1 Hydraulics

Download daily flow data from USGS.

```
library(hydroTSM)

source("hydrotools.R")  ## functions to plot in ggplot and to convert sets of months to
indices

### Load data

# require(IHA) # package to download from USGS, should just hard
#   code this function
# napa.fl <- read.flow('11458000', start.date=ISOdate(1930,1,1));
#   save(napa.fl, file='napa.Rdat');
# napa.fl<-read.csv('napagauge.csv'); class(napa.fl) <- c('flow',
#   'data.frame');

load("napa.Rdat")  #loads napa.fl, full time series values ca 1930 to 1959 are missing
napa.compact <- na.omit(napa.fl)

d <- napa.compact
clip.date <- as.POSIXct("1950-01-01")
d <- subset(d, date > clip.date)  #remove observations older than 1950

d.z <- zooreg(d$discharge, order.by = d$date, frequency = 1)  #convert to zoo
```

Some examples of aggregating with specified functions and extracting only one month.

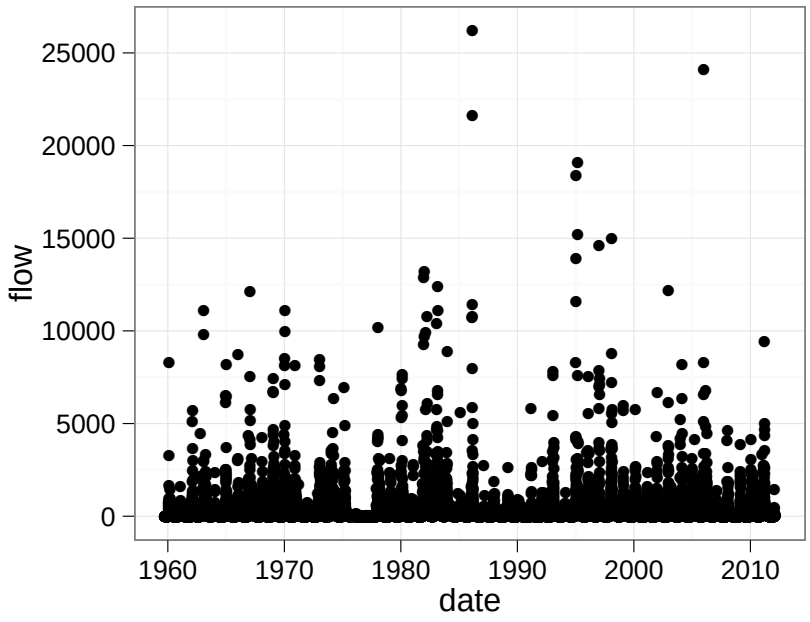
```
monthly.avg <- daily2monthly(d.z, FUN = mean)
monthly.range <- daily2monthly(d.z, FUN = function(x, na.rm = TRUE) {
    rng <- range(x, na.rm = TRUE)
    return(rng[2] - rng[1])
})

## subset oct from monthly avg
d.m <- monthly.avg
oct.numind <- Month2Index("Oct", d.m)
d.m[oct.numind]
```

To remember: OCTOBER is start of water year.
R commands for working with timeseries

- `month.abb`: abbreviated months in order
- `month.name`: full —"—
- `monthlyfunction( zoo, FUN)`: applies FUN to each monthly subseries (min/max, etc)

Plot the full hydograph.

```
AXISSIZE <- 14
LABELSIZE <- 22

hyd1 <- gHyd(d.z) + geom_smooth() + geom_point() + ylab("flow") +
    theme_bw()
hyd1
```




```
## opts(axis.text.x=theme_text(size=AXISSIZE),
#   axis.text.y=theme_text(size=AXISSIZE),
#   axis.title.x=theme_text(size=LABELSIZE),
#   axis.title.y=theme_text(size=LABELSIZE, angle=90))
```

1.1 Examples: aggregation and sliding window

```
## sliding window for minimum in 28 days
s.min <- rollapply(d.z, width = 28, FUN = min, align = "right",
    fill = NA)
g1 <- gHyd(s.min) + geom_point() + ylab("min flow in 28 days")

## aggregation to annual

### via summation
s.ann <- daily2annual(s.min, sum)
g2 <- gHyd(s.ann) + geom_line() + geom_point() + ylab("annual index: cumulative minima")
+
    theme_bw()


### via maximum in each year
g3 <- gHyd(daily2annual(s.min, max)) + geom_line() + geom_point() +
    ylab("annual index: max min") + theme_bw()

grid.arrange(g2, g3, ncol = 2)  ## require(gridExtra)
```

2 Relating flow timing and duration to population needs

With the hydrograph in hand, we develop metrics flows that meet population needs in each year by relating natural history of the species involved to (1) seasonality, and (2) duration of flows needed to support population processes. In the next section, we describe how these flows will be related to area of inundation by using a hydraulic model.

2.1 Within-year flow needs: "time" habitat

We consider population needs within a year by first filtering by season, then by duration appropriate to population needs.

