

Sagehen_HW

Flora & Kristina

2024-04-25

Part 1: Come up with a combined metric that you think is interesting

See: combined metric file. We used equally weighted nse, rmse, and relerr.

Comparing model and observed time series output

```
sager = read.table("sager.txt", header=T)
head(sager)
```

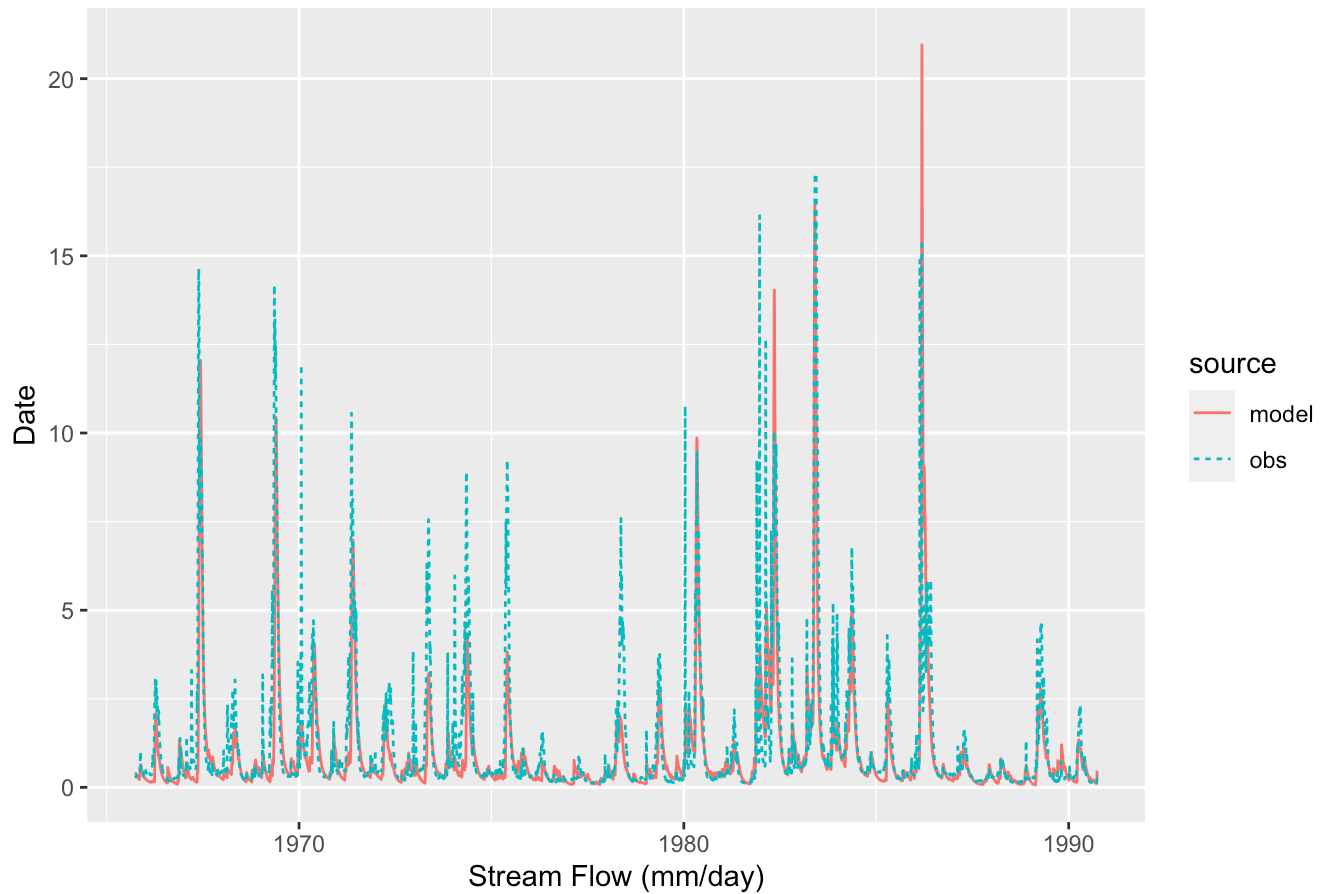
```
##      model      obs month day year  wy wyd
## 1 0.4238063 0.3358678    10   1 1965 1966  1
## 2 0.4133587 0.3208737    10   2 1965 1966  2
## 3 0.4032640 0.3058796    10   3 1965 1966  3
## 4 0.3935287 0.2968832    10   4 1965 1966  4
## 5 0.3841480 0.2968832    10   5 1965 1966  5
## 6 0.3751000 0.2968832    10   6 1965 1966  6
```

