Seasonal timing in fruit flies: figures, surival analysis, SVM

Andrew D. Nguyen

2023-08-25

Contents

I'll be demo'ing the tidyverse, surival analysis, and SVM on a project I worked on in my postdoc at the Unversity of Florida. The project details are publicly available through this github repository.

Load libraries

```
library(tidyverse) # for ggplot2, data visualization and data filtering with dplyr
library(ggbeeswarm) # for quasirandom plotting
library(caret)# for ML analysis
library(survival) # for survival analysis

#ggplot2 settings I like:
T<-theme_bw()+theme(,text=element_text(size=18),
                    axis.text=element_text(size=18),
                    panel.grid.major=element_blank(),
                    panel.grid.minor.x = element_blank(),
                    panel.grid = element_blank(),
                    legend.key = element_blank(),
                    axis.title.y=element_text(margin=margin(t=0,r=15,b=0,l=0)),
                    axis.title.x=element_text(margin=margin(t=15,r=,b=0,l=0)))#+ theme(legend.position=
```

The dataset! Let's explore different plots

There's a nice dataset I generated from my postdoctoral research in Dr. Dan Hahn's lab. I collected eclosion times (when an insect transitions from pupae to adult) for fruit flies (Rhagoletis) from either the apple or hawthorne fruit. The project and details are found here and has metadata.

Apologies, but latex cuts off the dataset URL, so **here** it is.

```
fruit.fly<-read.csv("https://raw.githubusercontent.com/adnguyen/Circadian_rhythm_runs_seasonal_timing/ma
#glimpse(fruit.fly)
#take out the data of interest
fruit.fly1<-fruit.fly%>%
  select(Site_name,mass_day10,Host,cohort_day,resp_day15,new.eclosions,organism,treatment)%>%
  dplyr::filter(organism=="fly" &treatment !="GC" & treatment!="")
# GC = genetic controls, those were saved for genomic analyses
#and we're just focusing on eclosion of flies (not the parasites)
glimpse(fruit.fly1)
```

```
## Rows: 564
## Columns: 8
## $ Site_name     <chr> "OG", "Ferris", "OG", "OG", "OG", "Ferris", "Ferris", "O~
## $ mass_day10    <dbl> 6.938, 6.719, 3.848, 6.413, 9.365, 7.978, 9.778, 6.499, ~
## $ Host          <chr> "Apple", "Apple", "Apple", "Apple", "Apple", "Apple", "A~
## $ cohort_day    <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,~
## $ resp_day15    <dbl> 0.1432514, 0.1076286, 0.1182286, 0.1601623, 1.4328830, 0~
## $ new.eclosions <int> 74, 65, 56, NA, 32, 63, 82, 39, 68, 32, 56, 56, 63, 46, ~
## $ organism      <chr> "fly", "fly", "fly", "fly", "fly", "fly", "fly", "fly", ~
## $ treatment     <chr> "SO", "RT", "RT", "SO", "RT", "SO", "SO", "RT", "SO", "R~
```

I need to find out proportions of eclosion and need to transform eclosion data such that NA = 0 and a number in days = 1. And I want to display the proportions by treatment (RT = Room Temperature vs SO = Simulated Overwintering) and host fruit (apple vs haw).

```
fruit.fly1$eclfac<-as.numeric(ifelse(fruit.fly1$new.eclosions>1,"1","0"))
fruit.fly1$eclfac[is.na(fruit.fly1$eclfac)]<-0

ecl.num<-fruit.fly1%>%
  group_by(treatment,Host,eclfac)%>%
  count()
knitr::kable(ecl.num)
```

treatment	Host	eclfac	n
RT	Apple	0	2
RT	Apple	1	198
RT	Haw	0	2
RT	Haw	1	34
SO	Apple	0	3
SO	Apple	1	242
SO	Haw	0	2
SO	Haw	1	81

You can see most flies have eclosed.