As an example, for splitail spawning season we consider only the late winter and spring months, and look for flows durations long enough to support spawning.

```
spawning.season <- c("Feb", "Mar", "Apr")
duration <- 15

example.year <- 1998
min.date <- as.POSIXct("1950-01-01")
```

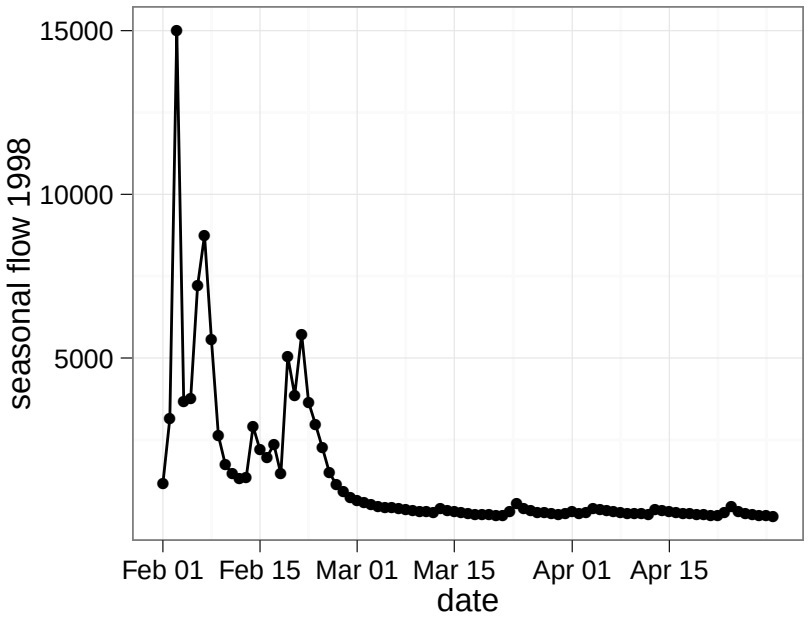
Example year

As an example, we look at applying these criteria to the flow from one year, 1998. First, we extract only the spring months from the full hydrograph.

```
# subset jan,feb,mar from full
spr.ind <- Month2Index(spawning.season, d.z)
d.spr <- d.z[spr.ind]

#d.spr.sum <- daily2annual(d.spr, sum)
#d.spr.min <- daily2annual(d.spr, min)
#gHyd(d.spr.min)+geom_line()+geom_point()+ylab('minimum spring
#   flow')+theme_bw()
```

Then, we extract the seasonal flow from our example year (1998):

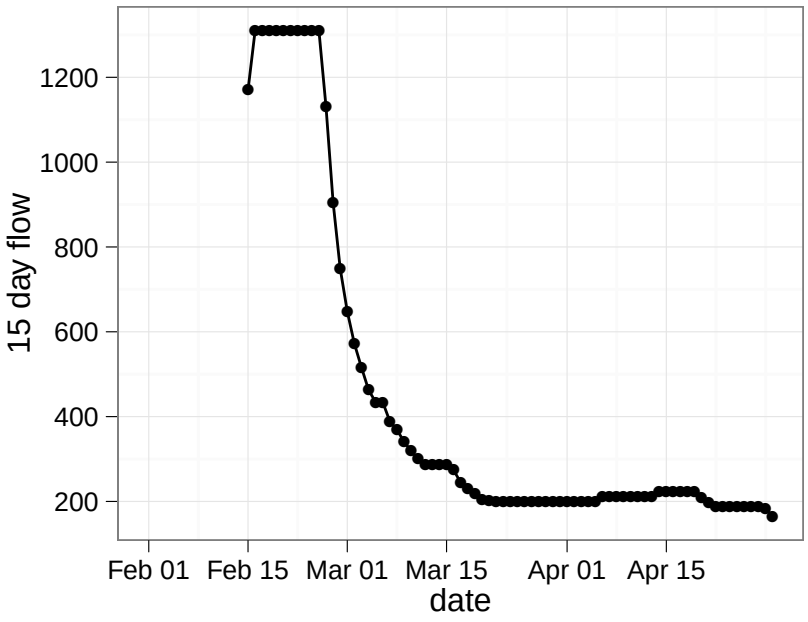
```
AXISSMALL <- 10

ind.1yr <- as.POSIXlt(index(d.spr))$year == (example.year -
    1900)
d.season.1yr <- d.spr[ind.1yr]

ex1 <- gHyd(d.season.1yr) + geom_line() + geom_point() +
    ylab(paste("seasonal flow", example.year)) + theme_bw()
ex1
```




```
## +opts(axis.text.x=theme_text(size=AXISSMALL),
#   axis.text.y=theme_text(size=AXISSIZE),
#   axis.title.x=theme_text(size=LABELSIZE),
#   axis.title.y=theme_text(size=LABELSIZE, angle=90))
```

With the seasonal flow in hand, we filter for flows that lasted longer than the minimum duration (15 days). This yields, for each date in the season, a value of flow meeting the population needs for spawning:

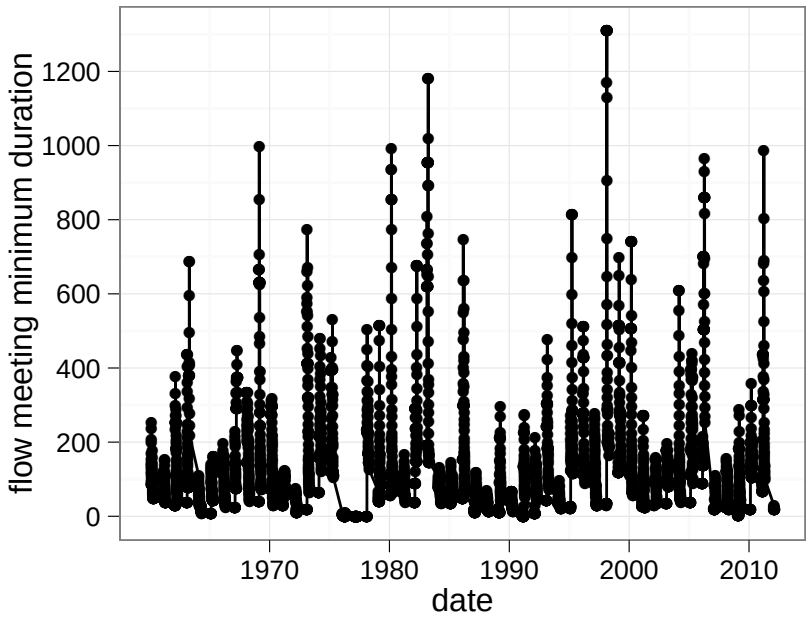
```
d.season.1yr <- rollapply(d.season.1yr, width = duration,
    FUN = min, align = "right", fill = NA)  # sliding window for min in last 15 days

ex2 <- gHyd(d.season.1yr)
ex2 <- ex2 + geom_line() + geom_point() + ylab(paste(duration,
    "day flow")) + theme_bw()
## ex2 <- ex2+opts(axis.text.x=theme_text(size=AXISSMALL),
#   axis.text.y=theme_text(size=AXISSIZE),
#   axis.title.x=theme_text(size=LABELSIZE),
#   axis.title.y=theme_text(size=LABELSIZE, angle=90))
ex2

## Warning message: Removed 14 rows containing missing values (geom_path).
## Warning message: Removed 14 rows containing missing values (geom_point).
```



All years in record

This same calculation can be applied to the entire hydrograph, yielding a graph of flows that meet the population needs within the appropriate season of every year. Note the considerable variation between years.

```
d.season.all <- rollapply(d.spr, width = duration, FUN = min,
    align = "right", fill = NA)  # sliding window for min
d1 <- gHyd(d.season.all) + geom_line() + geom_point() + ylab("flow meeting minimum
duration") +
    theme_bw()
d1

## Warning message: Removed 14 rows containing missing values (geom_path).
## Warning message: Removed 14 rows containing missing values (geom_point).
```




```
## +opts(axis.text.x=theme_text(size=AXISSIZE),
#   axis.text.y=theme_text(size=AXISSIZE),
#   axis.title.x=theme_text(size=LABELSIZE),
#   axis.title.y=theme_text(size=LABELSIZE, angle=90))
```

2.2 Relating flow to area inundated: "space" habitat

Flow-area

Relationships between river flows and are inundated are a critical link between population-appropriate flows and habitat in each year. We will use hyrdrological models run at a variety of flows to produce the relationship. First, we need to analyze the hydrograph to determine the range of flows that we must consider:

```
source("process.R")  ## functions to convert hydrograph to range of values

## load('napa.Rdat') #loads napa.fl, full time series values ca
#   1930 to 1959 are missing
## napa.compact <- na.omit(napa.fl)
## d <- napa.compact

rng <- rangeGauge(d, date.min = min.date, density = 10)
m1 <- 0.5
m2 <- 1.5
```

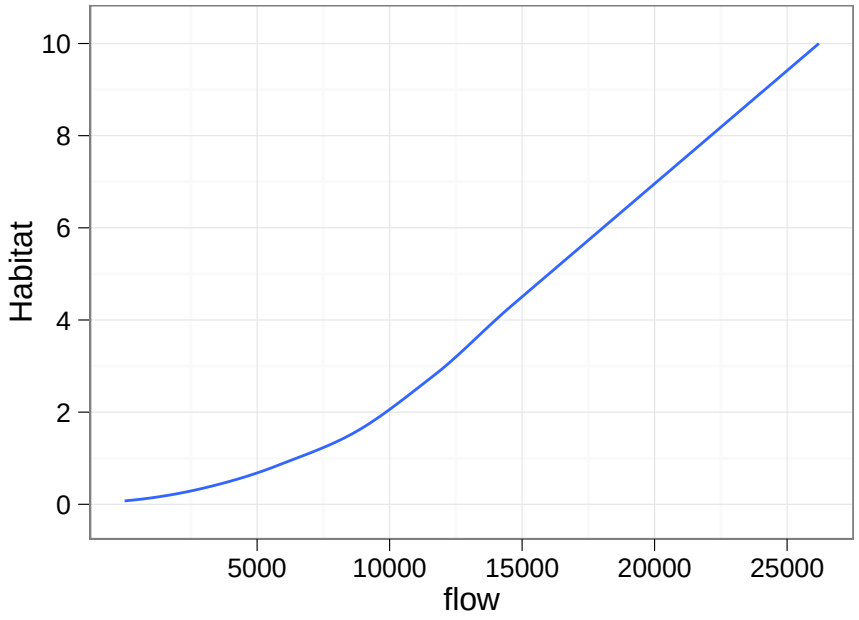
For now, without hydrological models, we must work with simulated examples. Below is a simulation of two qualitatively different area-flow relationships:

```
MUL <- 10  ## multiplier to get similar scale results
fa.df <- data.frame(flow = rng, area.cvx = c(0, m1 * rng[2:4],
    m2 * rng[5:length(rng)] + (m1 - m2) * rng[4]), area.cve = pmax(0,
```

```
    log(rng)))
# x1*(m1 - m2) = b
fa.df[, 2] <- fa.df[, 2]/max(fa.df[, 2]) * MUL
fa.df[, 3] <- fa.df[, 3]/max(fa.df[, 3])
```

