CPUE Training

Dan Ovando

October 11, 2016

Contents

Estandarización de CPUE

CPUE es una de las metricos mas comunes para estimando el estado de poblacíones de pesces. La idea claramente es basico: si estas capturando menos pescado con el mismo esfuerzo, probablemte hay menos peses para capturar.

Pero, es util pensar un poco mas en lo que nos dice CPUE en realidad.

Pensamos en una ecuación basico de captura:

$$captura = biomasa * esfuerzo * capturabilidad$$

Por ejemplo, si hay 100 peses en un lago (*biomasa*), pesco por una hora con pintado (*esfuerzo*), y tengo una 10% probablidad de capturar qualquir peses encuentran my carnada sobre la hora (*capturabilidad*), voy a capturar 10 peses. Si le poblacion de peses baja a 50, entonces voy a capturar 5 peses.

Por esta relacion, podemos decir que

$$\frac{capture}{esfuerzo} = biomass * capturabilidad$$

o mas clara…

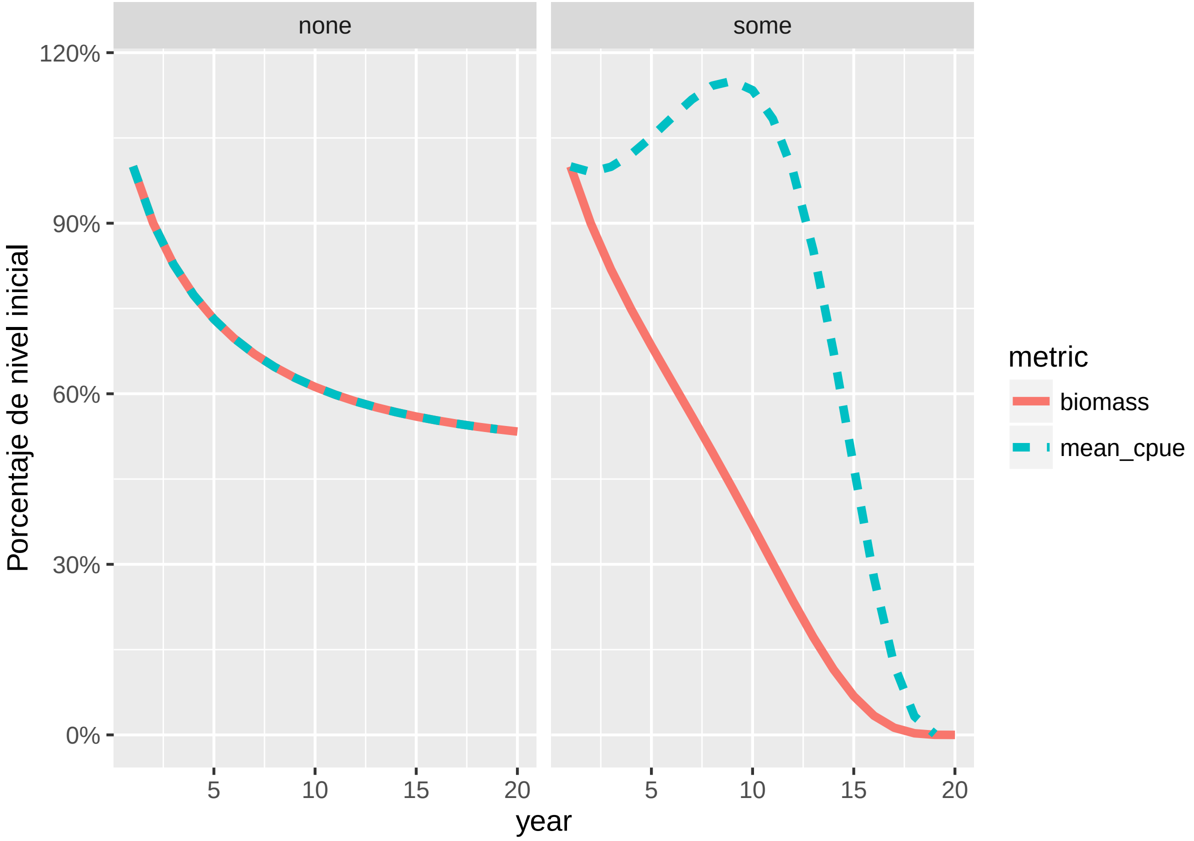
$$CPUE = qB$$

Entonces si B sube o baja, *CPUE* sube o baja. La problem aqui es que si tenemos años de *CPUE* datos, y tratamos de inferar el estado de la poblaciion (B) por la tendencias del *CPUE*.