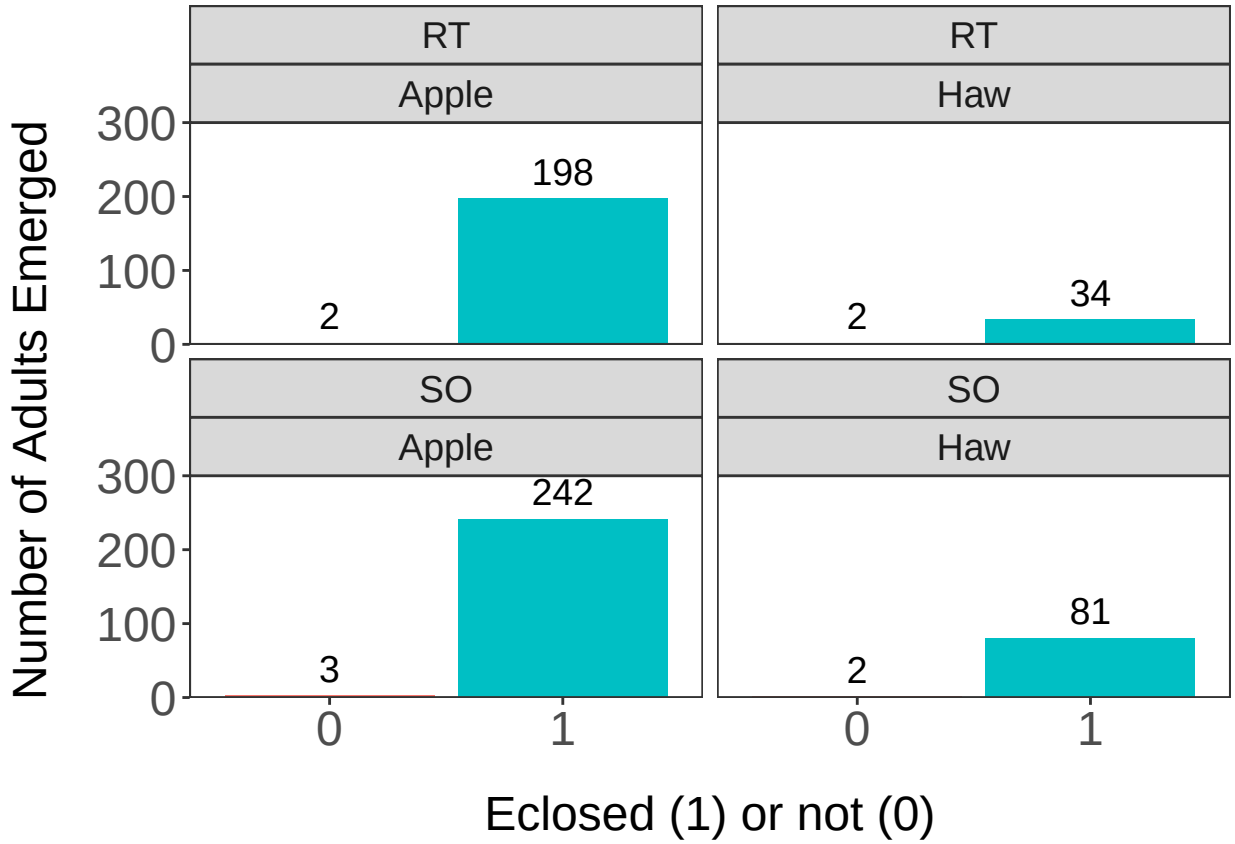
We can show the data in a number of different ways:

- barplots
- cumulative distribution plots

Barplots

Let's first start off with a barplot:

```
#let's create a barplot
# treatments, RT = room temperature, SO = simulated overwintering
ggplot(ecl.num,aes(y=n,x=factor(eclfac),fill=factor(eclfac)))+
  geom_bar(stat="identity")+
  facet_wrap(treatment~Host)+ #this function creates multi-panels
#and is super handy at conveying complex datasets with multiple treatments /categorical variables
  ylab("Number of Adults Emerged")+xlab("Eclosed (1) or not (0)")+
  T+ # my own plotting settings, see library
  theme(legend.position="none")+ # removing legend
  scale_y_continuous(expand=c(0,0),limits=c(0,300),breaks=seq(0,300,100),labels=seq(0,300,100))+
# makes sure the y-axis starts at 0, this is something
#I always use when making barplots with ggplot2
  geom_text(aes(label=n),size=5,vjust=-.5) # it is nice to annotate
```




```
#the bars to help the eyes interpret them
```

