CALIBRATION

CALIBRATION AND OPTIMIZATION

- Generate parameter sets (e.g., LHS)
- Compute metrics for each
- Pick the “best” parameter set?

EQUIFINALITY

- Many parameter sets yield similar performance
- Limits confidence in “best” calibration

Equifinality

ISSUE WITH PARAMETER SELECTION |

Parameter selection will be effected by:

- calibration period
- observation/measurement error
- poor identifiablity (equifinality)

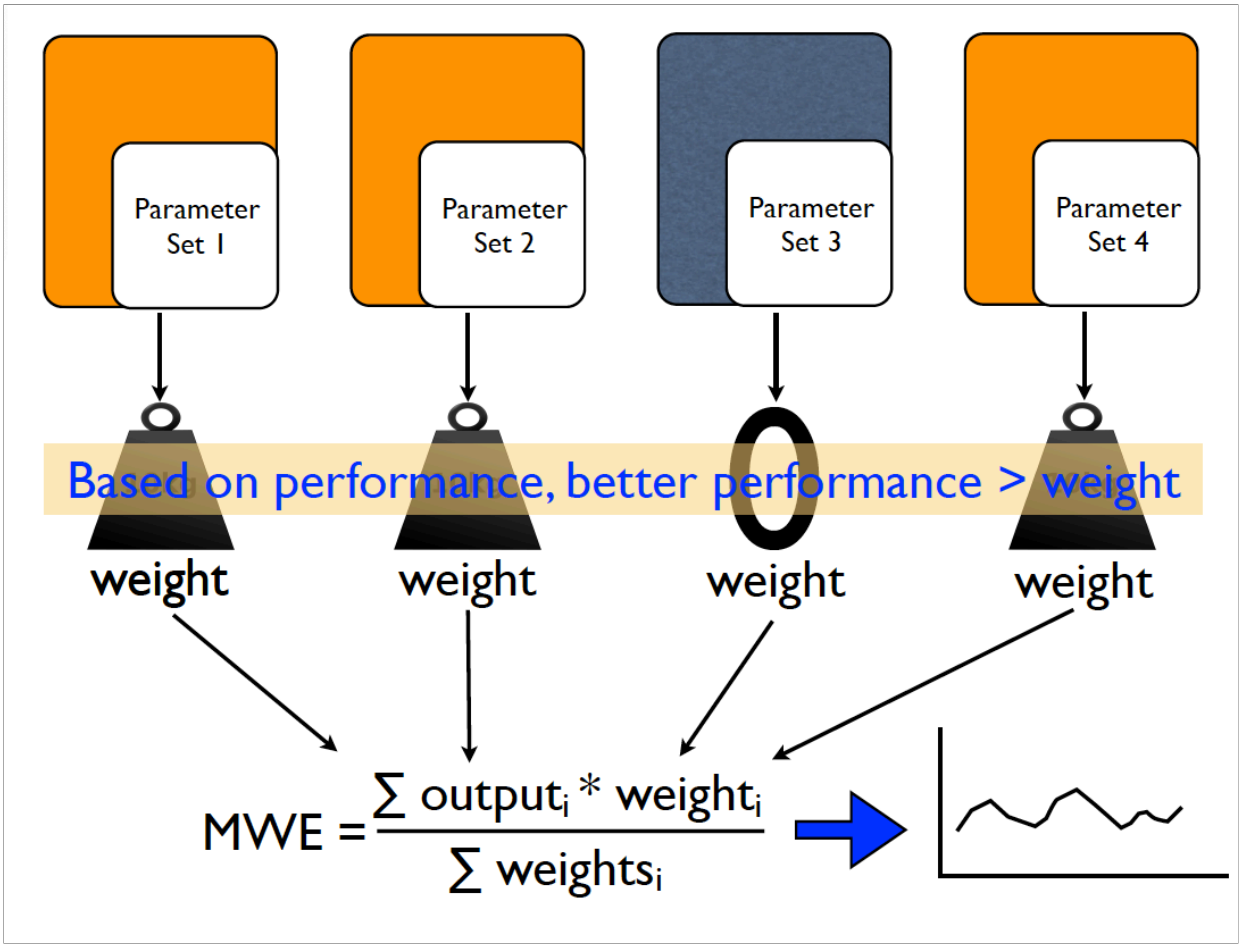
How can we be more robust in parameter selection?

