Model flow to Observed before 1975 Water Year",
       x = "Water Year",
       y = "Flow (CFS)") +
  theme_minimal()
```

## Comparison of Best and Worst Model flow to Observed before 1975 Water \



Our best model performance was on S6 with a metric of approximately 0.516. Our worst model performance was on S7 with a metric of approximately 0.175. The above plots illustrate model performance across the simulations and show the stream flow across the best, worst, and observed data for the calibration period (water year before 1975).

## Part 3:

Extra Credit.

## Part 4

As stated in part 1, we chose to look at the correlation between the modeled and observed max stream flow for December and January. The relationship between the observed and modeled max stream flow during this time is important because it could impact our understanding of max stream flows available during spawning salmon.

For our second metric we looked at the the transformed root mean squared error between the modeled and observed average stream flows. This metric is important in evaluating model performance when we are interested in average stream flow for water management purposes.

Combining these two metrics could be important for a water management agency who wants find a model that both helps them understand average water flows during the entire year and give staff doing fisheries monitoring an accurate idea of maximum flows during a season that is important for a specific species.

Our uncertainty analysis shows that there is a high amount of variation between model performance within the given data. It also highlights how evaluation of model performance is highly dependent on the metrics

used to assess accuracy.