Assignment 7: Calibration Part 1

Taylor Cook

2025-06-04

```r


library(tidyverse)
```

Develop another metric of performance (different from in-class example)

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(here)
```

```
## here() starts at /Users/taylorcook/Bren MESM/ESM 232 Env Modeling/ESM232_EnvModeling
```

```r
library(sensitivity)
```

```
## Registered S3 method overwritten by 'sensitivity':
##   method    from
##   print.src dplyr
##
## Attaching package: 'sensitivity'
##
## The following object is masked from 'package:dplyr':
##
##     src
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
msage = readRDS(here("data/msage.RDS"))

msagel = msage %>% gather(key="run",value="str", -date, -month, -day, -year, -wy,-obs)

source(here("R/combined_rmse.R"))
compute_all_metrics
```

```
## function (m, o, month)
## {
##     flow_data = cbind.data.frame(m, o, month)
##     flow_monthly <- flow_data %>% group_by(month) %>% summarize(model = sum(m),
##         obs = sum(o))
##     rmse <- sqrt(mean((flow_monthly$obs - flow_monthly$model)^2))
##     rmse_normal <- rmse/max(flow_monthly$obs)
##     cor <- cor(m, o)
##     combined_metric <- cor - rmse_normal
##     return(list(rmse_normal = rmse_normal, cor = cor, combined_metric = combined_metric))
## }
```

```r
# Apply compute_all_metrics() to each run and extract each metric from the returned list
results <- msagel %>%
  group_by(run) %>%
  summarise(
    combined_metric = compute_all_metrics(str, obs, month)$combined_metric,
    rmse_normal = compute_all_metrics(str, obs, month)$rmse_normal,
    cor = compute_all_metrics(str, obs, month)$cor
  )
```

Apply to all runs in the dataset

```r
# Find the run with the highest combined metric
# use max because high correlation and low RMSE is good
best_run <- results %>%
  filter(combined_metric == max(combined_metric, na.rm = TRUE)) %>%
  select(run, combined_metric)

print(best_run)
```

Find the “best” parameter set based on your metric

```
## # A tibble: 1 x 2
##   run   combined_metric
##   <chr>           <dbl>
## 1 V12             0.780
```

```r
# Define a threshold for acceptability based on the combined metric
quantile(results$combined_metric, 0.9)
```

Decide on a threshold for acceptability

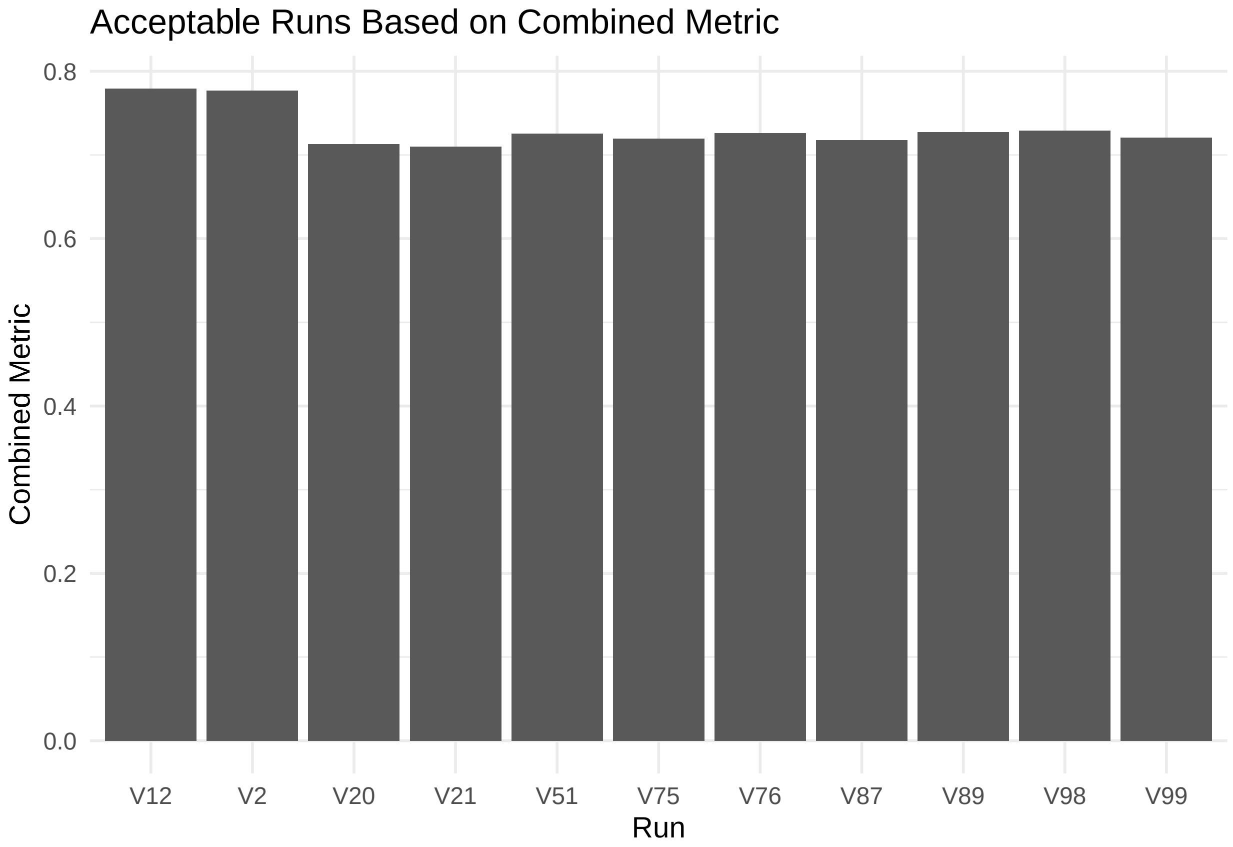
```
##       90%
## 0.7100555
```