CALIBRATION - AN ALTERNATIVE

- Generate parameter sets (e.g., LHS)
- Compute metrics for each
- Keep all acceptable parameter sets (or sample across them)
- Show uncertainty in estimates due to variation across these parameter sets
- If you need a single estimate
  - use an ensemble approach
  - average estimates from all parameters
  - weight estimate from each parameter by performance

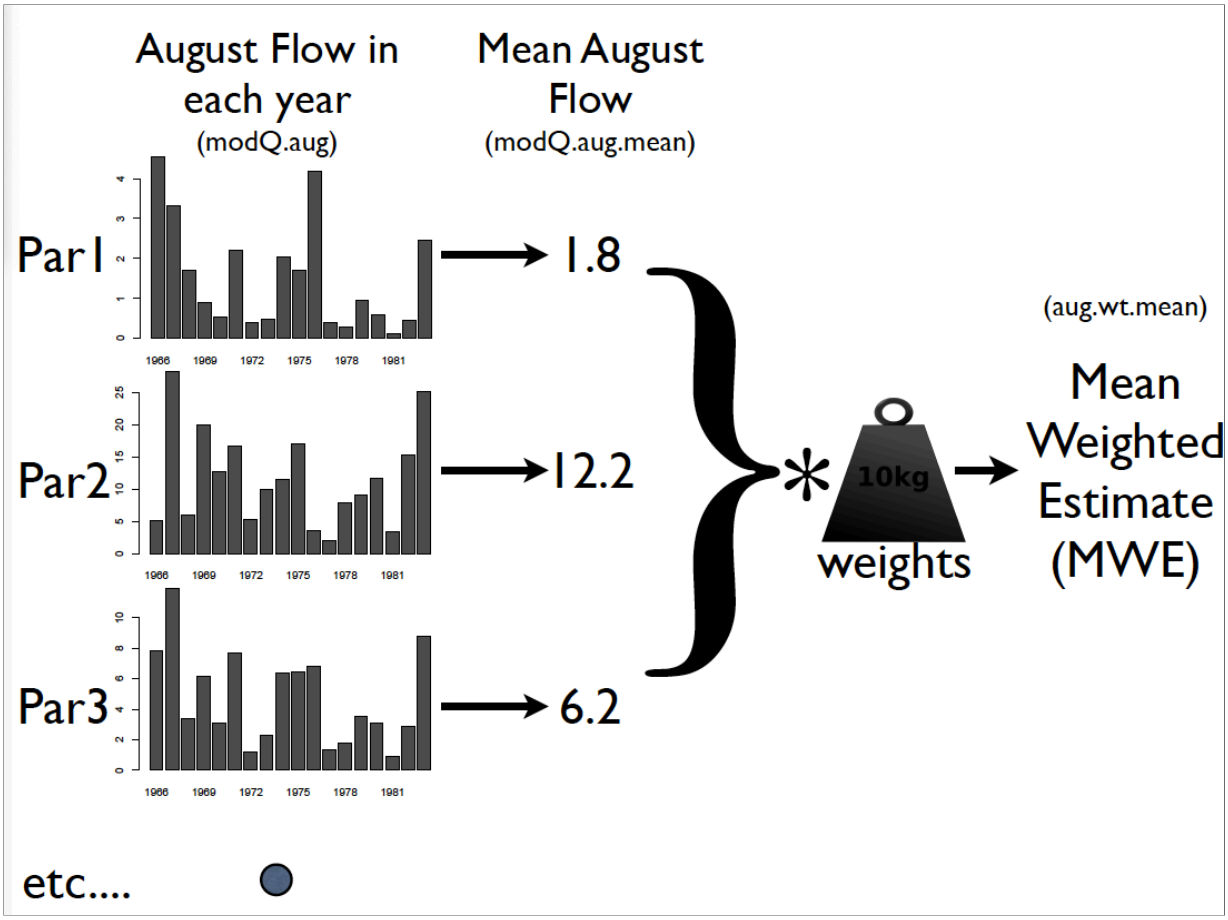
USING PERFORMANCE TO WEIGHT ESTIMATE

Mean weighted estimate (MWE) - Weighted average of outputs by performance - Produces MWE (Mean Weighted Estimate)