Se puede ver que si CPUE cambia sobre tiempo, hay dos posibildades que pueden explicar esto: biomasa esta cambiando **o** q esta cambiando.

Ejemplo

```
## Warning: Removed 1 rows containing missing values (geom_path).
```



Suponga que encontramos una pesqueria neovo.

Queremos monitorear el progresso y sostenibilidad de este pesqueria sobre tiempo, y para hacer esto collectamos CPUE sobre tiempo.

Si tecnologia es constante, CPUE es una señal perfecto de biomasa

```
a <-  cpue_example %>%
  filter(creep == 'none') %>%
  select(year,fleet_cpue,fleet,creep) %>%
  ggplot(aes(year,fleet_cpue, color = fleet, linetype = fleet)) +
  geom_line(size = 1.5) +
  facet_wrap(~creep)

a
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```

Como podemos coregir esto?

Usando regression para estandarizacion

```
cpue_example <- cpue_example %>%
  mutate(log_cpue = log(fleet_cpue))

nocreep_reg <- lm(log_cpue ~ factor(year), data = cpue_example %>% filter(creep == 'none'))
```

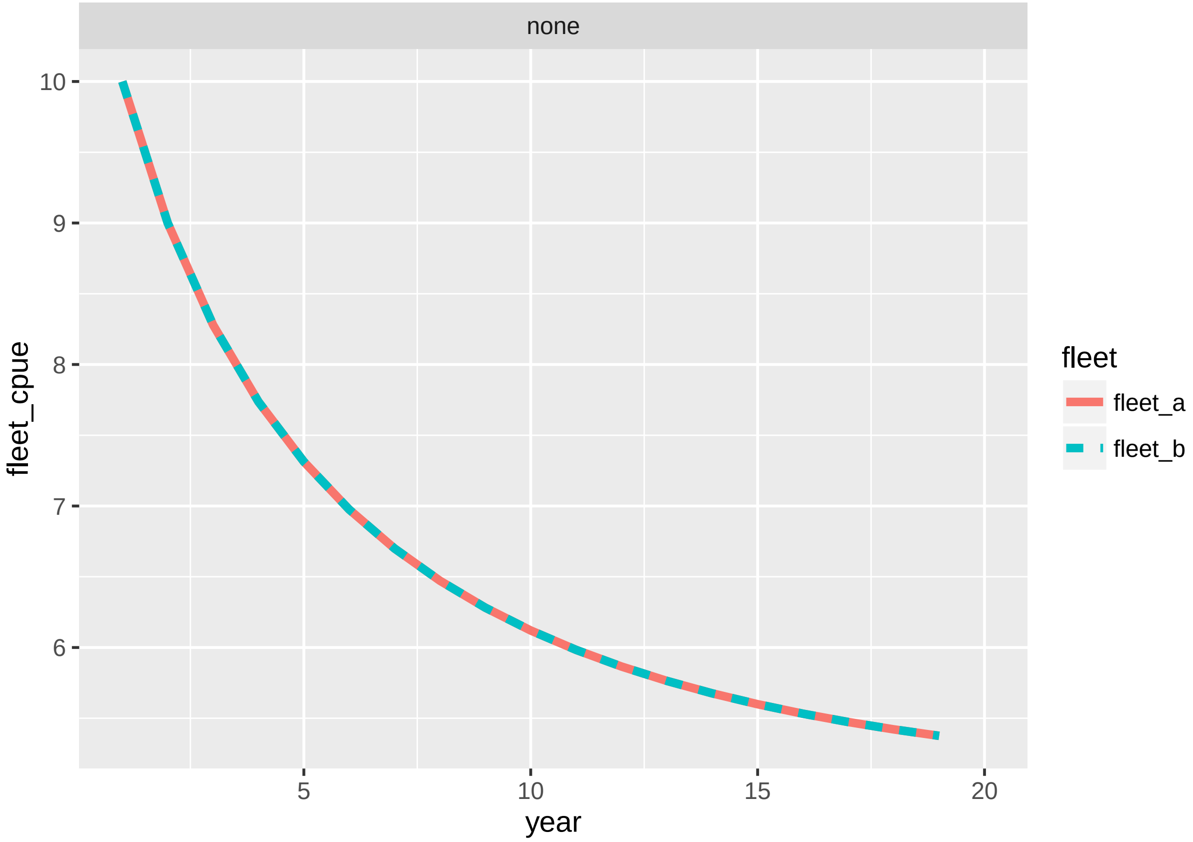


Figure 1: Relacion de CPUE y biomasa con tecnologia constante

