Maturity

Javier Martinez Arribas, CIBIO-TropiBIO (javimartinezarribas@gmail.com)

16 noviembre 2022

Contents

Age-Length Keys

as idades dos peixes são dados importantes para a compreensão a dinâmica das populações de peixes. Estimando a idade para um grande número de peixes, no entanto, é trabalhoso. Felizmente, geralmente há uma forte relação entre o comprimento e a idade, e medindo o comprimento de um grande número de peixes é relativamente fácil.

Assim, a estrutura etária para uma grande amostra de peixes pode ser estimado de forma confiável resumindo a relação entre idade e comprimento para uma subamostra de peixes e, em seguida, aplicar este resumo a toda a amostra. O resumo da relação entre idade e comprimento é uma chave de comprimento de idade (ALK).

A tabela de probabilidades condicionais, $p_{j|i} = \frac{n_{ij}}{n_i}$ derivadas da amostra envelhecida é o ALK.

Primeiro carregamos as bibliotecas que vamos usar e depois nos conectamos ao banco de dados para carregar os dados relacionados ao peso-tamanho do peixe.

```
load.libraries <- c('DBI','tidyverse','tidymodels','modeltime','timetk','lubridate',
                    'imputeTS','plm','tsibble','fable','FSA','missMethods','nnet',
                    'effects','ranger','vip','tune','randomForest','NeuralNetTools',
                    'mice')
install.lib <- load.libraries[!load.libraries %in% installed.packages()]

for(libs in install.lib) install.packages(libs, dependences = TRUE)
sapply(load.libraries, require, character = TRUE)
```

```
##            DBI      tidyverse     tidymodels      modeltime         timetk
##           TRUE           TRUE           TRUE           TRUE           TRUE
##      lubridate       imputeTS            plm        tsibble          fable
##           TRUE           TRUE           TRUE           TRUE           TRUE
##            FSA    missMethods           nnet        effects         ranger
##           TRUE           TRUE           TRUE           TRUE           TRUE
##            vip           tune   randomForest NeuralNetTools           mice
##           TRUE           TRUE           TRUE           TRUE           TRUE
```

e carregar os dados...

```
dbicon <-  DBI::dbConnect(RPostgres::Postgres(),
                          db = "PESCAz",
                          host = "localhost",
                          port = "5432",
                          user = "postgres",
                          password = "1234")
dbListTables(dbicon)
```

```
## [1] "tblmaturity"
```

```
sql <- 'SELECT * FROM "tblmaturity"'
df_maturity <- dbGetQuery(dbicon, sql)


dbDisconnect(dbicon)
```

Em seguida, selecionamos os preditores que vamos utilizar na análise e calculamos o sexo e a idade na primeira maturação.

```
maturity_tbl <- df_maturity %>%
  select(Data, Ano, Mes, Trimestre, Matura, Sexo, MaturaH1, SexoH1, Complf) %>%
  set_names(c("date", "ano", "mes", "trimestre", "matura", "sexo", "maturaH1",
              "sexoH1", "comprimento"))

# Criando coluna de sexo reunindo o sexo da primeira matura para M,F ou hermafroditas
maturity_tbl <- maturity_tbl %>% mutate(sexo = ifelse(sexoH1 %in% "", sexo, sexoH1))
# Criando coluna de matura reunindo a matura da primeira matura para M,F ou hermafroditas
maturity_tbl <- maturity_tbl %>% mutate(matura = ifelse(maturaH1 %in% "",
                                                        substr(matura,start=1,stop=1),
                                                        maturaH1))
maturity_tbl$day <- gsub("(.*)[-]", "", maturity_tbl$date)

maturity_tbl <- maturity_tbl %>%
  select(date, ano, mes, trimestre, day, comprimento, sexo, matura)

# Alterar alguns formatos
maturity_tbl <- maturity_tbl %>%
                    mutate(ano = as.numeric(ano),
                           day = as.numeric(day),
                           mes = as.numeric(mes),
                           trimestre = as.numeric(trimestre),
                           matura = as.factor(matura))
```

Filtramos as observações com valores da variável sexo ('IND', 'H' e ' ') e observações com compressão=0 e matura vazia, uma vez que serão mais difíceis de imputar.

```
with(maturity_tbl[maturity_tbl$sexo %in% c('','IND','H'),], table(matura))
```

```
## matura
##       0   1   2   3   4   5   I
## 321 286   0   0   0   1   3   3
```

```
maturity_tbl <- maturity_tbl %>%
                  filter(!sexo %in% c('IND','H',''))

with(maturity_tbl[maturity_tbl$comprimento==0,], table(matura))
```

```
## matura
##         0    1    2    3    4    5    I
##   13  723 3516 1585  105   32  332    0
```

```
maturity_tbl <- maturity_tbl %>%
                  filter(!(comprimento==0 & matura==''))


maturity_tbl$matura[maturity_tbl$matura %in% c('I','')] <- NA

# Descartamos os níveis que não são adequados no conjunto de dados:
maturity_tbl$matura <- droplevels(maturity_tbl$matura)

with(maturity_tbl[maturity_tbl$comprimento==0,], table(matura))
```

```
## matura
##    0    1    2    3    4    5
##  723 3516 1585  105   32  332
```

Criamos o preditor de intervalo.

```
maturity_tbl <- maturity_tbl %>% mutate(lcat10=lencat(comprimento,w=10))
```

Separamos as observações com maduro=NA:

```
maturity_pred <- maturity_tbl %>%
                  rowid_to_column() %>%
                  filter(is.na(matura))
maturity_train <- maturity_tbl[-as.numeric(maturity_pred$rowid),]
```

Dividimos o conjunto de dados sem NAs atribuídos à variável madura em duas subamostras. Um será usado para ajuste e o outro para obter uma avaliação mais realista do ajuste. O método utilizado será a amostragem estratificada pela variável madura pois parece que há muita diferença entre o número de registros em cada tipo de matura.

75% train / 25% test

```
data_split <- initial_split(maturity_train,
                            prop = 3/4,
                            strata = matura)
train <- training(data_split)
test <- testing(data_split)

train%>% select(matura) %>% table()
```

```
## .
##    0    1    2    3    4    5
## 1181 5462 3456  295  380  897
```

```
test%>% select(matura) %>% table()
```

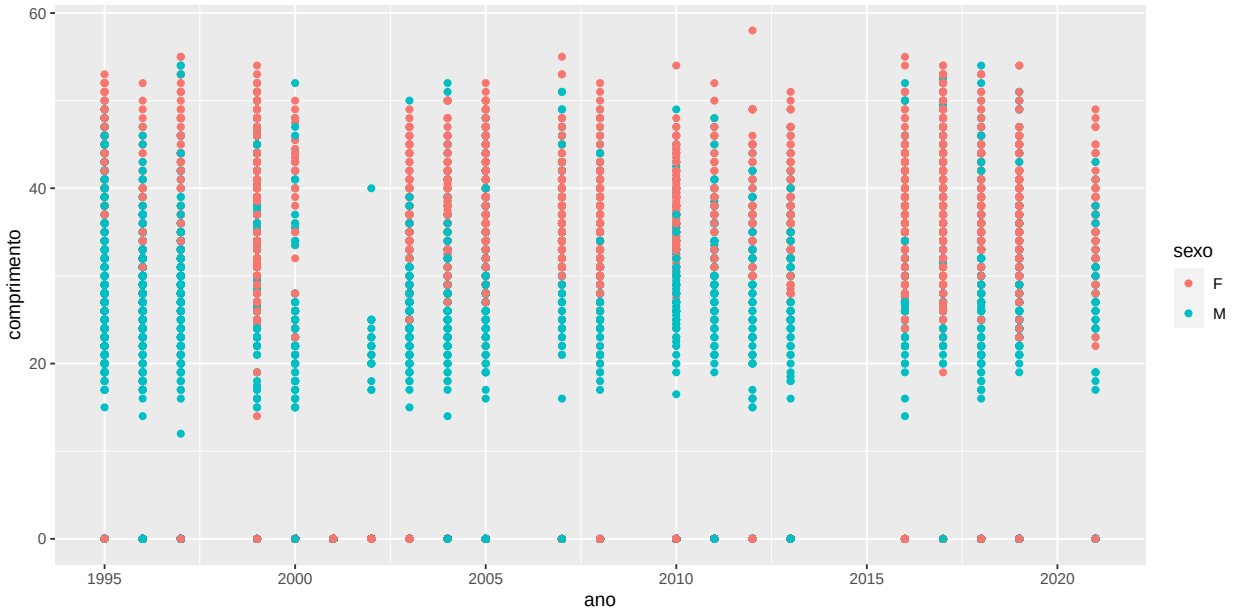
```
## .
##    0    1    2    3    4    5
##  388 1819 1177   94  108  305
```

Análise exploratória de dados

Primeiramente vamos tentar conhecer um pouco melhor as distribuições de nossas variáveis através dos gráficos a seguir.