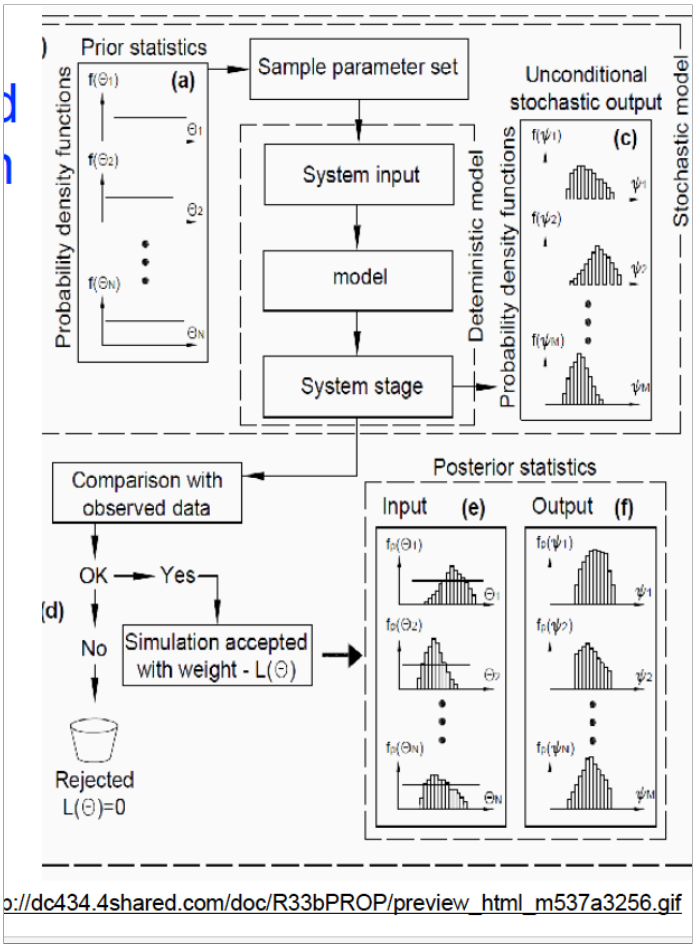


Equifinality

EXAMPLE


August Flow in each year
(modQ.aug)
Mean August Flow
(modQ.aug.mean)
Par1
1.8
Par2
12.2
Par3
6.2
1966
1969
1972
1975
1978
1981
10kg
weights
(aug.wt.mean)
Mean Weighted Estimate (MWE)
etc....

BAYESIAN FRAMEWORK



Bayesiany

K. Beven and A. Binley, “The future of distributed models: model calibration and uncertainty prediction,” Hydrological Processes, vol. 6, no. 3, pp. 279–298, 1992.

APPROACH SUMMARY

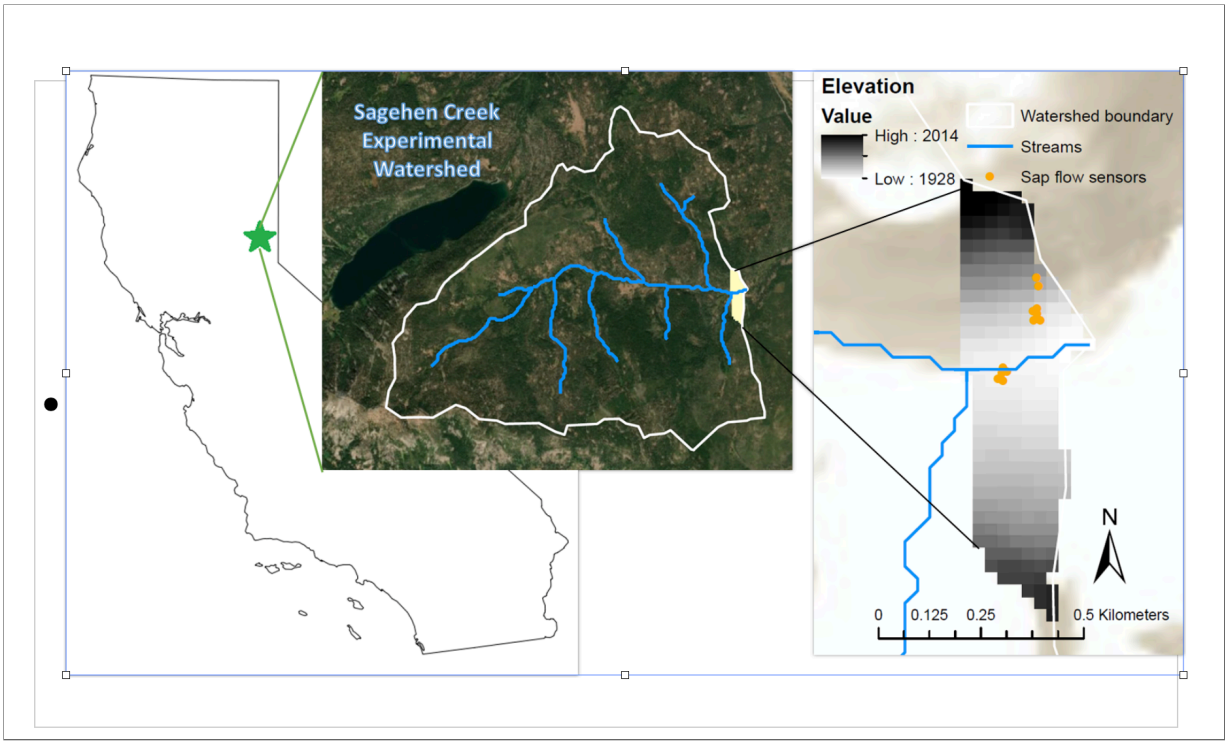
- Generate parameter sets (e.g., LHS)
- Compute metrics for each
- Reject poor performers
- Retain ALL acceptable sets
- Use ensemble to represent uncertainty