```r
creep_reg <- lm(log_cpue ~ factor(year) , data = cpue_example %>% filter(creep == 'some'))

full_creep_reg <- lm(log_cpue ~ factor(year) + fleet, data = cpue_example %>% filter(creep == 'some'))


summary(nocreep_reg)
```

```
## Warning in summary.lm(nocreep_reg): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = log_cpue ~ factor(year), data = cpue_example %>%
##     filter(creep == "none"))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -5.801e-16 -1.021e-16  0.000e+00  1.021e-16  5.801e-16
##
## Coefficients:
##                  Estimate Std. Error    t value Pr(>|t|)
## (Intercept)     2.303e+00  2.007e-16  1.147e+16   <2e-16 ***
## factor(year)2  -1.054e-01  2.839e-16 -3.712e+14   <2e-16 ***
## factor(year)3  -1.887e-01  2.839e-16 -6.649e+14   <2e-16 ***
## factor(year)4  -2.566e-01  2.839e-16 -9.040e+14   <2e-16 ***
## factor(year)5  -3.129e-01  2.839e-16 -1.102e+15   <2e-16 ***
## factor(year)6  -3.603e-01  2.839e-16 -1.269e+15   <2e-16 ***
## factor(year)7  -4.006e-01  2.839e-16 -1.411e+15   <2e-16 ***
## factor(year)8  -4.351e-01  2.839e-16 -1.533e+15   <2e-16 ***
## factor(year)9  -4.650e-01  2.839e-16 -1.638e+15   <2e-16 ***
## factor(year)10 -4.910e-01  2.839e-16 -1.730e+15   <2e-16 ***
## factor(year)11 -5.136e-01  2.839e-16 -1.810e+15   <2e-16 ***
## factor(year)12 -5.335e-01  2.839e-16 -1.879e+15   <2e-16 ***
## factor(year)13 -5.510e-01  2.839e-16 -1.941e+15   <2e-16 ***
## factor(year)14 -5.664e-01  2.839e-16 -1.995e+15   <2e-16 ***
## factor(year)15 -5.800e-01  2.839e-16 -2.043e+15   <2e-16 ***
## factor(year)16 -5.920e-01  2.839e-16 -2.086e+15   <2e-16 ***
## factor(year)17 -6.027e-01  2.839e-16 -2.123e+15   <2e-16 ***
## factor(year)18 -6.122e-01  2.839e-16 -2.157e+15   <2e-16 ***
## factor(year)19 -6.207e-01  2.839e-16 -2.187e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.839e-16 on 19 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 8.302e+29 on 18 and 19 DF,  p-value: < 2.2e-16
```