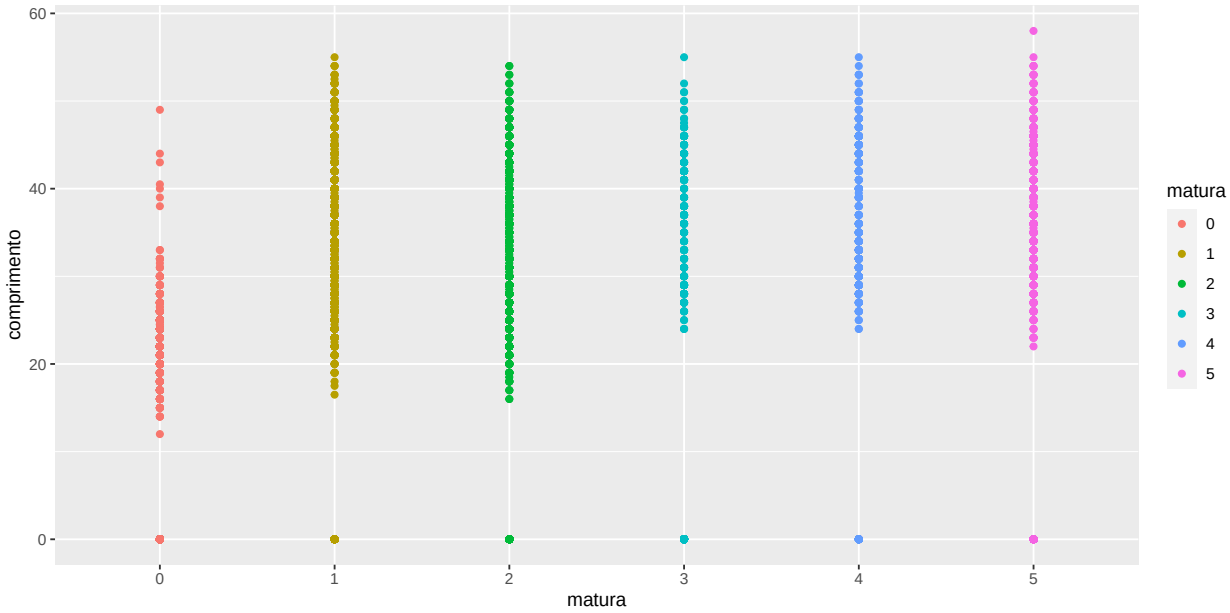
Comprimento ao longo dos anos por sexo:

```
ggplot(train, aes(x=ano, y=comprimento, col=sexo)) +
  geom_point()
```



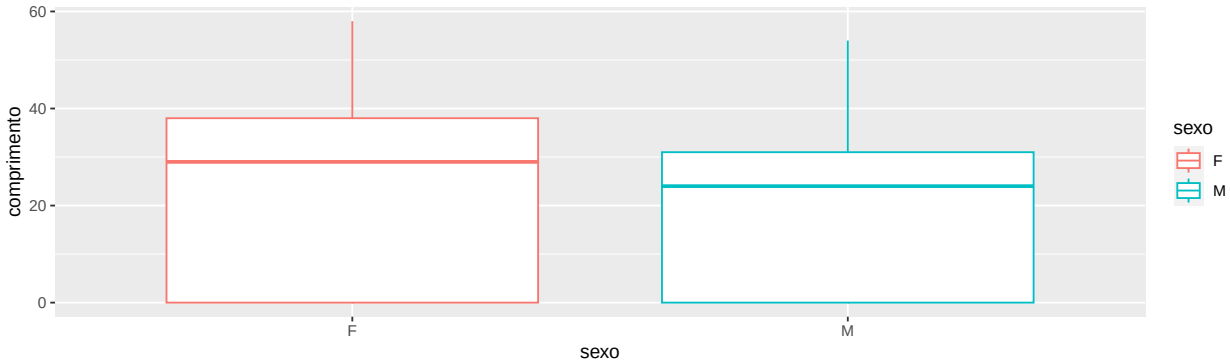
Comprimento por matura:

```
ggplot(train, aes(x=matura, y=comprimento, col=matura, group=matura)) +
  geom_point()
```



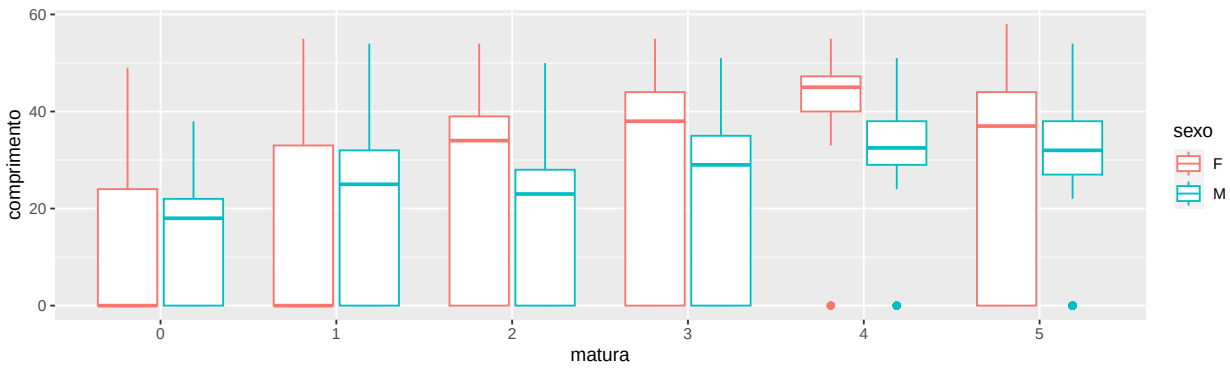
Comprimento por sexo:

```
ggplot(train, aes(x=sexo, y=comprimento, color=sexo)) +
  geom_boxplot()
```



Comprimento por sexo e matura

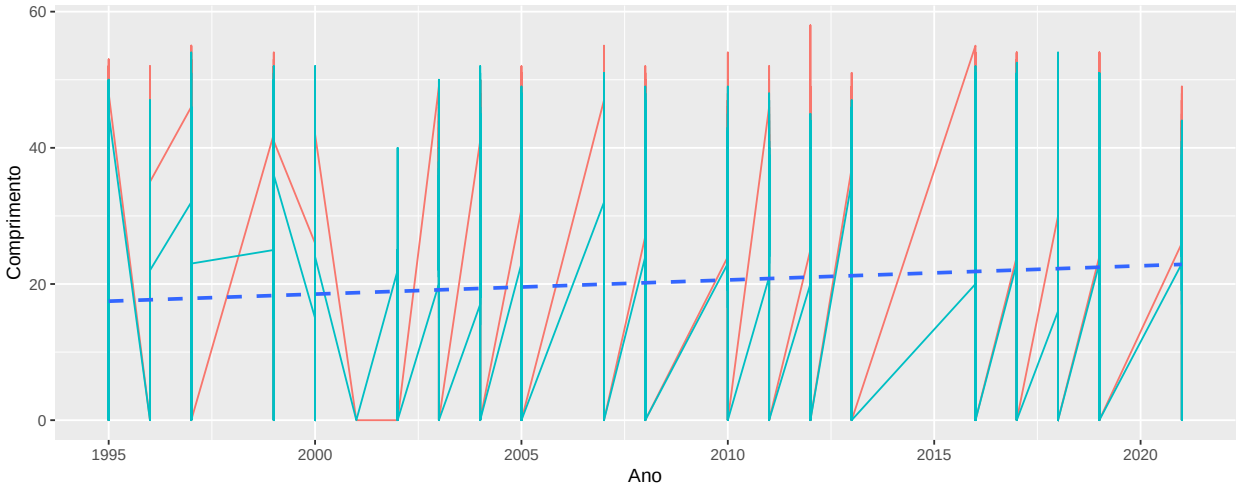
```
ggplot(train, aes(x=matura, y=comprimento, color=sexo)) +
  geom_boxplot()
```