A convex relationship might correspond to an incised channel with a large floodplain but one which the river only flows onto at high flows:

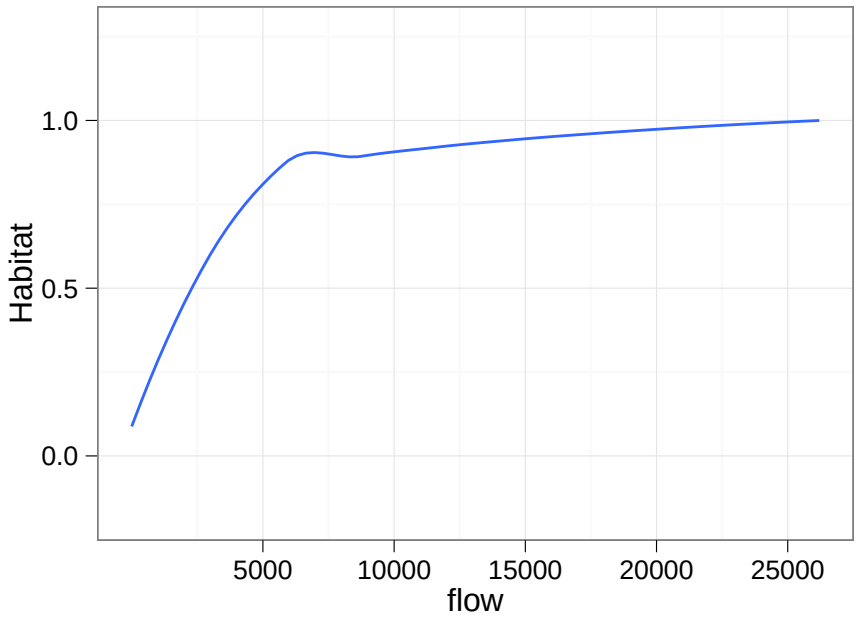
```
c1 <- ggplot(fa.df) + geom_smooth(aes(flow, area.cvx), alpha = 0) +
    theme_bw() + ylab("Habitat")
c1

## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to ch
```




```
## +opts(axis.text.x=theme_text(size=AXISSIZE),
#   axis.text.y=theme_text(size=AXISSIZE),
#   axis.title.x=theme_text(size=LABELSIZE),
#   axis.title.y=theme_text(size=LABELSIZE, angle=90))

lo.cvx <- loess(area.cvx ~ flow, data = fa.df)
Convex <- function(x, na.rm = TRUE) {
    predict(lo.cvx, data.frame(flow = na.omit(x)))
}
```

A concave relationship might correspond to an channel with a low, bench floodplain that is fortified, where even low flows activate the floodplain:

```
v1 <- ggplot(fa.df) + geom_smooth(aes(flow, area.cve), alpha = 0) +
    theme_bw() + ylab("Habitat")
v1

## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to ch
```




```
## +opts(axis.text.x=theme_text(size=AXISSIZE),
#   axis.text.y=theme_text(size=AXISSIZE),
#   axis.title.x=theme_text(size=LABELSIZE),
#   axis.title.y=theme_text(size=LABELSIZE, angle=90))

lo.cve <- loess(area.cve ~ flow, data = fa.df)
Concave <- function(x, na.rm = TRUE) {
    predict(lo.cve, data.frame(flow = na.omit(x)))
}
```