EXAMPLE: PART 1

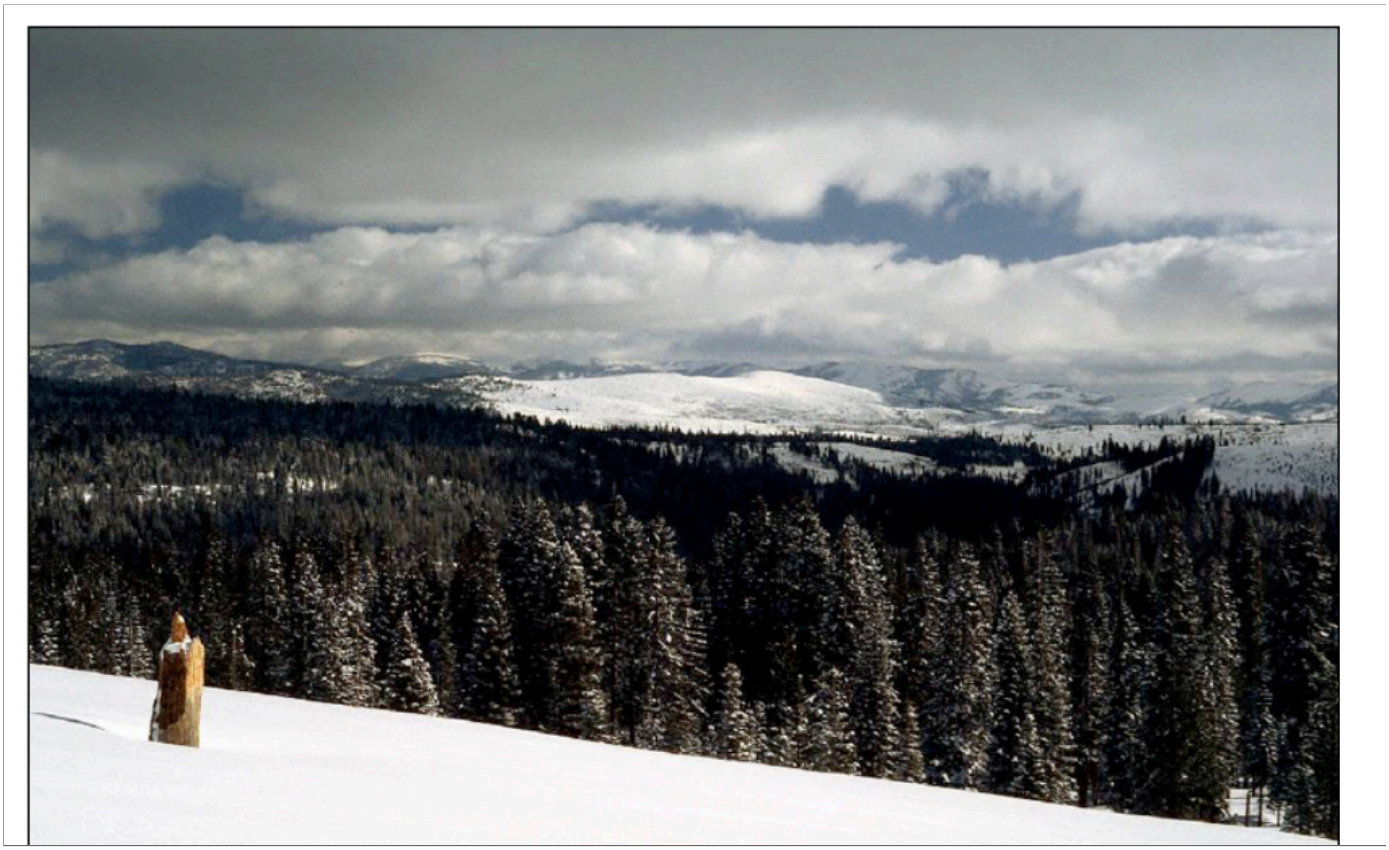
- Why metrics matter
- Equifinality
- R-code (re-usable for off the shelf models)