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

# plot
sagerl = sager %>% pivot_longer(cols=c("model","obs"), names_to="source",
                                values_to="flow")

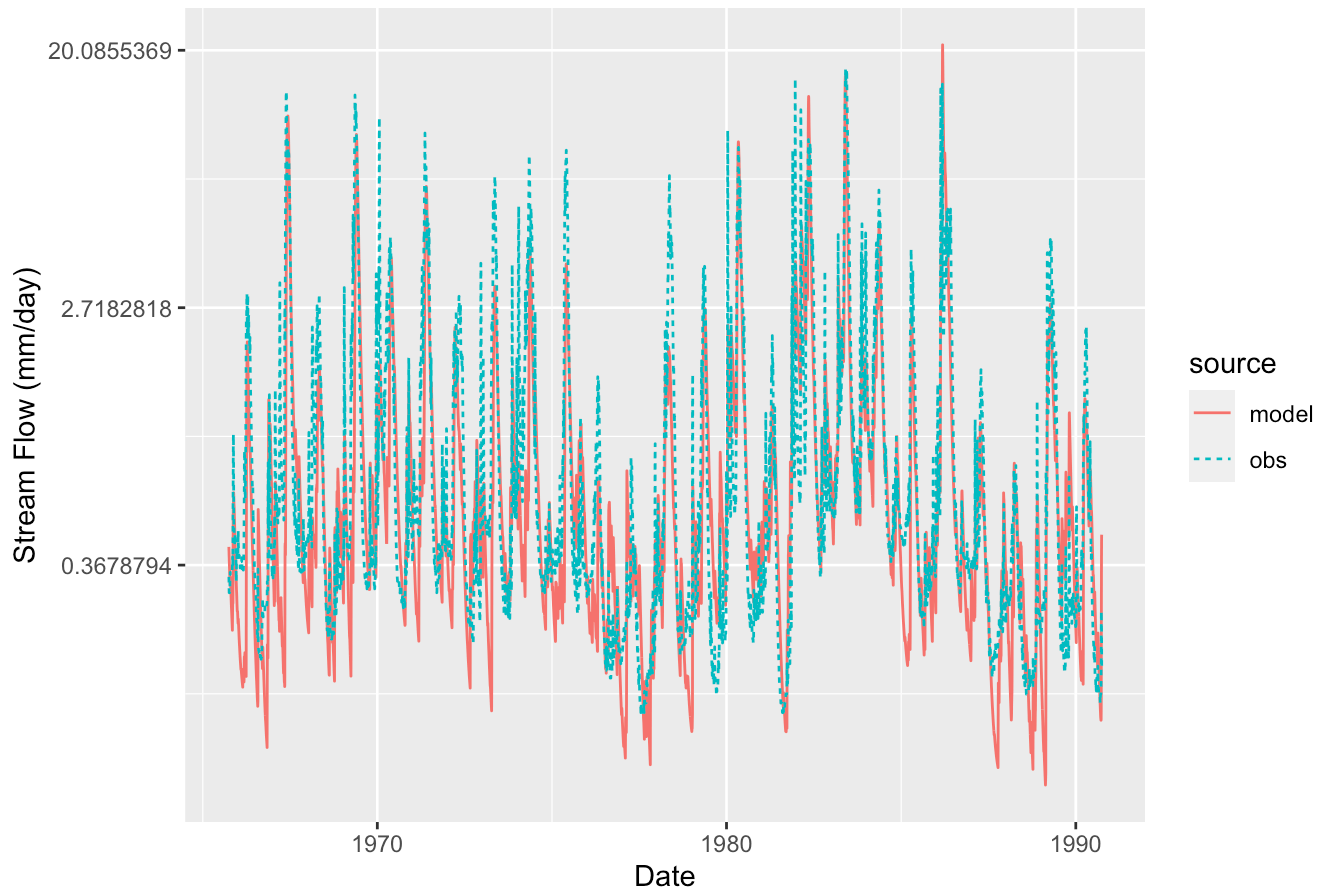
# basic plot
ggplot(sagerl, aes(date, flow, col=source, linetype=source))+geom_line() +
  labs(title= "Time Series of Observed and Modled Sagehen Creek Stream Flow", y = "Date", x = "Stream Flow (mm/day)")
```

Time Series of Observed and Modled Sagehen Creek Stream Flow