Empirical Cumulative Distribution Function plots

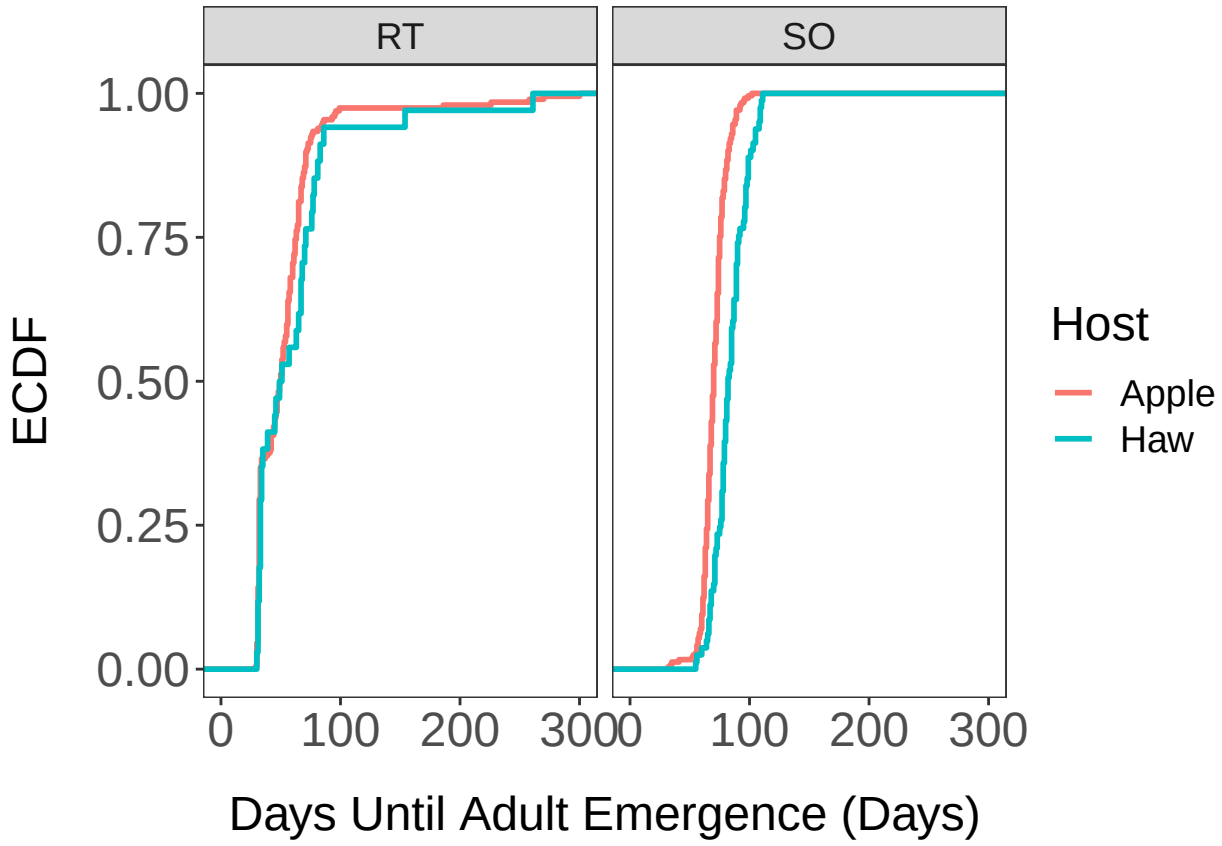
Let's remove the ones that have not eclosed and observe their differences in timing and we can visualize the data as a cumulative distribution plot. ==Remember to save the figure with ggsave() and you can specify the dpi and dimensions of the plot.==

I think this is a nice plot to show for these data. You can see that fruit flies on apple host eclose (adult emergence from pupae) sooner than fruit flies on hawthorn fruit. (See survival analysis below!)

```
ff1<-fruit.fly1%>%
  dplyr::filter(eclfac==1)

ggplot(ff1,aes(x=new.eclosions,colour=Host))+
  stat_ecdf(linewidth=1)+
  facet_wrap(~treatment)+T+
  scale_x_continuous(limits=c(0,300),
                     labels=seq(0,300,100),breaks=seq(0,300,100))+
  xlab("Days Until Adult Emergence (Days)")+ylab("ECDF")
```