HYDROLOGIC MODEL EXAMPLE

RHESSYs application to Sagehen Creek in California Sierra

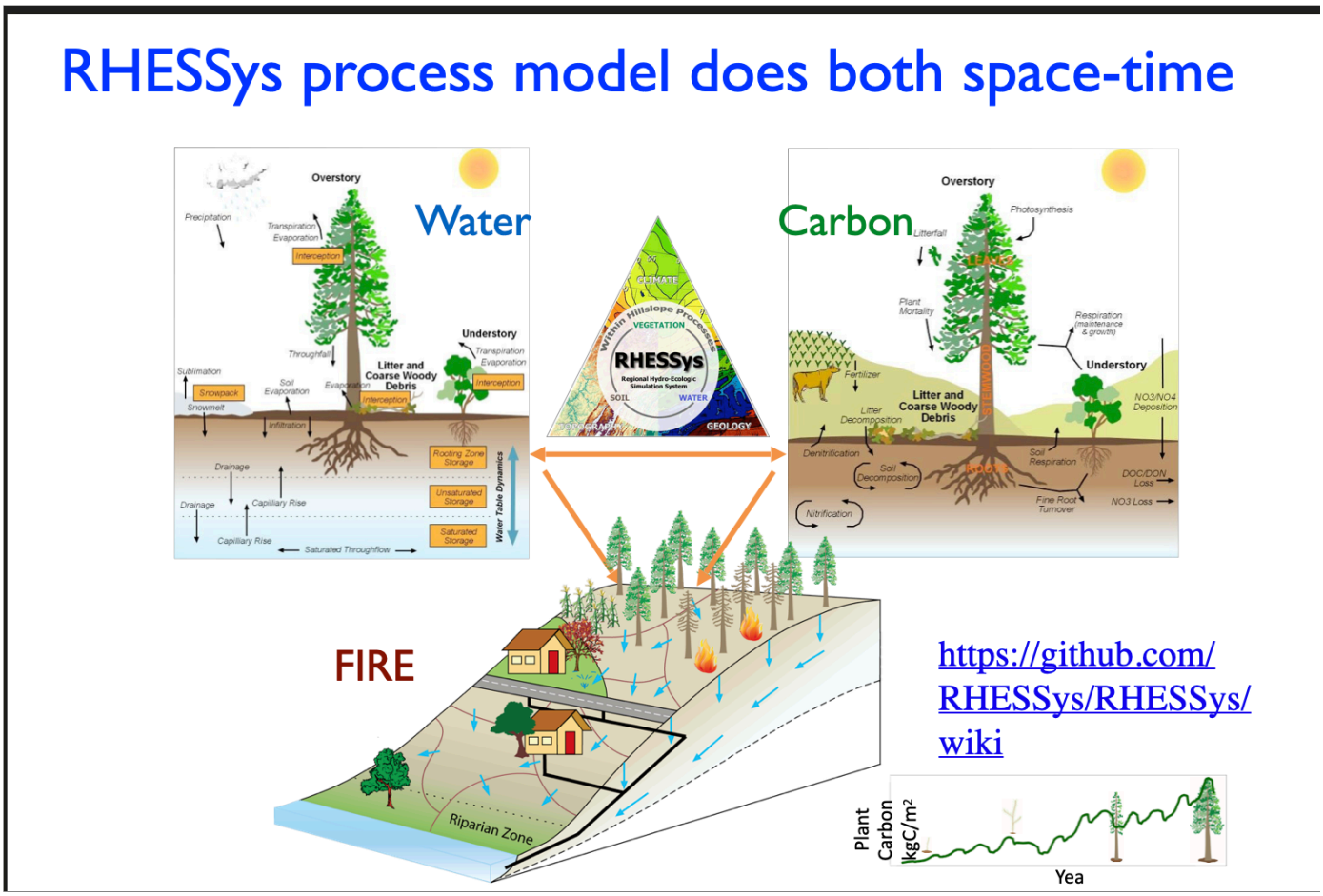


HYDROLOGIC MODEL EXAMPLE



Sagehen

RHESSYS



RHESSys

RHESSYS

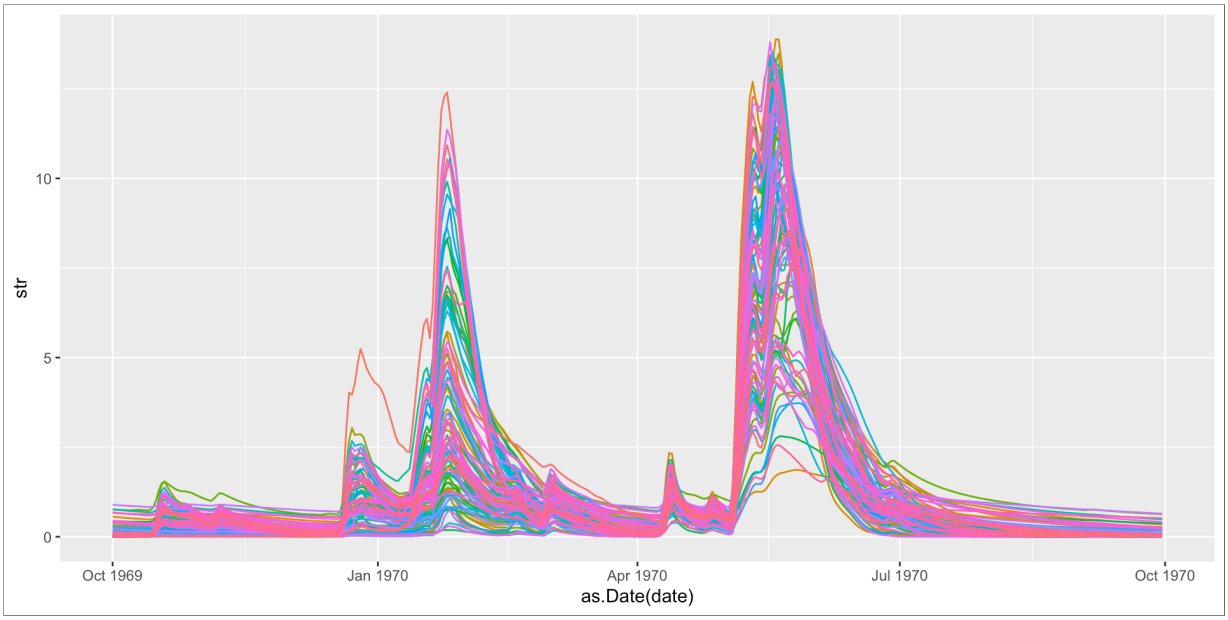
- run for 101 parameter sets