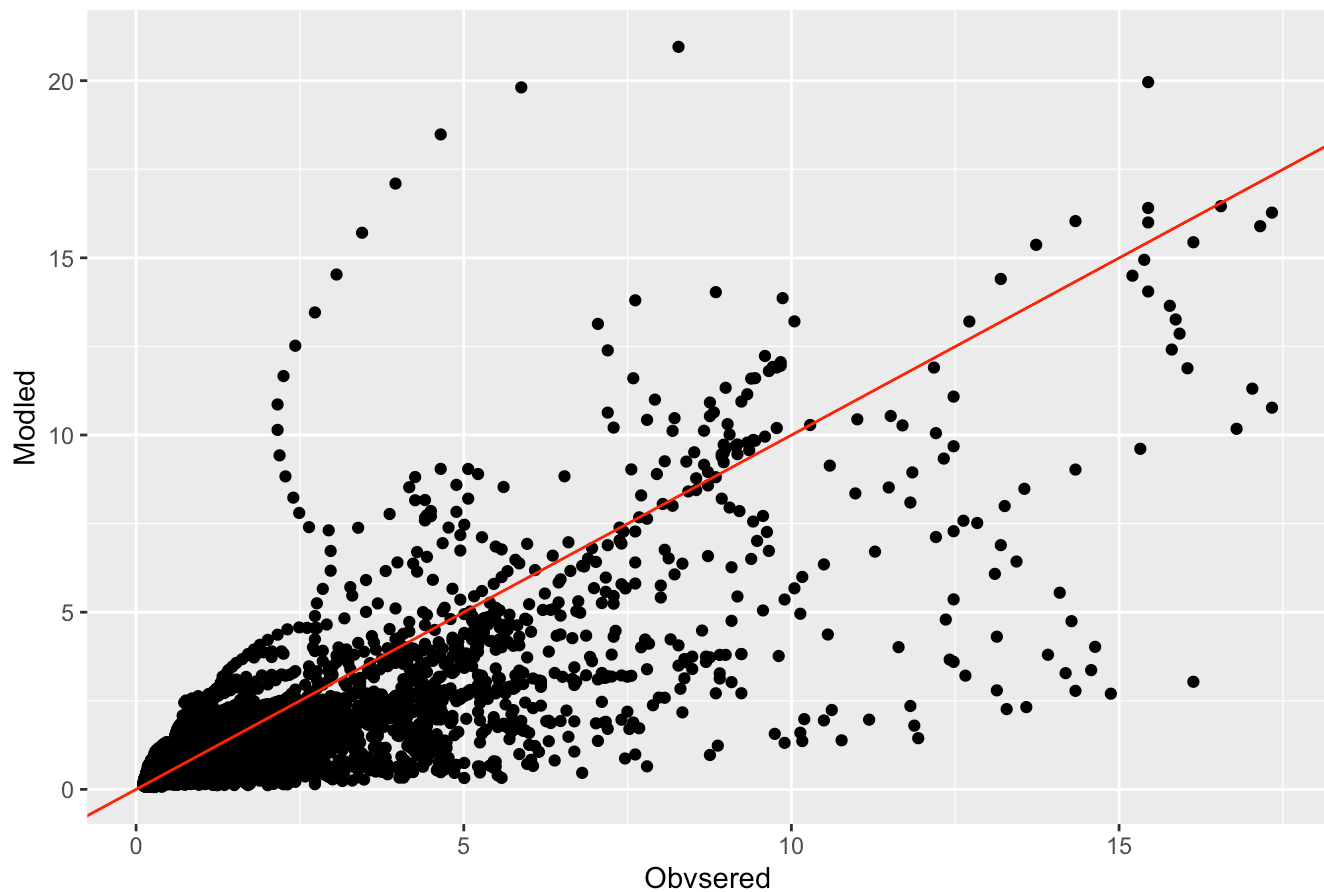
```
# change axis to get a closer look at performance at low values and take log
ggplot(sagerl, aes(date, flow, col=source, linetype=source))+geom_line()+scale_y_continuous(trans="log")+labs(y="streamflow mm/day")+
  labs(title= "Time Series of Observed and Modled Sagehen Creek Stream Flow (Log Transformed)", x = "Date", y = "Stream Flow (mm/day)")
```

Time Series of Observed and Modled Sagehen Creek Stream Flow (Log T