2.3 Putting the pieces together: flow timing, duration, and area = habitat

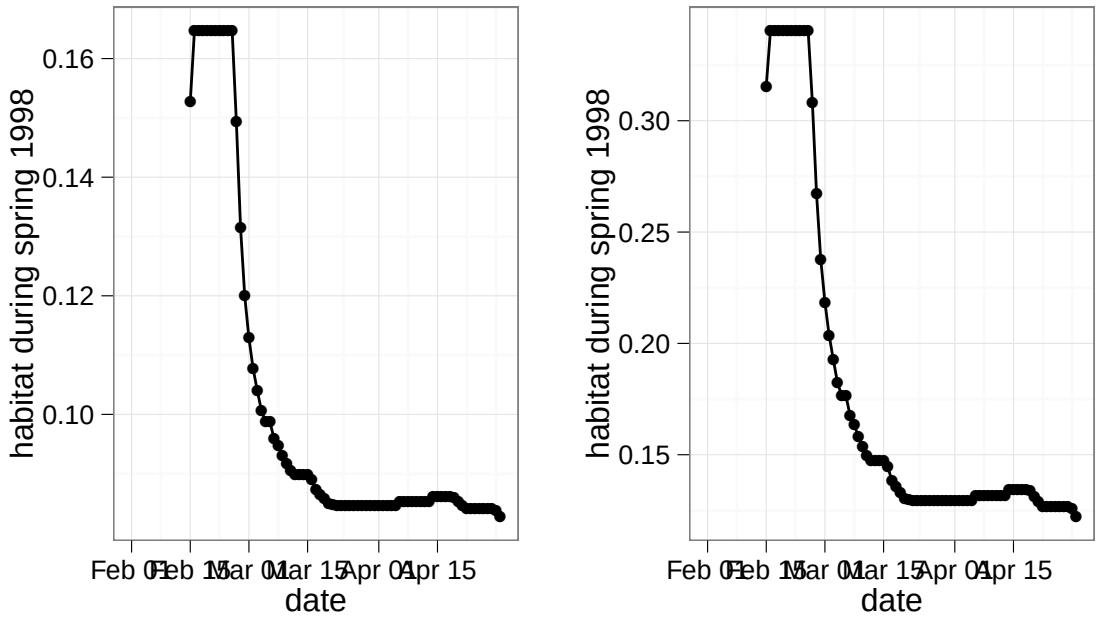
By taking the flows that occur with appropriate timing and duration, as determined in Section **??**, and feeding these into the flow-area relationships described in the previous section, we can determine the amount of habitat for each day in the season.

Example year

Returning to our example year, 1998, we see a difference between an incised (Convex) and reconnected (Concave) floodplain:

```
## d.season.1yr is output of filtering for time/duration above
p1 <- Plot.flow.area(d.season.1yr, Convex) + ylab(paste("habitat during spring",
    example.year)) + theme_bw()
## p1 <- p1 + opts(axis.text.x=theme_text(size=AXISSIZE),
##          axis.text.y=theme_text(size=AXISSIZE),
##          axis.title.x=theme_text(size=LABELSIZE),
## axis.title.y=theme_text(size=LABELSIZE,
#   angle=90))+ylim(c(0,0.4))

p2 <- Plot.flow.area(d.season.1yr, Concave) + ylab(paste("habitat during spring",
```

```
    example.year)) + theme_bw()
grid.arrange(p1, p2, ncol = 2)

## Warning message: Removed 14 rows containing missing values (geom_path).
## Warning message: Removed 14 rows containing missing values (geom_point).
## Warning message: Removed 14 rows containing missing values (geom_path).
## Warning message: Removed 14 rows containing missing values (geom_point).
```



2.4 From habitat to a yearly vital rate

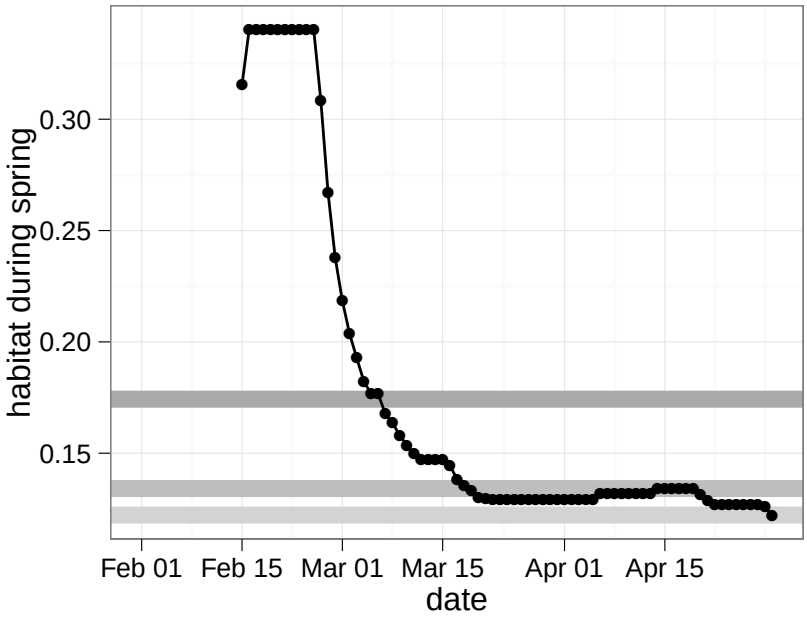
To obtain a *yearly* metric, we can aggregate over the habitat within the year in a way that makes biological sense. For example

- spawning in splittail, which can spawn repeatedly in good years: *sum* of habitat over the year
- enhanced survival of a cohort: *minimum* of habitat during the year because only ...
- enhanced growth of a cohort: *median* of habitat because a number of different individuals can access the habitat at different times and benefit