Comprimento médio por ano e sexo

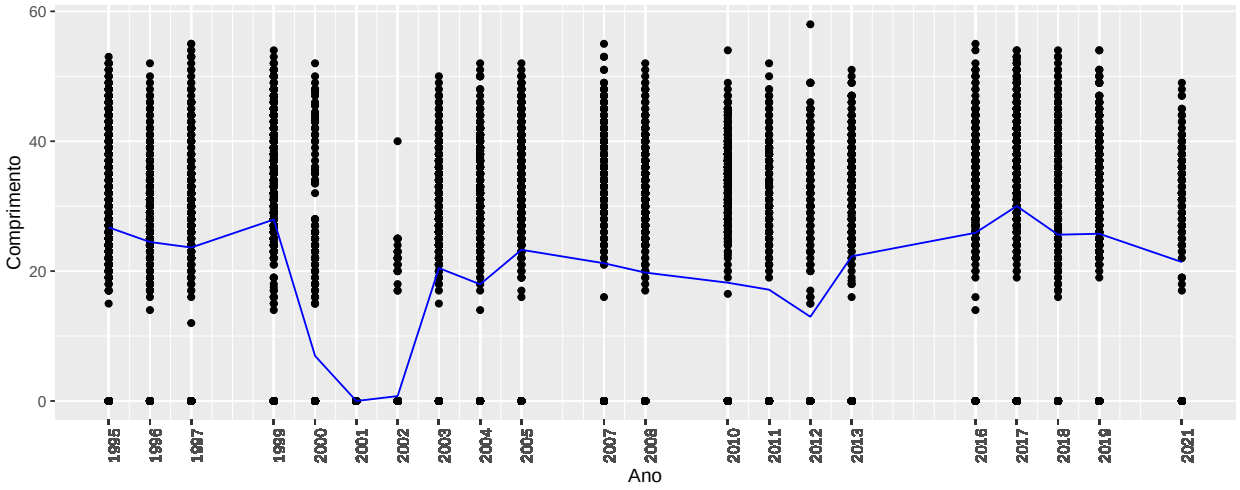
```
ggplot(data = train, aes(x = ano, y = comprimento)) +
  geom_line(aes(colour = as.factor(sexo))) +
  geom_smooth(method = "lm", se = F, lty = "dashed") +
  labs(x = "Ano",  y = "Comprimento") +
  theme(legend.position = "none")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



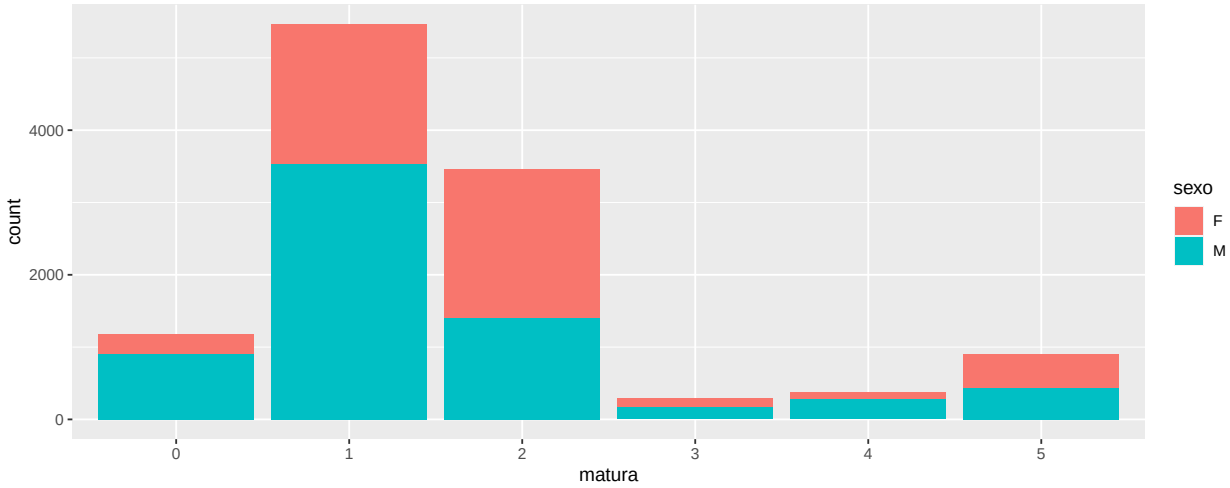
Heterogeneidade ao longo do tempo:

```
#Heterogeneity along the time
train %>%
  group_by(ano) %>%
  summarise(comp_mean = mean(comprimento)) %>%
  left_join(train,by="ano") %>%
  ggplot(data = .,
         aes(x = ano, y = comprimento)) +
  geom_point() +
  geom_line(aes(x = ano, y = comp_mean), col = "blue") +
  scale_x_continuous(labels = as.character(maturity_tbl$ano),
                     breaks = maturity_tbl$ano) +
  labs(x = "Ano", y = "Comprimento") +
  theme(axis.text.x = element_text(angle = 90))
```



Distribuição da matura entre os sexos:

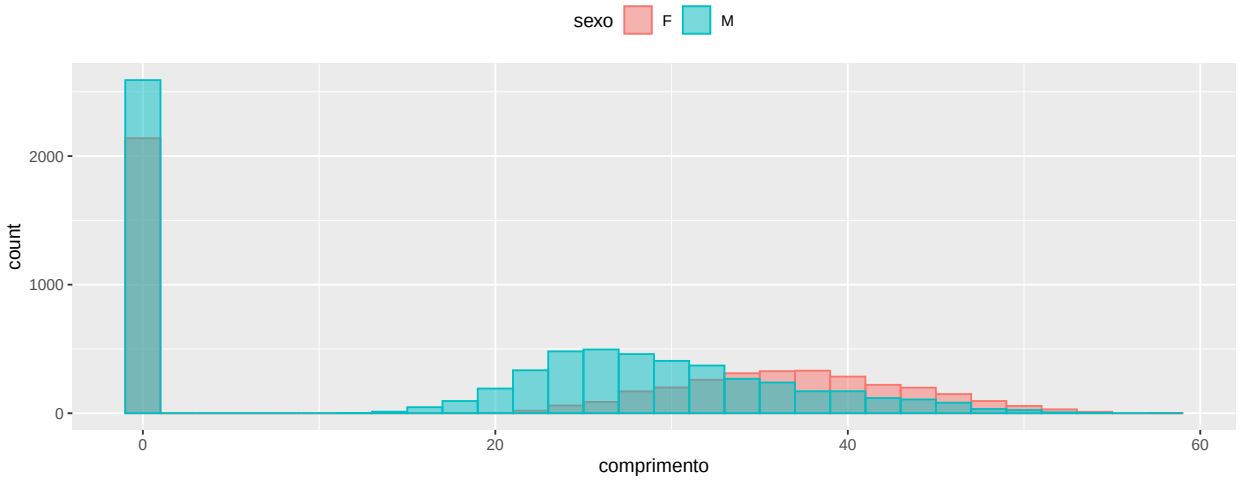
```
ggplot(data=train, aes(x=matura,  fill=sexo)) +
  geom_bar(stat="count")
```



Distribuição do comprimento entre os sexos:

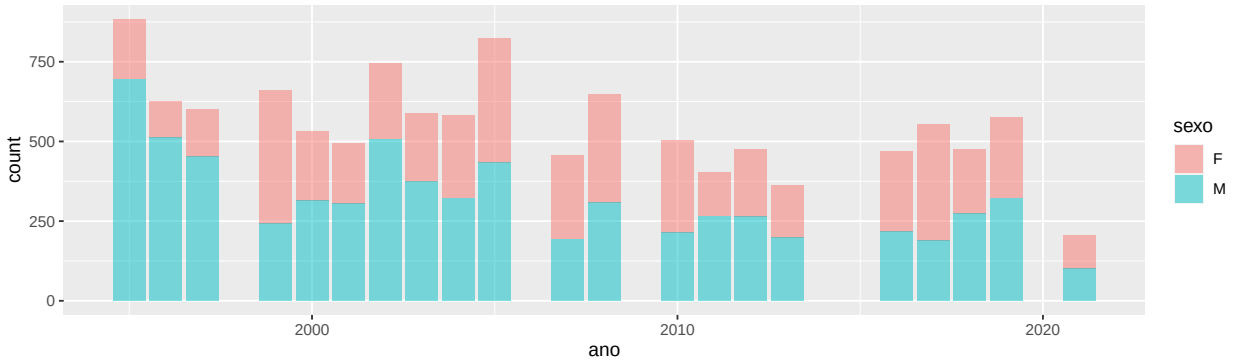
```
# comprimento by sexo distribution
ggplot(data=train, aes(x=comprimento,  fill= sexo, color=sexo)) +
  geom_histogram(position="identity",alpha=0.5)+
  theme(legend.position="top")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