```
# look at it another way
ggplot(sager, aes(obs, model))+geom_point()+geom_abline(intercept=0, slope=1, col="red")
+
  labs(title = "Modled and Observed Streamflow of Sagehen Creek", x = "Observed", y = "Modled")
```

Modled and Observed Streamflow of Sagehen Creek



Measure Performance using different metrics

```
source("combined_metrics_K.R")
source("nse.R")
source("relerr.R")
source("cper.R")

nse(m=sager$model, o=sager$obs)
```

```
## [1] 0.6253416
```

```
relerr(m=sager$model, o=sager$obs)*100
```

```
## [1] -18.9577
```

```
cper(m=sager$model, o=sager$obs, weight.nse=0.8)
```

```
## [1] 0.5002733
```

```
combined_function(m=sager$model,o=sager$obs, weight.nse=0.33, weight.rmse=0.33, weight.r
elerr=0.34)
```

```
## [1] 0.5176637
```

Scale and subsetting

We decided to subset for rainy season, which is known to be between Nov. 1 and March 31. in California; rainy season brings higher flow-rate

```
# turn your evaluation metric into a function
source("high_flow_K.R")
# use different high flow months
#change from august to Nov - March
high_flow_metrics_K(m=sager$model,o=sager$obs, month=sager$month, day=sager$day, year=sager$year, wy=sager$wy, high_flow_months = c(11:3))
```

```
## $annual_max_err
## [1] -1.930857
##
## $annual_max_cor
## [1] 0.8897863
##
## $high_month_err
## [1] -7.861297
##
## $high_month_cor
## [1] 0.8537229
```

Create a combined metric

```
#see combined_function
combined_function(m=sager$model,o=sager$obs, weight.nse=0.33, weight.rmse=0.33, weight.r
elerr=0.34)
```