.

```
d.area.1yr <- Flow.area.ts(d.season.1yr, Concave)

p3 <- gHyd(d.area.1yr) + geom_hline(yintercept = min(d.area.1yr,
    na.rm = TRUE), col = "lightgray", size = 3) + geom_hline(yintercept =
median(d.area.1yr,
    na.rm = TRUE), col = "gray", size = 3) + geom_hline(yintercept = mean(d.area.1yr,
    na.rm = TRUE), col = "darkgray", size = 3)
p3 <- p3 + geom_point() + geom_line() + ylab("habitat during spring") +
    theme_bw()
## p3 <- p3 + opts(axis.text.x=theme_text(size=AXISSIZE),
##          axis.text.y=theme_text(size=AXISSIZE),
```

```
##          axis.title.x=theme_text(size=LABELSIZE),
## axis.title.y=theme_text(size=LABELSIZE,
#   angle=90))+ylim(c(0,0.4))
p3

## Warning message: Removed 14 rows containing missing values (geom_point).
## Warning message: Removed 14 rows containing missing values (geom_path).
```




```
sum(d.area.1yr, na.rm = TRUE)  ## sum of habitat in example.year

## [1] 13.07
```

The plot illustrates habitat in an example year, along with minimum, median, and mean flow. The sum (≈ 13) is computed above.

In this example, we are working with splittail spawning, so we use a sum.

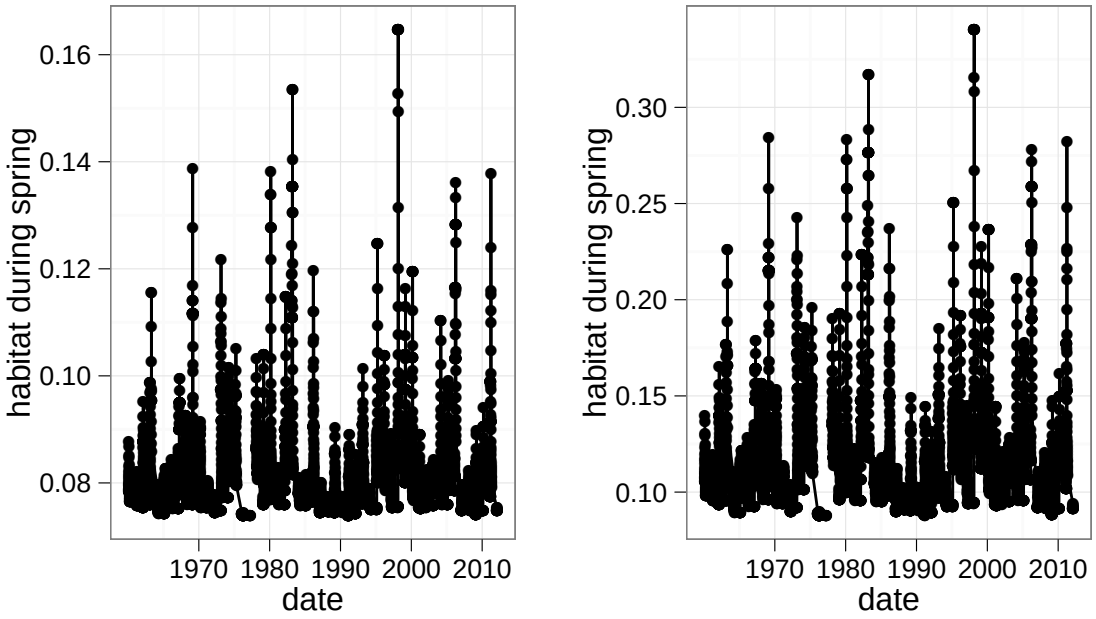
The sensitivity of conclusions to choice among these metrics can and should be examined.

All years

We can apply the flow-area relationships and vital-rate-based metrics to all years in the period of record. First, applying the flow-area relationship yields a hindcast of the spawning habitat available on each day in the season for each year in the record:

```
## full period is d.season.all

p4 <- Plot.flow.area(d.season.all, Convex) + ylab("habitat during spring") +
    theme_bw()
## p4 <- p4 + opts(axis.text.x=theme_text(size=AXISSIZE),
##          axis.text.y=theme_text(size=AXISSIZE),
##          axis.title.x=theme_text(size=LABELSIZE),
## axis.title.y=theme_text(size=LABELSIZE,
#   angle=90))+ylim(c(0,0.4))
```

```
p5 <- Plot.flow.area(d.season.all, Concave) + ylab("habitat during spring") +
    theme_bw()
grid.arrange(p4, p5, ncol = 2)

## Warning message: Removed 14 rows containing missing values (geom_path).
## Warning message: Removed 126 rows containing missing values (geom_point).
## Warning message: Removed 14 rows containing missing values (geom_path).
## Warning message: Removed 126 rows containing missing values (geom_point).
```