```r
summary(creep_reg)
```

```
##
## Call:
## lm(formula = log_cpue ~ factor(year), data = cpue_example %>%
##     filter(creep == "some"))
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6409 -0.7977  0.0000  0.7977  1.6409
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.30259    0.96044   2.397  0.02695 *
## factor(year)2   -0.01420    1.35827  -0.010  0.99177
## factor(year)3   -0.01735    1.35827  -0.013  0.98994
## factor(year)4   -0.01589    1.35827  -0.012  0.99079
## factor(year)5   -0.01482    1.35827  -0.011  0.99141
## factor(year)6   -0.01855    1.35827  -0.014  0.98925
## factor(year)7   -0.03150    1.35827  -0.023  0.98174
## factor(year)8   -0.05850    1.35827  -0.043  0.96609
## factor(year)9   -0.10524    1.35827  -0.077  0.93905
## factor(year)10 -0.17872    1.35827  -0.132  0.89670
## factor(year)11 -0.28800    1.35827  -0.212  0.83434
## factor(year)12 -0.44511    1.35827  -0.328  0.74672
## factor(year)13 -0.66663    1.35827  -0.491  0.62919
## factor(year)14 -0.97624    1.35827  -0.719  0.48105
## factor(year)15 -1.40932    1.35827  -1.038  0.31249
## factor(year)16 -2.02257    1.35827  -1.489  0.15288
## factor(year)17 -2.91783    1.35827  -2.148  0.04481 *
## factor(year)18 -4.32696    1.35827  -3.186  0.00487 **
## factor(year)19 -7.45134    1.35827  -5.486 2.72e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.358 on 19 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.7937, Adjusted R-squared:  0.5983
## F-statistic: 4.062 on 18 and 19 DF,  p-value: 0.001956
```

```
summary(full_creep_reg)
```

```
##
## Call:
## lm(formula = log_cpue ~ factor(year) + fleet, data = cpue_example %>%
##     filter(creep == "some"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8205 -0.4330  0.0000  0.4330  0.8205
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.12303    0.52632   5.934 1.29e-05 ***
## factor(year)2   -0.01420    0.72548  -0.020   0.9846
## factor(year)3   -0.01735    0.72548  -0.024   0.9812
## factor(year)4   -0.01589    0.72548  -0.022   0.9828
## factor(year)5   -0.01482    0.72548  -0.020   0.9839
## factor(year)6   -0.01855    0.72548  -0.026   0.9799
## factor(year)7   -0.03150    0.72548  -0.043   0.9658
## factor(year)8   -0.05850    0.72548  -0.081   0.9366
## factor(year)9   -0.10524    0.72548  -0.145   0.8863
## factor(year)10 -0.17872    0.72548  -0.246   0.8082
## factor(year)11 -0.28800    0.72548  -0.397   0.6961
## factor(year)12 -0.44511    0.72548  -0.614   0.5472
## factor(year)13 -0.66663    0.72548  -0.919   0.3703
## factor(year)14 -0.97624    0.72548  -1.346   0.1951
## factor(year)15 -1.40932    0.72548  -1.943   0.0679 .
## factor(year)16 -2.02257    0.72548  -2.788   0.0121 *
## factor(year)17 -2.91783    0.72548  -4.022   0.0008 ***
## factor(year)18 -4.32696    0.72548  -5.964 1.21e-05 ***
## factor(year)19 -7.45134    0.72548 -10.271 5.91e-09 ***
## fleetfleet_b   -1.64089    0.23538  -6.971 1.64e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7255 on 18 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.9443, Adjusted R-squared:  0.8854
## F-statistic: 16.05 on 19 and 18 DF,  p-value: 1.163e-07
```

```
termfun <- function(x) {

  out = as.numeric(paste(x, collapse = ''))