```
## [1] 0.5176637
```

Part II Split Sample Calibration

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

msage$date = sager$date
msage$month = sager$month
msage$year = sager$year
msage$day = sager$day
msage$wy = sager$wy

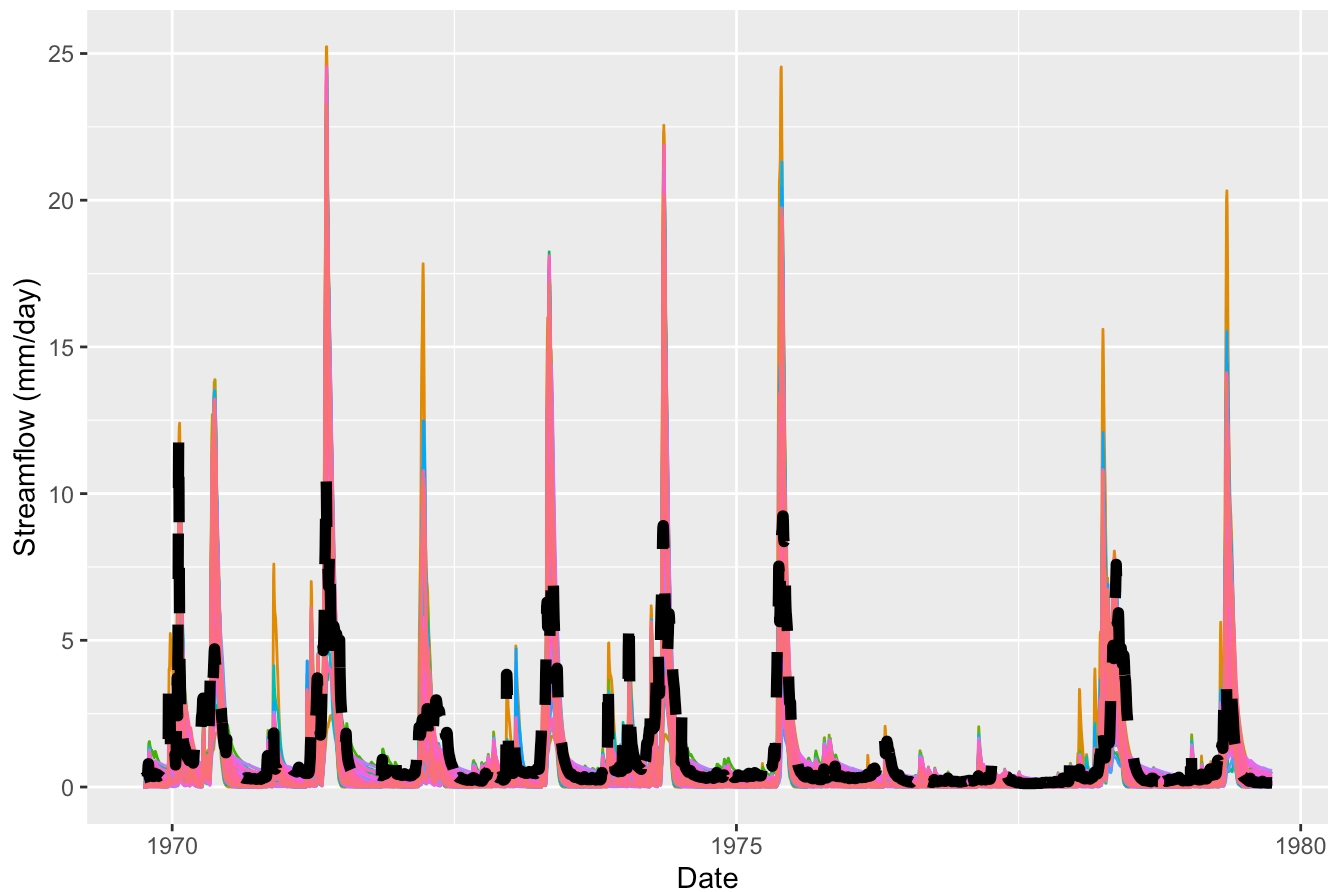
# Transform data for analysis
msagel <- msage %>% pivot_longer(cols =!c(date, month, year, day, wy), names_to = "run",
values_to = "flow")

# Split the data into pre and post calibration years; pre_calibration is 1970 to 1979; p
ost is 1980 to 1989
pre_calibration <- subset(msage, wy >= 1970 & wy <= 1979) #subset for year between 1970
and 1979 #training dataset
post_calibration <- subset(msage, wy >= 1980 & wy <= 1989) #subset for year between 1980
and 1989 #validation dataset

##check out the pre_calibration graph
pre_calibration_graph=ggplot(subset(msagel, wy >= 1970 & wy <= 1979), aes(as.Date(date),
flow, col=run))+geom_line()+theme(legend.position = "none")

# lets add observed streamflow
pre_calibration_graph+geom_line(data=subset(sager, wy >= 1970 & wy <= 1979), aes(as.Dat
e(date), obs), size=2, col="black", linetype=2)+ labs(title = "Pre Calibration of Sagehe
n Creek Streamflow from 1970 - 1979", x = "Date", y = "Streamflow (mm/day)")
```

Pre Calibration of Sagehen Creek Streamflow from 1970 - 1979



```
# using our high flow statistics use apply to compute for all the data using the updated
flow metrics routing that also computed combined metrics
```

```
res = pre_calibration %>% select(!c("date","month","year","day","wy")) %>% map_df(combined_function, o=sager$obs)
```

```
resl = res %>% pivot_longer(cols = starts_with("V"), names_to="run", values_to="value")
```

Find the best and worst parameter set, given your performance metric

```
# extract best and worst runs
```

```
best_run <- resl[which.max(resl$value), ]
```

```
worst_run <- resl[which.min(resl$value), ]
```

```
paste("Best run V_num:", best_run$run, " Best run value:", best_run$value)
```

```
## [1] "Best run V_num: V127 Best run value: 0.301229770688417"
```

```
paste("Worst run V_num:", worst_run$run, " Worst run value:", worst_run$value)
```