```
threshold <- 0.71
acceptable_runs <- results %>%
  filter(combined_metric >= threshold)
print(acceptable_runs)
```

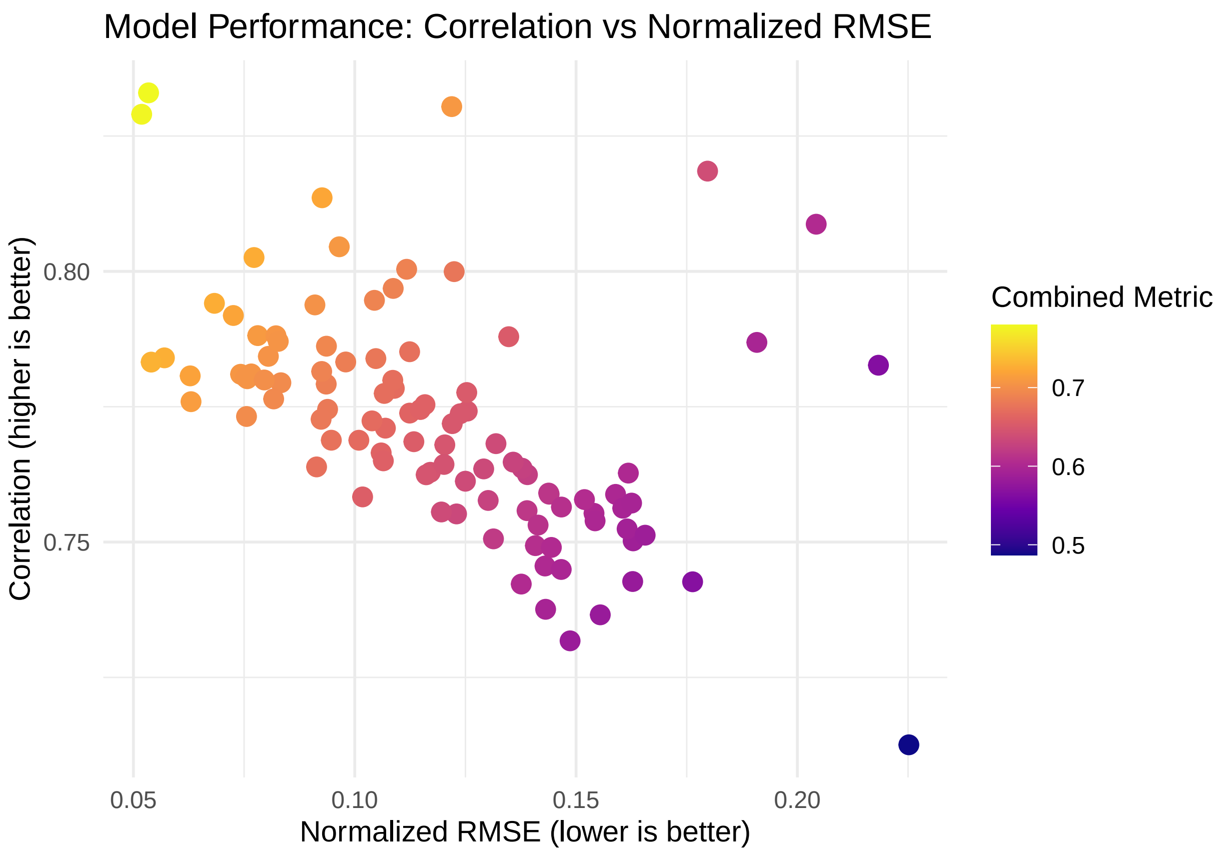
```
## # A tibble: 11 x 4
##    run   combined_metric rmse_normal   cor
##    <chr>           <dbl>       <dbl> <dbl>
##  1 V12             0.780      0.0534 0.833
##  2 V2              0.777      0.0519 0.829
##  3 V20             0.713      0.0630 0.776
##  4 V21             0.710      0.0781 0.788
##  5 V51             0.725      0.0772 0.803
##  6 V75             0.719      0.0726 0.792
##  7 V76             0.726      0.0683 0.794
##  8 V87             0.718      0.0628 0.781
##  9 V89             0.727      0.0570 0.784
## 10 V98             0.729      0.0540 0.783
## 11 V99             0.721      0.0926 0.814
```

plot results for all acceptable parameters and *choose a plot that shows model performance in a way that fits your metric*

```
# Plot the results for all acceptable runs
ggplot(acceptable_runs, aes(x = run, y = combined_metric)) +
  geom_bar(stat = "identity") +
  labs(title = "Acceptable Runs Based on Combined Metric",
       x = "Run",
       y = "Combined Metric") +
  theme_minimal()
```



```
#Chose scatter plot to show model perforamnce in a way that fits my metric
ggplot(results, aes(x = rmse_normal, y = cor, color = combined_metric)) +
  geom_point(size = 3) +
  scale_color_viridis_c(option = "C", direction = 1) +
  labs(title = "Model Performance: Correlation vs Normalized RMSE",
       x = "Normalized RMSE (lower is better)",
       y = "Correlation (higher is better)",
       color = "Combined Metric") +
  theme_minimal()
```


Model Performance: Correlation vs Normalized RMSE
Correlation (higher is better)
0.80
0.75
0.05
0.10
0.15
0.20
Normalized RMSE (lower is better)
Combined Metric
0.7
0.6
0.5