  Each parameter set
- Water year 1966 to 1990
- Streamflow each day
- Table organized each run in a column

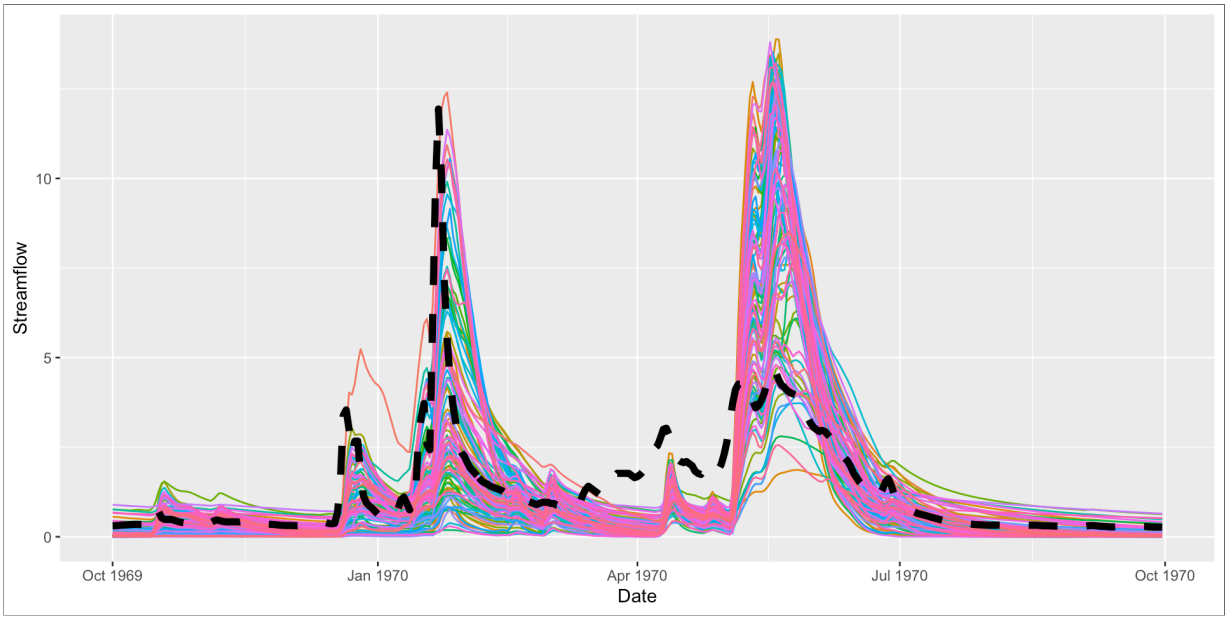
  And we have observed data
- Which parameter sets are acceptable?
- What if we just pick the best one?