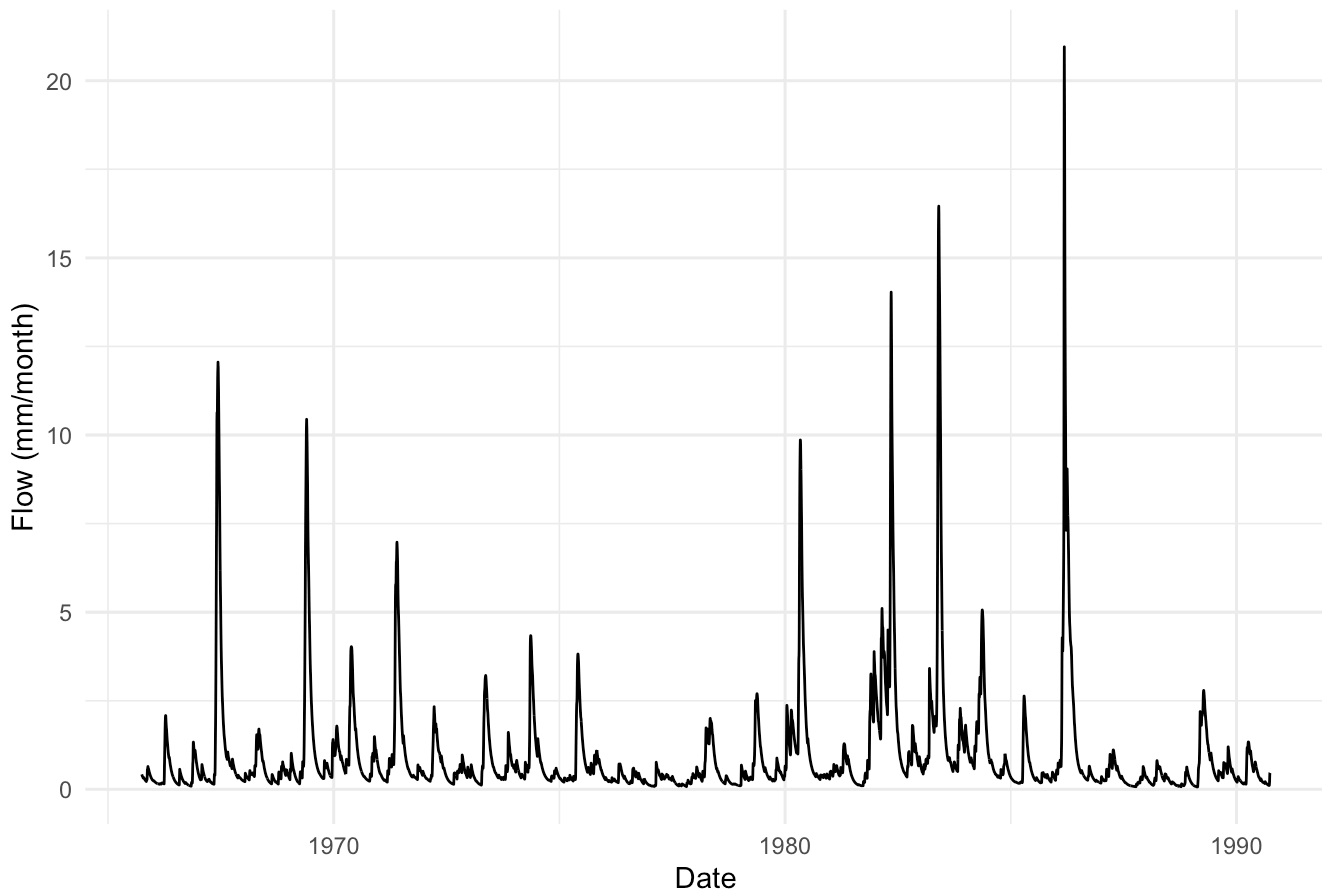
```
## [1] "Worst run V_num: V147 Worst run value: 0.282159842289503"
```

validation step – run the model with these parameters from V127

#the best parameter set for this performance metric was V130;

```
ggplot(subset(msagel, run == "V127"), aes(x= date, flow)) +  
  geom_line() +  
  labs(x = "Date", y = "Flow (mm/month)", title = "Monthly Aggregate Maximum Stream Flow  
for Best Model Run (V127)") + theme_minimal()
```

Monthly Aggregate Maximum Stream Flow for Best Model Run (V127)