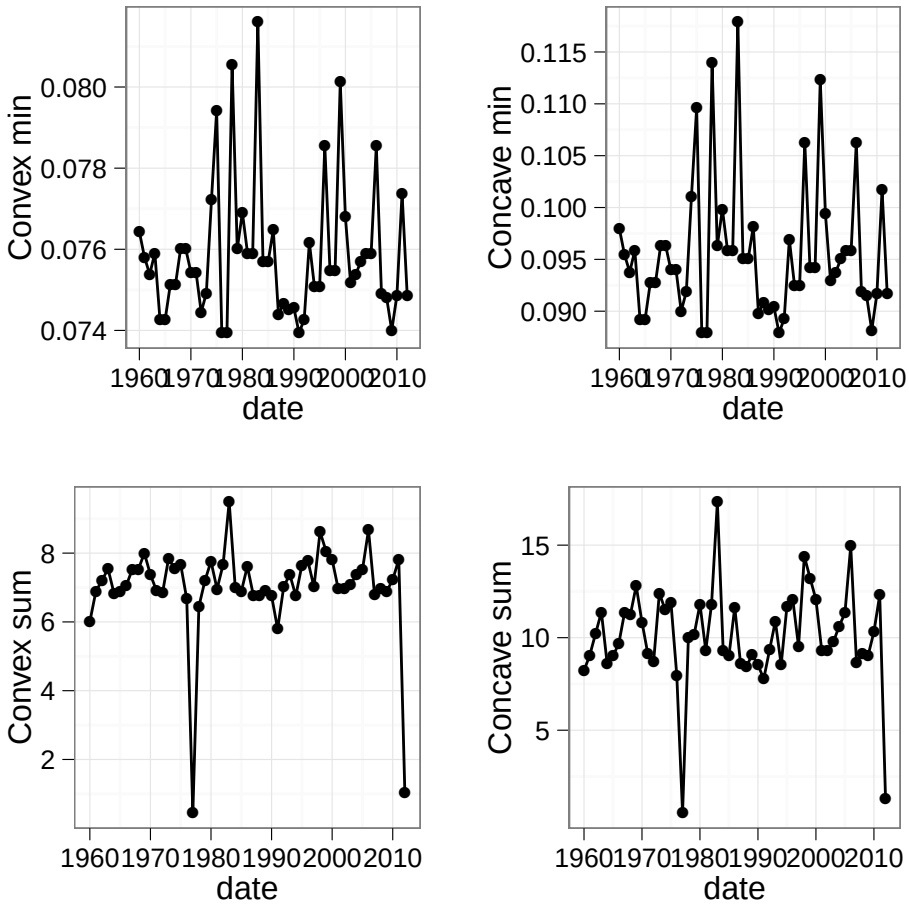


Second, applying vital rate-based metric yields an index of habitat value in each year.

Check out the difference between incised channels (convex flow-area relationship) and reconnected channels (concave flow-area relationship) for different metrics.

Change in hindcast values for the time series/ hydrograph:

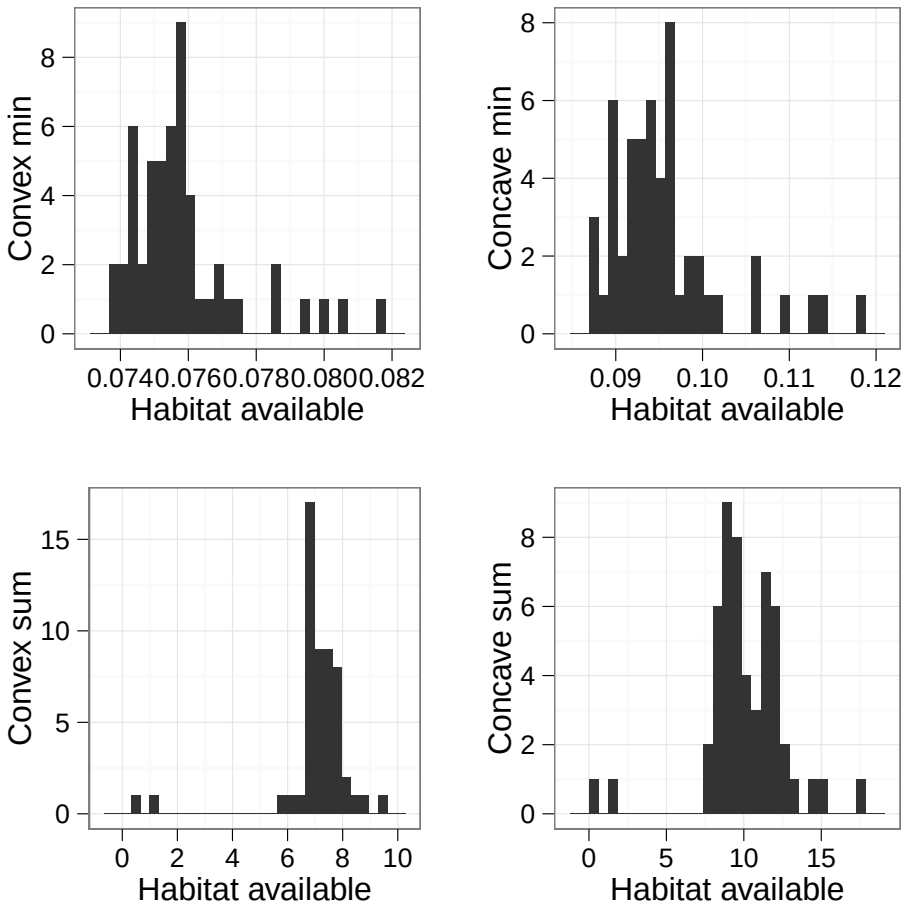
```
flow.area.fn <- c("Convex", "Concave")  ## for the two relationships
vital.metrics <- c("min", "sum")  ## for the two metrics

plots <- c()
for (i in 1:length(vital.metrics)) {
    metric <- vital.metrics[i]
    plot <- lapply(1:length(flow.area.fn), function(j) {
        flow.area <- flow.area.fn[j]
        d.all <- Flow.area.ts(d.season.all, get(flow.area))  ## seasonal flow to area
        yearly.values <- daily2annual(d.all, get(metric))
        p <- gHyd(yearly.values) + geom_line() + geom_point() + theme_bw()
        p <- p + ylab(paste(flow.area, metric))
        return(p)
    })
    plots <- c(plots, plot)
}

do.call(grid.arrange, plots)
```