  return(out)
}

  tidy_nocreep <- tidy(nocreep_reg) %>%
  mutate(year_term = str_detect(term,'year')) %>%
  mutate(trans_term = exp(estimate + std.error^2/2)) %>%
  filter(year_term == T) %>%
  mutate(year = str_extract_all(term,'[\\d]', simplify = F),
         year = map_dbl(year,termfun))
```

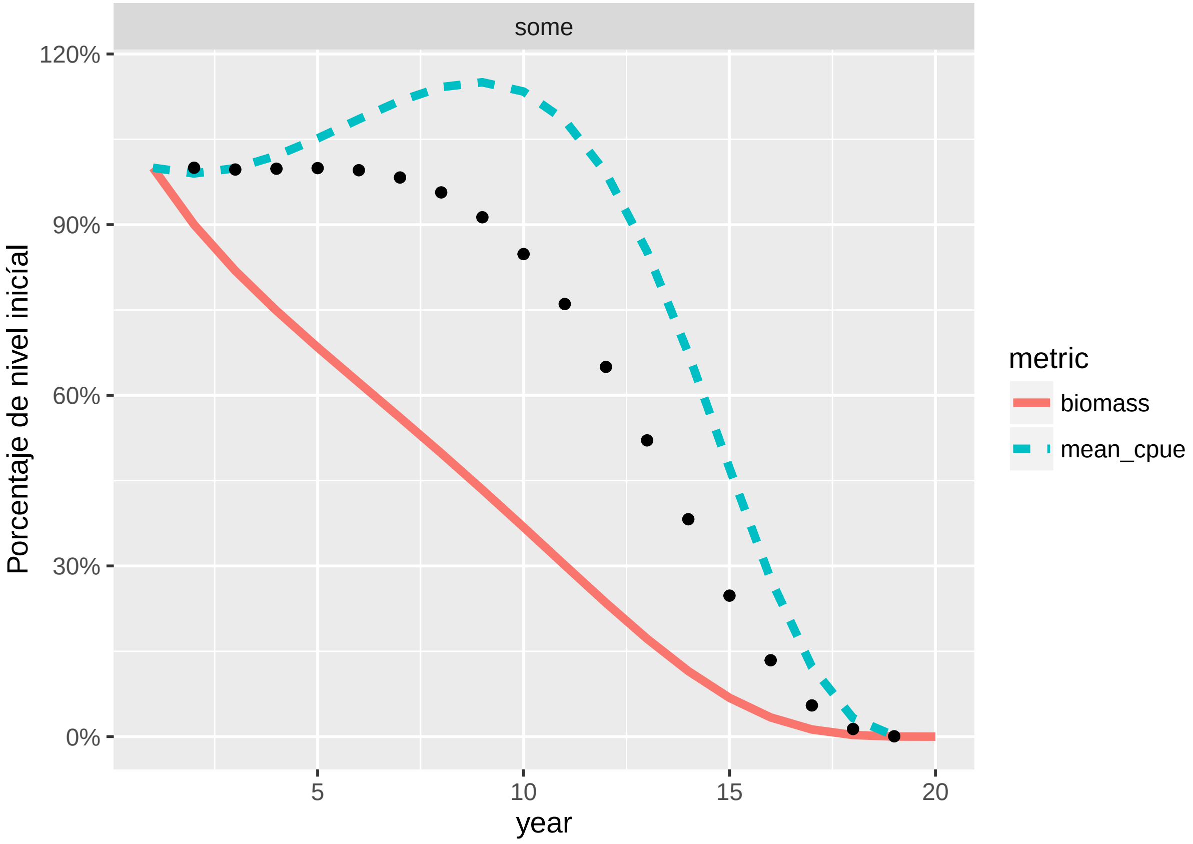
```
## Warning in summary.lm(x): essentially perfect fit: summary may be
## unreliable
```

```
  tidy_creep <- tidy(creep_reg) %>%
  mutate(year_term = str_detect(term,'year')) %>%
  mutate(trans_term = exp(estimate + std.error^2/2)) %>%
  filter(year_term == T) %>%
  mutate(year = str_extract_all(term,'[\\d]', simplify = F),
         year = map_dbl(year,termfun))

   tidy_fullcreep <-  tidy(full_creep_reg) %>%
  mutate(year_term = str_detect(term,'year')) %>%
  mutate(trans_term = exp(estimate + std.error^2/2)) %>%
  filter(year_term == T) %>%
  mutate(year = str_extract_all(term,'[\\d]', simplify = F),
         year = map_dbl(year,termfun),
         rel_trans_term = trans_term/trans_term[year == min(year)])

 cpue_example %>%
   filter(creep == 'some') %>%
  group_by(year, creep) %>%
  summarise(biomass = mean(b, na.rm = T), mean_cpue = mean(fleet_cpue, na.rm = T),
            median_cpue = median(fleet_cpue, na.rm = T)) %>%
  gather(metric,value, biomass,mean_cpue) %>%
  group_by(creep, metric) %>%
  mutate(value = value/value[year == 1]) %>%
  ggplot(aes(year,value)) +
  geom_line(size = 1.5, aes(color = metric, linetype = metric)) +
  facet_wrap(~creep) +
  scale_y_continuous(name = 'Porcentaje de nivel inicíal', labels = percent) +
   geom_point(data = tidy_fullcreep, aes(year,rel_trans_term))
```

```
## Warning: Removed 1 rows containing missing values (geom_path).
```


some
Porcentaje de nivel inicíal
year
metric
biomass
mean_cpue