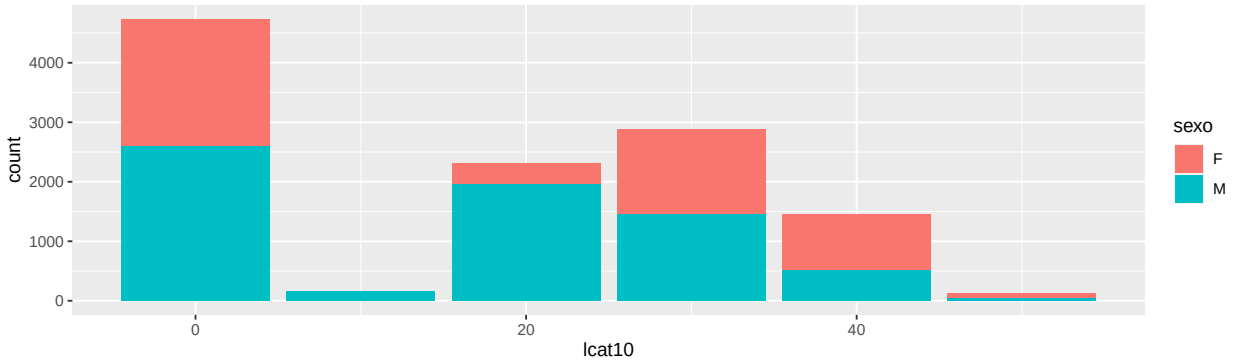
Distribuição do número de observações ao longo do tempo e entre os dois sexos:

```
# Number obs along the time by sexo
ggplot(data = train, aes(x = ano, fill = sexo)) +
  geom_bar(stat="count",alpha=0.5)
```



Distribuição dos intervalos entre os sexos:

```
# lcat10 distribution
ggplot(data=train, aes(x=lcat10,  fill=sexo)) +
  geom_bar(stat="count")
```



Em seguida, vamos estudar 3 modelos diferentes de regressão multinomial:

matura ~ lcat10

matura ~ lcat10 + sexo + lcat10*sexo

matura ~ lcat10 + sexo + ano + lcat10:sexo + lcat10:ano + sexo:ano

E veja se eles são significativamente diferentes um do outro:

Imputamos los valores de comprimento=0 utilizando K-Nearest Neighbourhood = 3

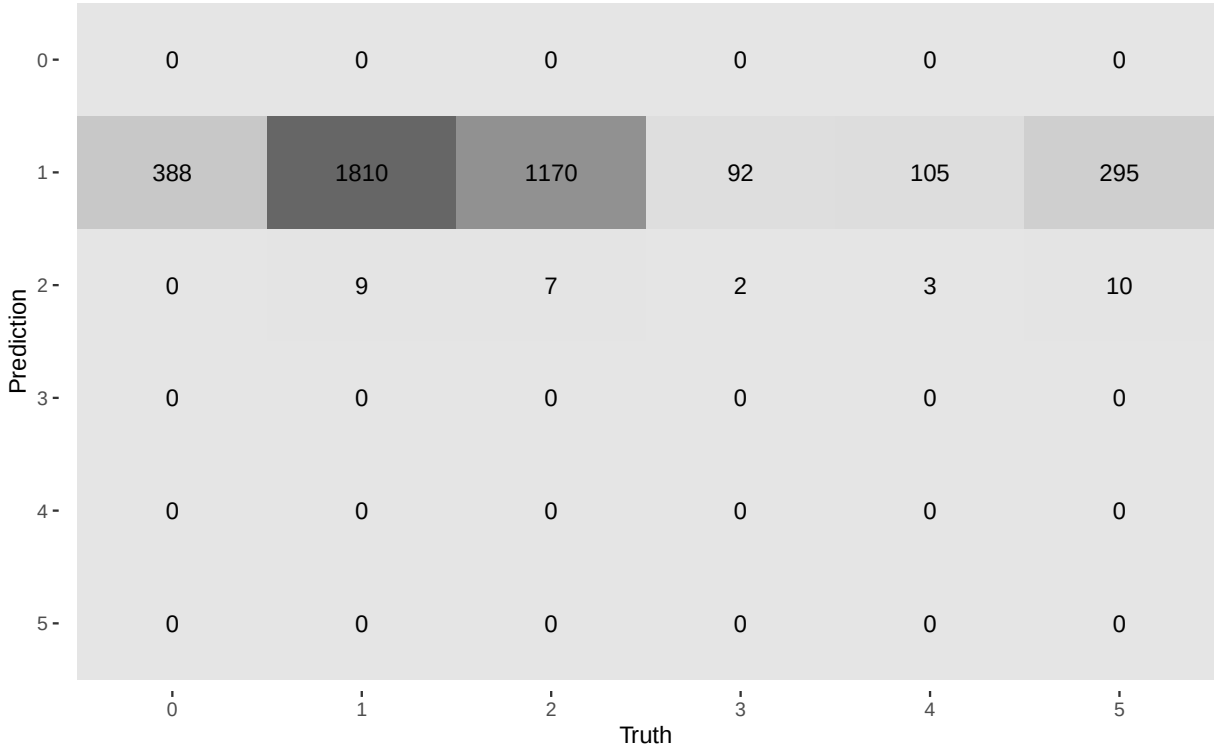
Fazemos validação cruzada para encontrar o melhor valor do hiperparâmetro de penalty:

```
ml1_rec <-
  recipe(matura ~  lcat10,
         data = train) %>%
  step_naomit(everything(), skip = TRUE)

# K-fold cross-validation
set.seed(1234)
cv_folds <- vfold_cv(train, v = 10)

ml_spec <- multinom_reg(
    mode = "classification",
    engine = "nnet",
    penalty = tune(),
    mixture = NULL
)

ml1_wflow <-
  workflow() %>%
  add_recipe(ml1_rec) %>%
  add_model(ml_spec)

random_tune1 <-
  ml1_wflow %>%
  tune_grid(
    resamples = cv_folds, grid = 5
  )


random_final1 <-
  finalize_workflow(ml1_wflow, select_best(random_tune1)) %>%
  fit(train)

last_fit_ml1 <- last_fit(random_final1,
                         split = data_split,
                         metrics = metric_set(
                         recall, precision,
                         roc_auc, sens, spec)
)


last_fit_ml1 %>%
  collect_metrics()
```

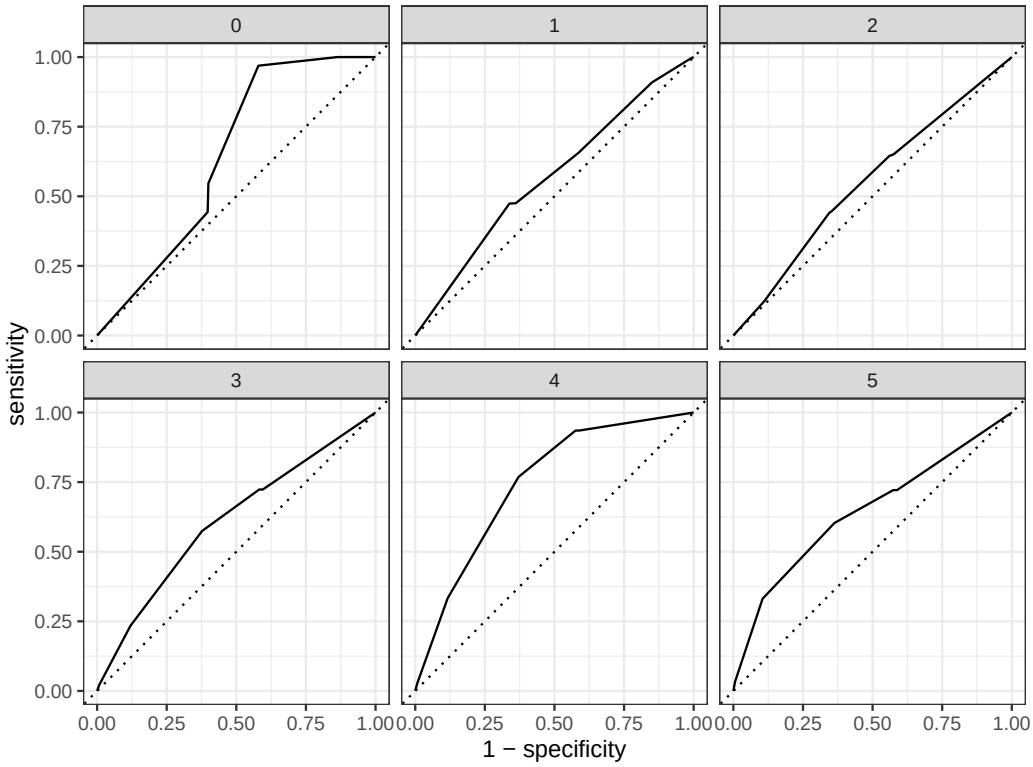
```
## # A tibble: 5 x 4
```

```
##   .metric   .estimator .estimate .config
##   <chr>     <chr>          <dbl> <chr>
## 1 recall    macro          0.167 Preprocessor1_Model1
## 2 precision macro          0.347 Preprocessor1_Model1
## 3 sens      macro          0.167 Preprocessor1_Model1
## 4 spec      macro          0.834 Preprocessor1_Model1
## 5 roc_auc   hand_till      0.615 Preprocessor1_Model1
```

```
last_fit_ml1 %>%
  collect_predictions() %>%
  conf_mat(matura, .pred_class) %>%
  autoplot(type = "heatmap")
```




```
last_fit_ml1 %>%
  collect_predictions() %>%
  roc_curve(matura,'.pred_0','.pred_1','.pred_2','.pred_3','.pred_4','.pred_5') %>%
  autoplot()
```




```
# Ajustamos o modelo final em todo o conjunto de dados.
final_model1 <-
  random_final1 %>%
  fit(data = maturity_train)
```

O ajuste não é muito bom como podemos ver nas curvas ROC.