LOAD THE DATA

```
library(tidyverse)
library(here)

msage = readRDS(here("Data/msage.RDS"))
#View(msage)

# first rearrange so we can plot all results
msagel = msage %>% gather(key="run",value="str", -date, -month, -da

# now plot
p1=ggplot(subset(msagel, wy == 1970), aes(as.Date(date), str, col=r
p1
```




```
# lets add observed streamflow
p1+geom_line(data=subset(msage, wy == 1970), aes(as.Date(date), obs
```




```
1 # try another year
```

CALIBRATION METRICS FOR EACH PARAMETER SET

- NSE
- Low flow metrics

```
source(here("R/nse.R"))
nse
```

```
function (m, o)
{
    err <- m - o
    meanobs <- mean(o)
    mse <- sum(err * err)
    ovar <- sum((o - meanobs) * (o - meanobs))
    nse <- 1 - mse/ovar
    return(nse)
}
```

```
# compute nse for each parameter set
res = msage %>% select(-date, -month, -day, -year, -wy, -obs ) %>%
head(res)
```

```
        V1         V2         V3         V4         V5         V6
-0.5443680  0.4855281  0.2654724  0.2178853  0.2715622  0.5031371
```

```
summary(res)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.4024 -0.1614  0.1247  0.0404  0.3293  0.6859
```

```
# basic calibration would be picking a single parameters set
best_par = which.max(res)
# lets plot the best parameter set

ggplot(msage, aes(as.Date(date), msage[,best_par]))+geom_line(col="
  geom_line(data=msage, aes(as.Date(date), obs), col="black", linet
```

```
# what would be acceptable?
```

TRY AGAIN WITH LOW FLOW METRICS

a function that generates

- error in estimating minimum annual flow
- correlation in minimum annual flow
- error in estimates of a given month
  - user can pick low flow month
- correlation for that month (each year)

```
source(here("R/compute_lowflowmetrics.R"))

# another example using our low flow statistics
# use apply to compute for all the data
res = msage %>% select(-date, -month, -day, -year, -wy, -obs ) %>%

# extract information from the list
results = as.data.frame(matrix(unlist(res), byrow=T, ncol=4))
colnames(results)=c("annual_min_err","annual_min_cor", "low_month_e
             "low_month_cor")

# interesting to look at range of metrics - could use this to decid
# acceptable values
summary(results)
```

```
annual_min_err    annual_min_cor   low_month_err    low_month_cor
Min.   :-0.2036   Min.   :0.4350   Min.   :-9.857   Min.   :0.1076
1st Qu.:-0.1977   1st Qu.:0.5760   1st Qu.:-7.899   1st Qu.:0.2050
Median :-0.1853   Median :0.6034   Median :-6.044   Median :0.5889
Mean   :-0.1526   Mean   :0.6296   Mean   :-4.434   Mean   :0.5211
3rd Qu.:-0.1471   3rd Qu.:0.7315   3rd Qu.:-3.183   3rd Qu.:0.7762
Max.   : 0.2004   Max.   :0.8192   Max.   :13.004   Max.   :0.8774
```

```
# graph range of performance measures
resultsl = results %>% gather(key="metric",value="value")
ggplot(resultsl, aes(metric, value))+geom_boxplot()+facet_wrap(~met
```

```
# how might you pick best parameter sets
```

ONE OPTION

```
# pick parameter set with greater low flow month correlation

best_par2 = which.max(results$low_month_cor)

# is it the same as what we got with nse
```