(need to fix yaxes)
Change in distribution of values for the time series/ hydrograph:

```
plots <- c()
for (i in 1:length(vital.metrics)) {
    metric <- vital.metrics[i]
    plot <- lapply(1:length(flow.area.fn), function(j) {
        flow.area <- flow.area.fn[j]
        d.all <- Flow.area.ts(d.season.all, get(flow.area))  ## seasonal flow to area
        yearly.values <- daily2annual(d.all, get(metric))
        habitat.dist <- data.frame(habitat = coredata(yearly.values))
        h <- qplot(habitat, data = habitat.dist, geom = "histogram",
            xlab = "Habitat available")
        h <- h + theme_bw()
        h <- h + ylab(paste(flow.area, metric))
        return(h)
    })
    plots <- c(plots, plot)
}
do.call(grid.arrange, plots)

## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

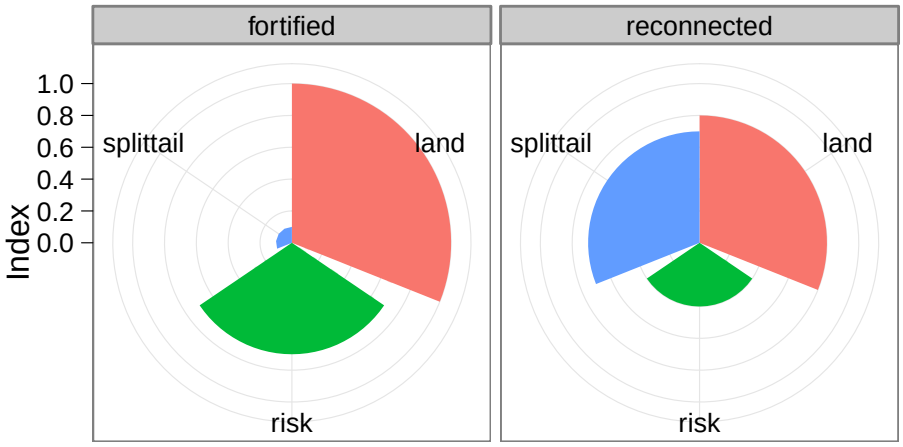
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



(need to fix yaxes)

2.5 Example results

```
rose.ex <- data.frame(alternative = c(rep("fortified", 3),
    rep("reconnected", 3)), index = rep(c("splittail", "risk", "land"),
    2), value = c(0.1, 0.7, 1, 0.7, 0.4, 0.8))

#rose.ex2 <-
#   data.frame(alternative=c(rep('fortified',2),rep('reconnected',2)),
#                       index=rep(c('splittail','land'), 2),
#                       value=c(0.1,  1.0, 0.7,  0.8))

rose <- ggplot(rose.ex)
r1 <- rose + geom_bar(aes(index, value, fill = index)) +
    coord_polar() + theme_bw() + ylab("Index") + xlab("") + opts(legend.position =
"none") +
    facet_wrap(~alternative)
r1
```




```
## rose <- ggplot(subset(rose.ex, alternative=='reconnected'))
## r2 <- rose+geom_bar(aes(index,value,
#
fill=index))+coord_polar()+theme_bw()+ylab('Index')+xlab('')+opts(legend.position='none',
#   title='reconnected')
## r2+opts(axis.text.x=theme_text(size=AXISSIZE),
#   axis.text.y=theme_text(size=AXISSIZE),
#   axis.title.x=theme_text(size=LABELSIZE),
#   axis.title.y=theme_text(size=LABELSIZE, angle=90),
#   plot.title=theme_text(size=28))
```

Colophon

```
require(knitr)  ### the package
knit(paste(getwd(), "hydrologytovital.Rnw", sep = "/"))  ## to run
```

2.6 Appendix

```


## example of window extraction
en <- as.POSIXct("1961-09-29")
st <- as.POSIXct("1959-09-29")
x <- window(d.z, start = st, end = en)
x.spr.ind <- Month2Index(c("Feb", "Mar", "Apr"), x)
x.spr <- x[x.spr.ind]
```

```
x.m <- daily2monthly(x.spr, sum)

## hydroplot working
hydroplot(d.z, var.type = "Flow", pfreq = "dma")  #, from='1959-10-01')

## rolling computation from zoo??
year.max <- rollmax(d.z, 365)
length(year.max)
year.mins


# another subset way, using modular month 1 = october
oct.base <- match(mon, month.abb) + 1
oct.numind <- (seq_along(i.m) + oct.base)%%12 == 0
oct.ind <- i.m[oct.numind]
```