Se calcularmos o expoente dos coeficientes, podemos ver como eles influenciam os odds de matura:

```
exp(coefficients(final_model1$fit$fit$fit))
```

```
##   (Intercept)   lcat10
## 1  3.47893459 1.023599
## 2  1.62754802 1.042359
## 3  0.09014426 1.062961
## 4  0.03758012 1.106845
## 5  0.25204603 1.067419
```

Conclusões:

Aumentar o valor do intervalo de comprimento um nível aumenta em 10% a probabilidade de que a idade do peixe seja 4 em vez de 0

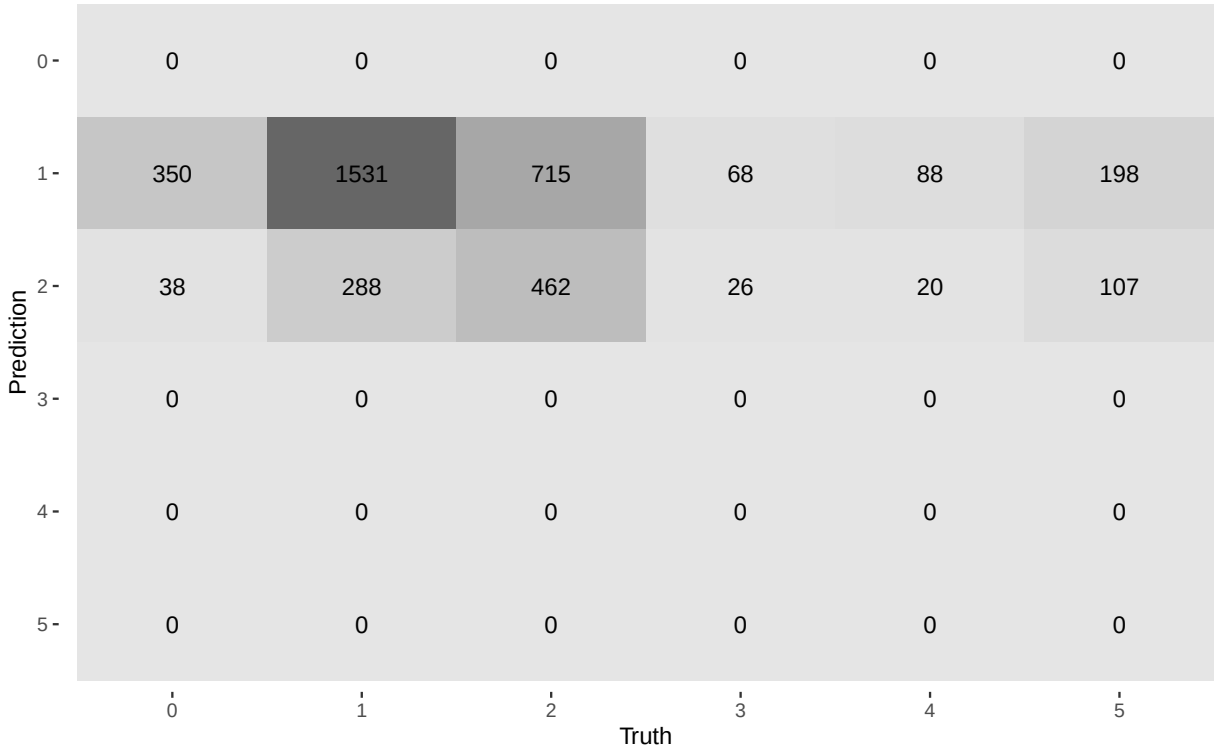
Vejamos o que acontece quando adicionamos a variável sex e suas iterações com a variável interval lcat10:

```
ml2_rec <-
  recipe(matura ~  lcat10 + sexo,
         data = train) %>%
```

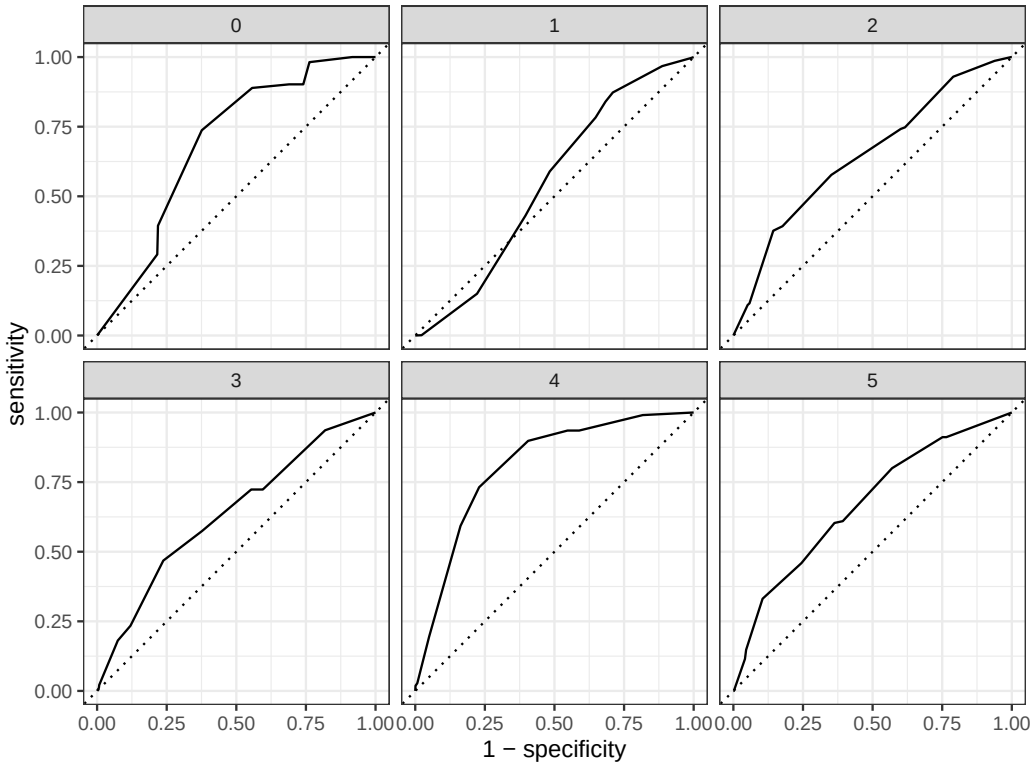
```
  step_naomit(everything(), skip = TRUE) %>%
  step_novel(sexo) %>%
  step_interact(terms = ~ lcat10:sexo)

# K-fold cross-validation
set.seed(1234)
cv_folds <- vfold_cv(train, v = 10)

ml2_wflow <-
  workflow() %>%
  add_recipe(ml2_rec) %>%
  add_model(ml_spec)


set.seed(100)
random_tune2 <-
  ml2_wflow %>%
  tune_grid(
    resamples = cv_folds, grid = 5
  )

random_final2 <-
  finalize_workflow(ml2_wflow, select_best(random_tune2)) %>%
  fit(train)

last_fit_ml2 <- last_fit(random_final2,
                         split = data_split,
                         metrics = metric_set(
                         recall, precision,
                         roc_auc, sens, spec)
)


last_fit_ml2 %>%
  collect_metrics()
```

```
## # A tibble: 5 x 4
##   .metric   .estimator .estimate .config
##   <chr>     <chr>          <dbl> <chr>
## 1 recall    macro          0.206 Preprocessor1_Model1
## 2 precision macro          0.505 Preprocessor1_Model1
## 3 sens      macro          0.206 Preprocessor1_Model1
## 4 spec      macro          0.856 Preprocessor1_Model1
## 5 roc_auc   hand_till      0.656 Preprocessor1_Model1
```

```
last_fit_ml2 %>%
  collect_predictions() %>%
  conf_mat(matura, .pred_class) %>%
  autoplot(type = "heatmap")
```




```
last_fit_ml2 %>%
  collect_predictions() %>%
  roc_curve(matura,'.pred_0','.pred_1','.pred_2','.pred_3','.pred_4','.pred_5') %>%
  autoplot()
```




```
# Ajustamos o modelo final em todo o conjunto de dados.
final_model2 <-
  random_final2 %>%
  fit(data = maturity_train)
```

O ajuste também não parece muito exato.