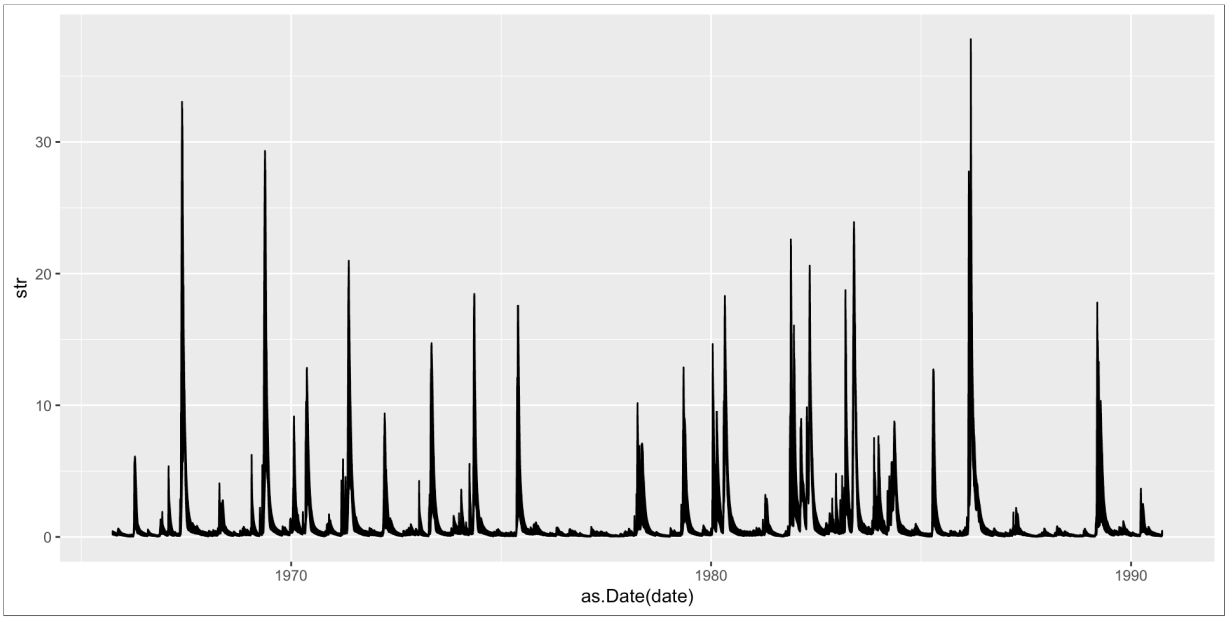
PARAMETER SELECTION

Lets keep all parameters that give reasonable values

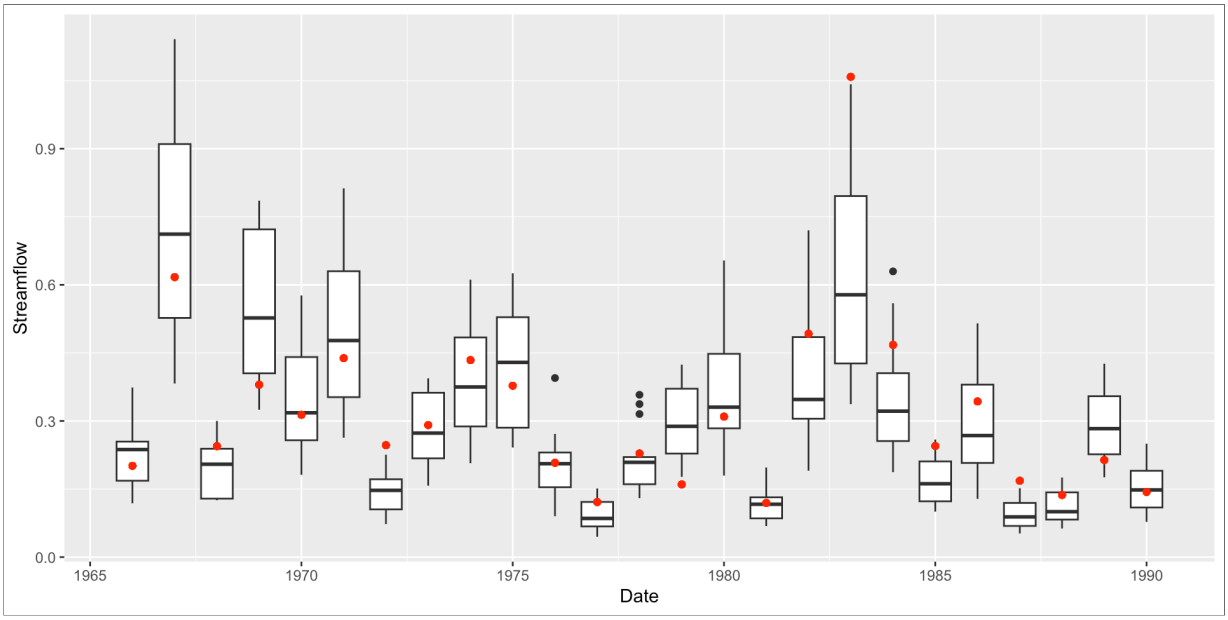
```
# first we need to be able to identify parameter sets in results

ID = msage %>% select(-date, -month, -day, -year, -wy, -obs ) %>%
results$id = ID

# now we can pick only good parameters
accept_par = subset(results, annual_min_cor > 0.7 & low_month_cor >
nrow(accept_par)
```

```
[1] 13
```

```
#  plot these to compare with all parameters
msagel$accept = ifelse(msagel$run %in% accept_par$id, TRUE, FALSE)
ggplot(subset(msagel, accept), aes(as.Date(date), str))+geom_line()
```




```
# or with observed but focus on August flow
msagel_mth = msagel %>% select(-date,-day) %>%  group_by(month, wy,

# just august flow
ggplot(subset(msagel_mth,  month==8 & accept),
       aes(wy, str, group=wy))+geom_boxplot(position="dodge")+
  geom_point(aes(wy, obs), col="red")+labs(y="Streamflow", x="Date"
```




```
mean_august_obs = mean(subset(msagel_mth, month==8)$obs)
ggplot(subset(msagel_mth, month==8), aes(accept, str))+geom_boxplot
  geom_hline(yintercept=mean_august_obs, col="red")+labs(y="Streamf
```