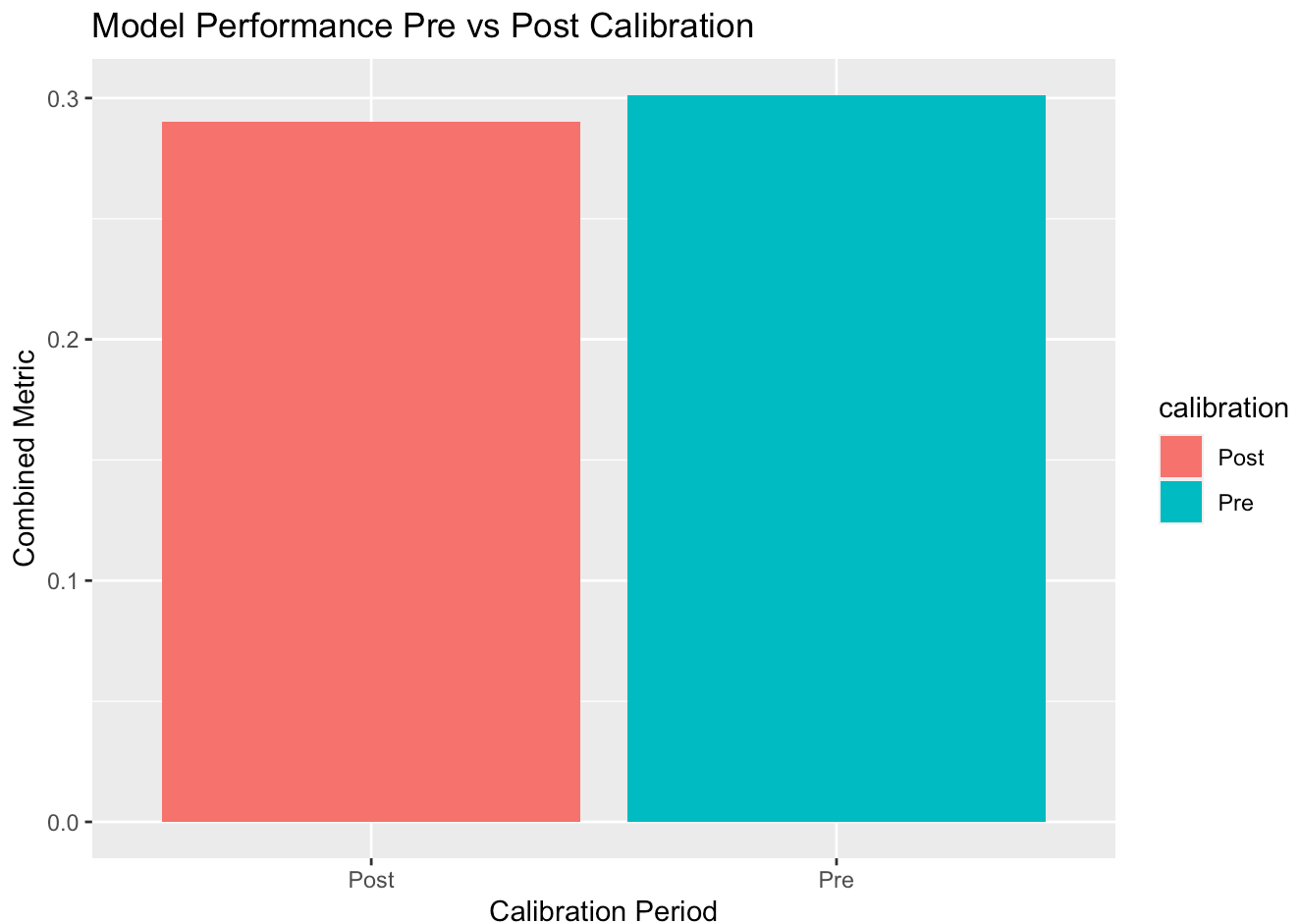
Compute and plot how the performance of the model using the best parameter set changed in pre and post calibration periods


```
#performance for pre calibration period
#performance for post calibration period

# Compute combined metric for pre and post calibration for the best run
pre_combined_func <- combined_function(pre_calibration[["V127"]], sager$obs)
post_combined_func <- combined_function(post_calibration[["V127"]], sager$obs)

# Prep plot
both_combine <- data.frame(calibration = c("Pre", "Post"), combined = c(pre_combined_func, post_combined_func))

# Plot changes in combined metrics, pre & post calibration
ggplot(both_combine, aes(x = calibration, y = combined , fill = calibration)) + geom_bar(
  stat = "identity", position = "dodge") + labs(title = "Model Performance Pre vs Post Ca
  libration", x = "Calibration Period", y = "Combined Metric")
```



Write 2-3 sentences to explain your metric design and comment on model performance based on your metric

All three metrics chosen to create the combined metric have different strengths in evaluating model performance, including errors within the model and model efficiency. We chose to weight nse, rmse, and relerr equally (1/3, 1/3, ~1/3). Based on the above results, the model performed slightly better for the Pre-Calibration data (~0.29) compared to the Post-Calibration data (~0.3). This makes sense, as we noticed that the post calibration years had a lot more variation in their data than pre-calibration years. This may require further adjustments of the model to improve performance Post-Calibration results