Se calcularmos o expoente dos coeficientes, podemos ver como eles influenciam os odds de matura:

```
exp(coefficients(final_model2$fit$fit$fit))
```

```
##   (Intercept)   lcat10     sexoM sexonew lcat10_x_sexoM lcat10_x_sexonew
## 1 5.234109969 1.026148 0.5324531       1      1.0007912                1
## 2 3.240930677 1.057359 0.3725817       1      0.9667784                1
## 3 0.117582027 1.078610 0.6817404       1      0.9796628                1
## 4 0.008086712 1.160613 4.6607760       1      0.9565715                1
## 5 0.665124097 1.063029 0.1668999       1      1.0200415                1
```

Conclusões:

Ao aumentar lcat10 um nível aumentará a probabilidade de ter um matura em 16% 4 em vez de um 0 maduro.

Parece que um matura 4 é 4 vezes mais provável de ocorrer em peixes do sexo M do que em peixes do sexo F.

O aumento de lcat10 em um nível parece afetar a maturidade da mesma forma em ambos os sexos M como em F.

Um teste de razão de verossimilhança para identificar uma diferença em ALKs entre dois grupos requer o ajuste de dois modelos multinomiais. O primeiro modelo mais simples tem intervalo de comprimento como a única variável explicativa. O segundo modelo mais complexo tem intervalo de comprimento, a variável fator que identifica os grupos, e a interação entre essas duas variáveis como Variáveis explicativas.

```
anova(final_model1$fit$fit$fit,final_model2$fit$fit$fit)
```

```
## Likelihood ratio tests of Multinomial Models
##
## Response: ..y
##                                                 Model Resid. df Resid. Dev   Test
## 1                                               lcat10     77800   40573.67
## 2 lcat10 + sexo + lcat10_x_sexoM + lcat10_x_sexonew     77790   39325.37 1 vs 2
##      Df LR stat. Pr(Chi)
## 1
## 2    10   1248.3       0
```

Los dos modelos son estadisticamente diferentes, luego la distribución de los tamaños dentro de cada edad es diferente para cada sexo.

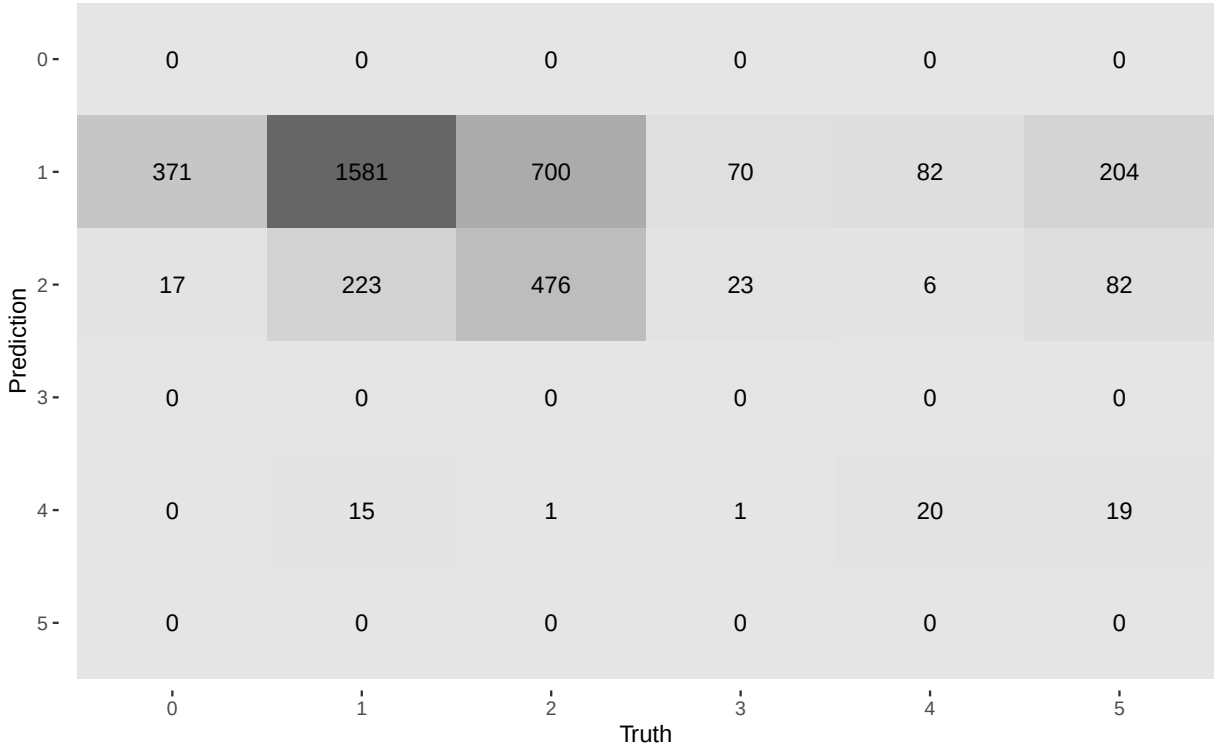
Se agora adicionarmos o ano: Comprimento de primeira maturação por sexo e por ano bem como todas as iterações de variáveis de dois por dois que podem ser obtidas:

```
ml3_rec <-
  recipe(matura ~  lcat10 + sexo + ano,
         data = train) %>%
  step_naomit(everything(), skip = TRUE) %>%
  step_novel(sexo) %>%
  step_interact(~all_predictors() * all_predictors())


# ml_spec sigue siendo el mismo. Modelo regresión multinomial

ml3_wflow <-
  workflow() %>%
  add_recipe(ml3_rec) %>%
  add_model(ml_spec)

random_tune3 <-
  ml3_wflow %>%
  tune_grid(
    resamples = cv_folds, grid = 5
  )

random_final3 <-
  finalize_workflow(ml3_wflow, select_best(random_tune3)) %>%
  fit(train)

last_fit_ml3 <- last_fit(random_final3,
                         split = data_split,
                         metrics = metric_set(
                         recall, precision,
                         roc_auc, sens, spec)
)
```

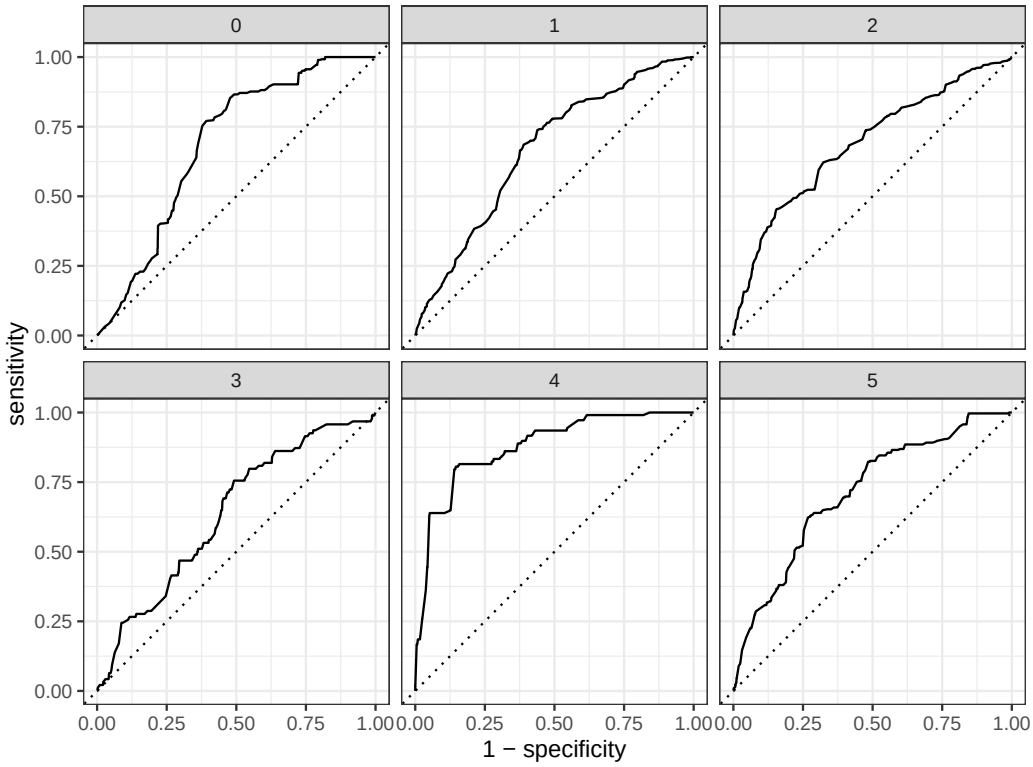
```
last_fit_ml3 %>%
  collect_metrics()
```

```
## # A tibble: 5 x 4
##   .metric   .estimator .estimate .config
##   <chr>     <chr>          <dbl> <chr>
## 1 recall    macro          0.243 Preprocessor1_Model1
## 2 precision macro          0.486 Preprocessor1_Model1
## 3 sens      macro          0.243 Preprocessor1_Model1
## 4 spec      macro          0.862 Preprocessor1_Model1
## 5 roc_auc   hand_till      0.688 Preprocessor1_Model1
```

```
last_fit_ml3 %>%
  collect_predictions() %>%
  conf_mat(matura, .pred_class) %>%
  autoplot(type = "heatmap")
```




```
last_fit_ml3 %>%
  collect_predictions() %>%
  roc_curve(matura,'.pred_0','.pred_1','.pred_2','.pred_3','.pred_4','.pred_5') %>%
  autoplot()
```




```
# Ajustamos o modelo final em todo o conjunto de dados ee estudamos os coeficientes:
final_model3 <-
  random_final3 %>%
  fit(data = maturity_train)
```

```
exp(coefficients(final_model3$fit$fit$fit))
```

```
##    (Intercept)       lcat10        sexoM sexonew       ano lcat10_x_sexoM
## 1 1.184001e+00 6.984567e-01 6.297307e-25       1 1.0007420      0.9985308
## 2 1.570489e-63 1.013560e+00 3.402493e+86       1 1.0753450      0.9703583
## 3 3.423172e-79 9.300283e+02 1.179558e+29       1 1.0930634      0.9784271
## 4 2.429496e+02 1.234203e+02 8.062050e+98       1 0.9950938      0.9706961
## 5 3.094536e+69 2.785309e-03 1.360551e+84       1 0.9230775      1.0363072
##   lcat10_x_sexonew lcat10_x_ano sexoM_x_ano sexonew_x_ano
## 1                1    1.0001918   1.0278639             1
## 2                1    1.0000197   0.9050057             1
## 3                1    0.9966355   0.9670165             1
## 4                1    0.9976667   0.8928280             1
## 5                1    1.0029696   0.9068102             1
```

Conclusões:

O ajuste melhora um pouco, talvez um pouco mais para peixes de maturidade 4.

Parece que aumentando a variável ano em uma unidade, é 11% mais provável que matura seja igual a 3 do que a 0.

Um teste de razão de verossimilhança:

```
anova(final_model2$fit$fit$fit,final_model3$fit$fit$fit)
```

```
## Likelihood ratio tests of Multinomial Models
##
## Response: ..y
##                                                                                                                    Mode
## 1                                                    lcat10 + sexo + lcat10_x_sexoM + lcat10_x_sexone
## 2 lcat10 + sexo + ano + lcat10_x_sexoM + lcat10_x_sexonew + lcat10_x_ano + sexoM_x_ano + sexonew_x_an
##   Resid. df Resid. Dev   Test    Df LR stat. Pr(Chi)
## 1     77790   39325.37
## 2     77775   37733.36 1 vs 2    15 1592.008       0
```

O terceiro modelo é estatististicamente diferente do segundo, então podemos dizer que a distribuição de tamanhos dentro de cada idade e sexo é diferente ao longo os anos.

Se finalmente compararmos os 3 modelos para ver qual deles tem um valor menor para o critério de Akaike, veremos que:

```
AIC(final_model1$fit$fit$fit,final_model2$fit$fit$fit,final_model3$fit$fit$fit)
```

```
##                          df      AIC
## final_model1$fit$fit$fit 10 40593.67
## final_model2$fit$fit$fit 20 39365.37
## final_model3$fit$fit$fit 35 37803.36
```

Podemos afirmar que conseguimos um melhor ajuste com o terceiro modelo, inclua a variável ano e as iterações duas a duas entre todas as variáveis.

Se calculássemos a maturação da subamostra com valores de NAs por meio da matriz ALK, obteríamos as seguintes previsões:

```
mALK <- round(prop.table(table(test$lcat10,test$matura),1),2)
df_ALK <- data.frame(apply(mALK,1,function(row) colnames(mALK)[which.max(row)]))
colnames(df_ALK) <- c('matura')
df_ALK$matura <- as.factor(df_ALK$matura)
df_ALK <- rownames_to_column(df_ALK, "lcat10")
df_ALK$lcat10 <- as.numeric(df_ALK$lcat10)



df <- as.data.frame(cbind(predict(random_final1, test),predict(random_final2, test),
                          predict(random_final3, test),
                          left_join(test,df_ALK, by="lcat10")$matura.y,test$matura))
colnames(df) <- c("model_1","model_2","model_3","model_ALK","observed")

table(df$model_1)
```

```
##
##    0    1    2    3    4    5
##    0 3860   31    0    0    0
```

```
table(df$model_2)
```

```
##
##    0    1    2    3    4    5
##    0 2950  941    0    0    0
```

```
table(df$model_3)
```

```
##
##    0    1    2    3    4    5
##    0 3008  827    0   56    0
```

```
table(df$model_ALK)
```

```
##
##    0    1    5
##   49 3811   31
```

```
table(df$observed)
```

```
##
##    0    1    2    3    4    5
##  388 1819 1177   94  108  305
```

Agora se calcularmos as previsões podemos ver quais são os valores que mais prevê no conjunto de dados com NAs:

Vamos ahora a probar otro modelo para predecir la submuestra con valores NAs en la variable matura.

```
maturity_pred <- maturity_tbl %>%
                  rowid_to_column() %>%
                  filter(is.na(matura))

maturity_train <- maturity_tbl[-as.numeric(maturity_pred$rowid),]


data_split <- initial_split(maturity_train,
                            prop = 3/4,
                            strata = matura)

# Create dataframes for the two sets:
train <- training(data_split)
test <- testing(data_split)
```

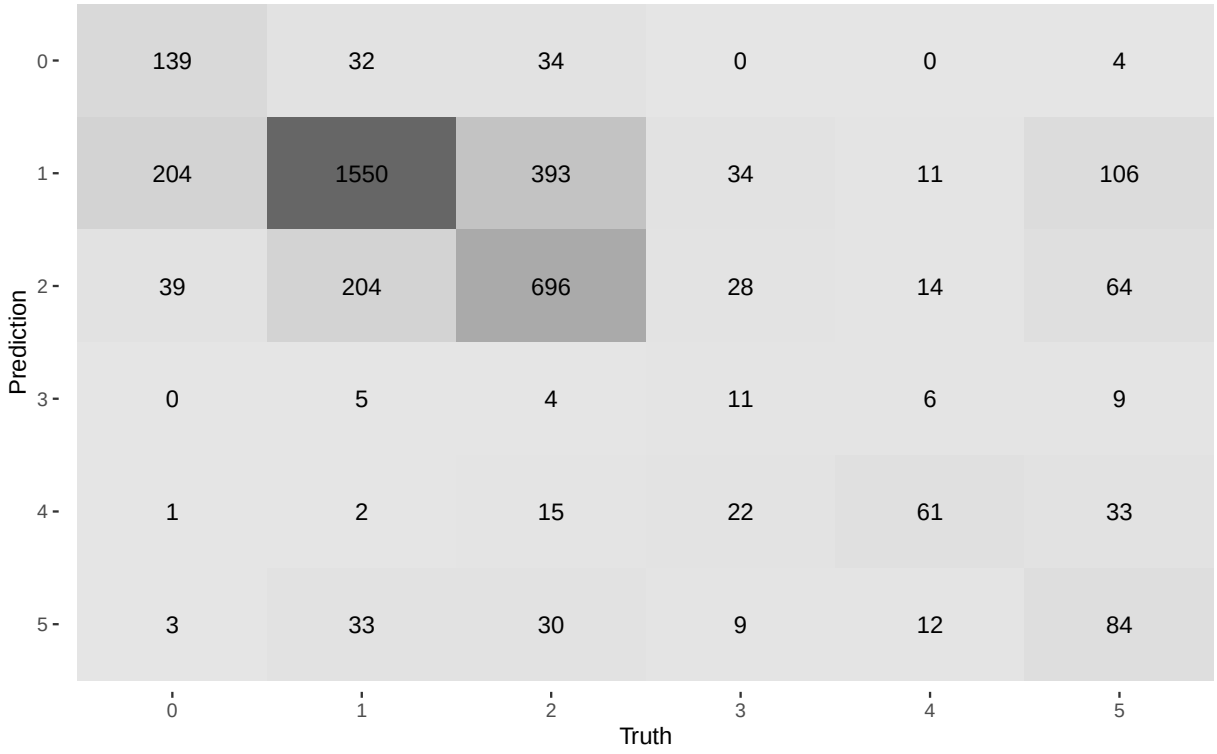
Y construimos el modelo siguiendo los mismos pasos que en el caso de la regresión multinomial:

```
rf_rec <-
  recipe(matura ~  comprimento + sexo + ano + mes + trimestre + day,
         data = train) %>%
         step_naomit(everything(), skip = TRUE) %>%
```

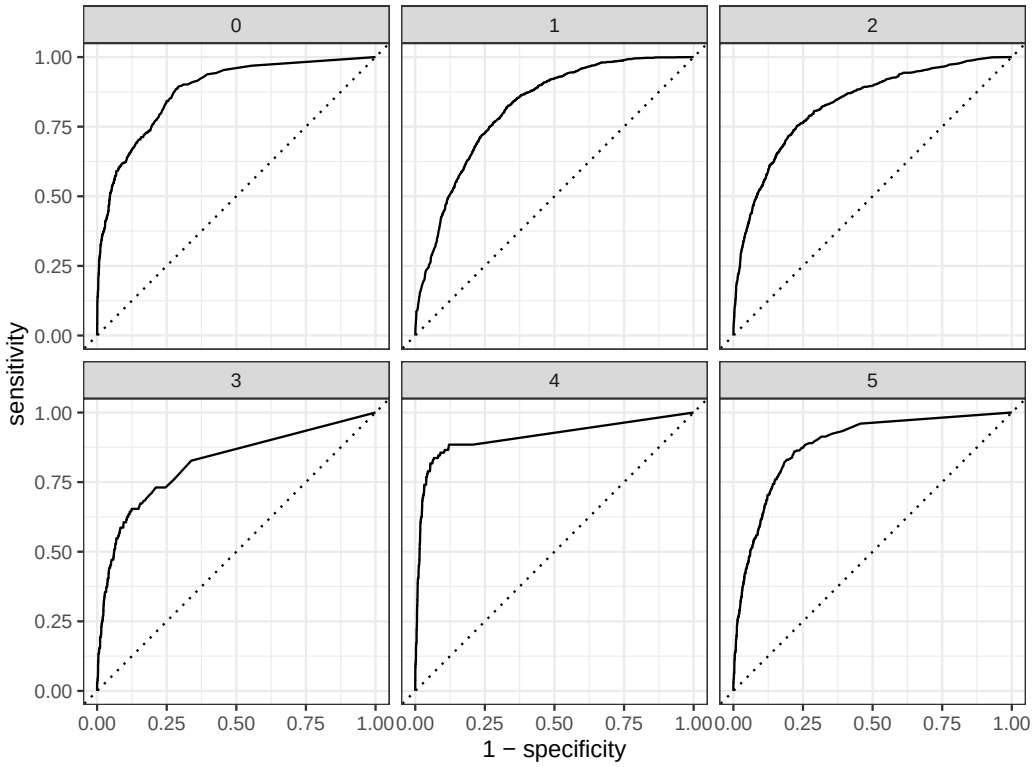
```
        step_novel(sexo) %>%
        step_impute_knn(comprimento, neighbors = 3)

# K-fold cross-validation
set.seed(1234)
cv_folds <- vfold_cv(train, v = 10)

rf_spec <-
  rand_forest(
    mode = "classification",
    mtry = tune(),
    trees = tune()
  ) %>%
  set_engine("randomForest")

rf_wflow <-
  workflow() %>%
  add_recipe(rf_rec) %>%
  add_model(rf_spec)

random_tune_rf <-
  rf_wflow %>%
  tune_grid(
    resamples = cv_folds, grid = 5
  )

random_final_rf <-
  finalize_workflow(rf_wflow, select_best(random_tune_rf)) %>%
  fit(train)

last_fit_rf <- last_fit(random_final_rf,
                        split = data_split,
                        metrics = metric_set(
                        recall, precision,
                        roc_auc, sens, spec)
)

last_fit_rf %>%
  collect_metrics()
```

```
## # A tibble: 5 x 4
##   .metric   .estimator .estimate .config
##   <chr>     <chr>          <dbl> <chr>
## 1 recall    macro          0.463 Preprocessor1_Model1
## 2 precision macro          0.544 Preprocessor1_Model1
## 3 sens      macro          0.463 Preprocessor1_Model1
## 4 spec      macro          0.907 Preprocessor1_Model1
## 5 roc_auc   hand_till      0.837 Preprocessor1_Model1
```

```
last_fit_rf %>%
  collect_predictions() %>%
  conf_mat(matura, .pred_class) %>%
  autoplot(type = "heatmap")
```

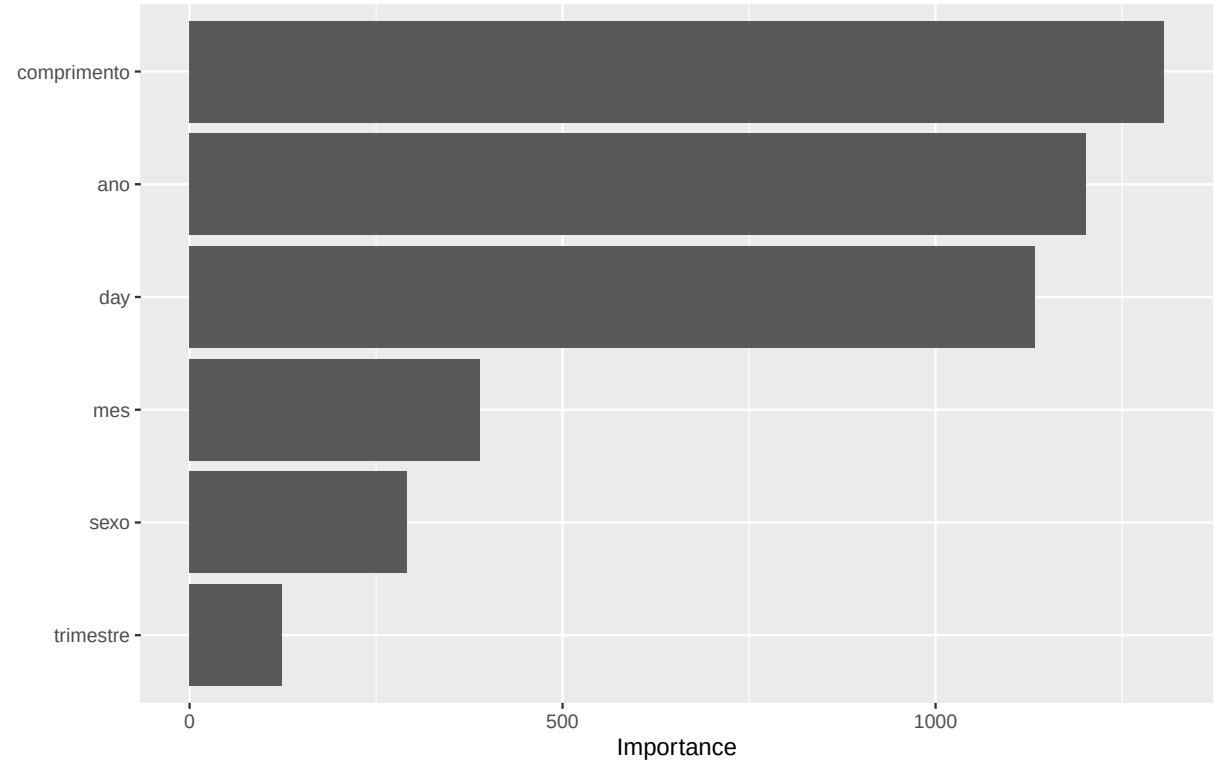



```
last_fit_rf %>%
  collect_predictions() %>%
  roc_curve(matura,'.pred_0','.pred_1','.pred_2','.pred_3','.pred_4','.pred_5') %>%
  autoplot()
```



Podemos também estudar a importância que o modelo dá a cada variável no ajuste:

```
last_fit_rf %>%
  pluck(".workflow", 1) %>%
  extract_fit_parsnip() %>%
  vip(num_features = 10)
```



O comprimento, o dia e o ano de pesca parecem ser as variáveis que melhor discriminam as maturas.

Finalmente, podemos fazer as previsões das amostras com NAs atribuídas na maturidade para poder reconstruir a série original

```
pred <- random_final_rf %>%
          predict(maturity_pred) %>%
          bind_cols(maturity_pred)

maturity_pred <- maturity_pred %>%
                  mutate(matura=pred$.pred_class)


maturity_pred <- maturity_pred %>% select(-rowid)
cc.fnl <- rbind(maturity_train,maturity_pred)

cc.sumlen <- cc.fnl %>% group_by(matura) %>%
  summarize(n_obs=validn(comprimento),mean_comprimento=mean(comprimento,na.rm=TRUE),
            standard_deviation=sd(comprimento,na.rm=TRUE),standard_error=se(comprimento,na.rm=TRUE)) %>%
  as.data.frame()
cc.sumlen
```

```
##   matura n_obs mean_comprimento standard_deviation standard_error
## 1      0  1569         12.21415           11.68091      0.2948934
## 2      1  7310         17.29896           17.36242      0.2030727
## 3      2  4634         22.03025           16.96045      0.2491491
## 4      3   389         26.88689           17.43490      0.8839843
## 5      4   488         34.68648           11.59370      0.5248220
## 6      5  1202         27.46339           18.05262      0.5207005
```

```
plot(comprimento~matura,data=cc.fnl,pch=19,col=rgb(0,0,0,1/10),
     xlab="Age",ylab="Total Length (mm)",ylim=c(0,205))
lines(mean_comprimento~matura,data=cc.sumlen,lwd=2,lty=2)
```