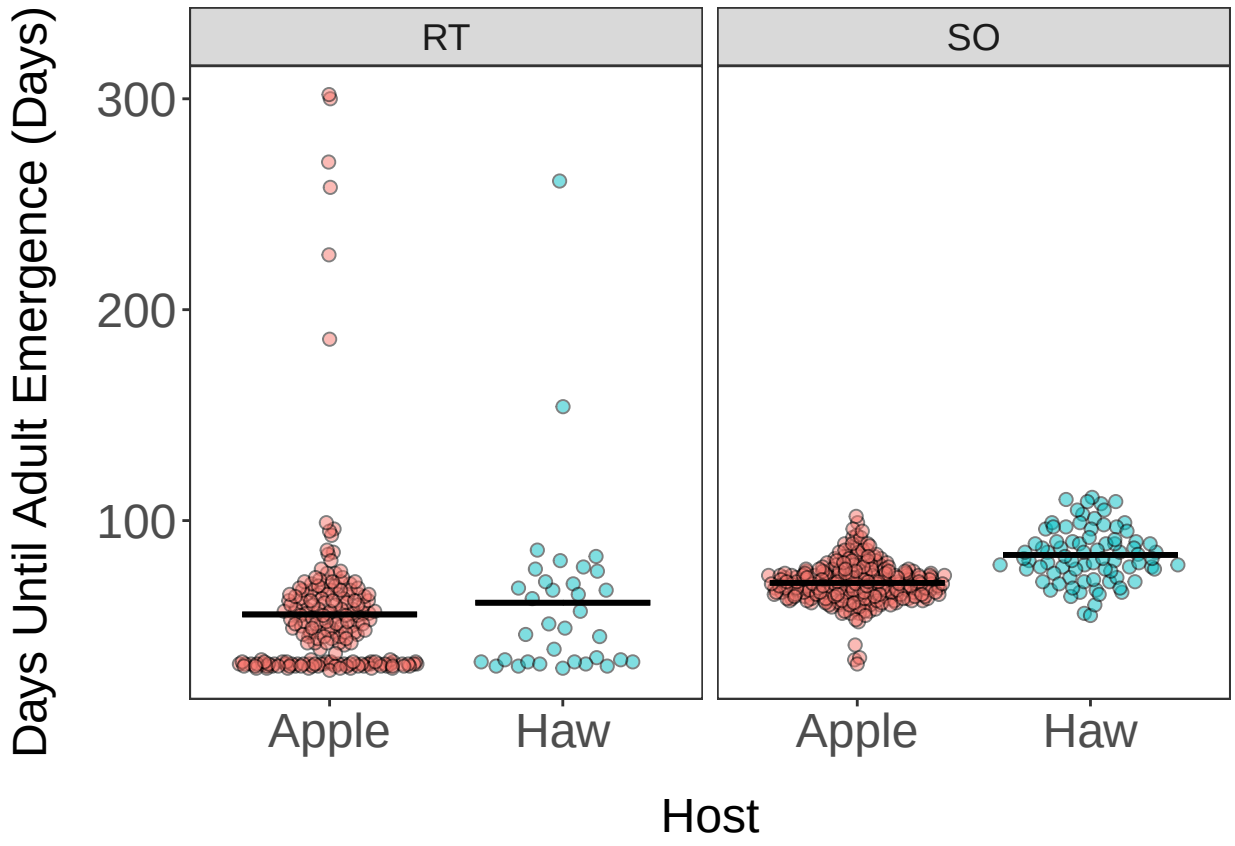
```
## Warning: Removed 1 rows containing non-finite values (‘stat_ecdf()‘).
```



Quasirandom plots

These types of plots are nice to show the distribution of the data. In this case, we have continuous data on the y axis and categorical on the x.

```
ggplot(ff1,aes(x=Host,y=new.eclosions,fill=Host))+
  geom_quasirandom(size=2,alpha=.5,
                   shape=21,colour="black")+ # change the points
#so that there is a black outline and color is filled in by host
  facet_wrap(~treatment)+
  ylab("Days Until Adult Emergence (Days)")+
  stat_summary(fun = mean, geom = "errorbar",
               aes(ymax =after_stat(y) , ymin = after_stat(y)),width = .75,lwd=1,colour="black")+
  T+ theme(legend.position="none")
```




```
#add average horizontal line, which adds a nice touch
```

It looks like there are two populations within the RT treatment. When conditions are optimal for growth (they're at room temperature and not diapausing), then there looks to be a subpopulation that emerges very quickly.

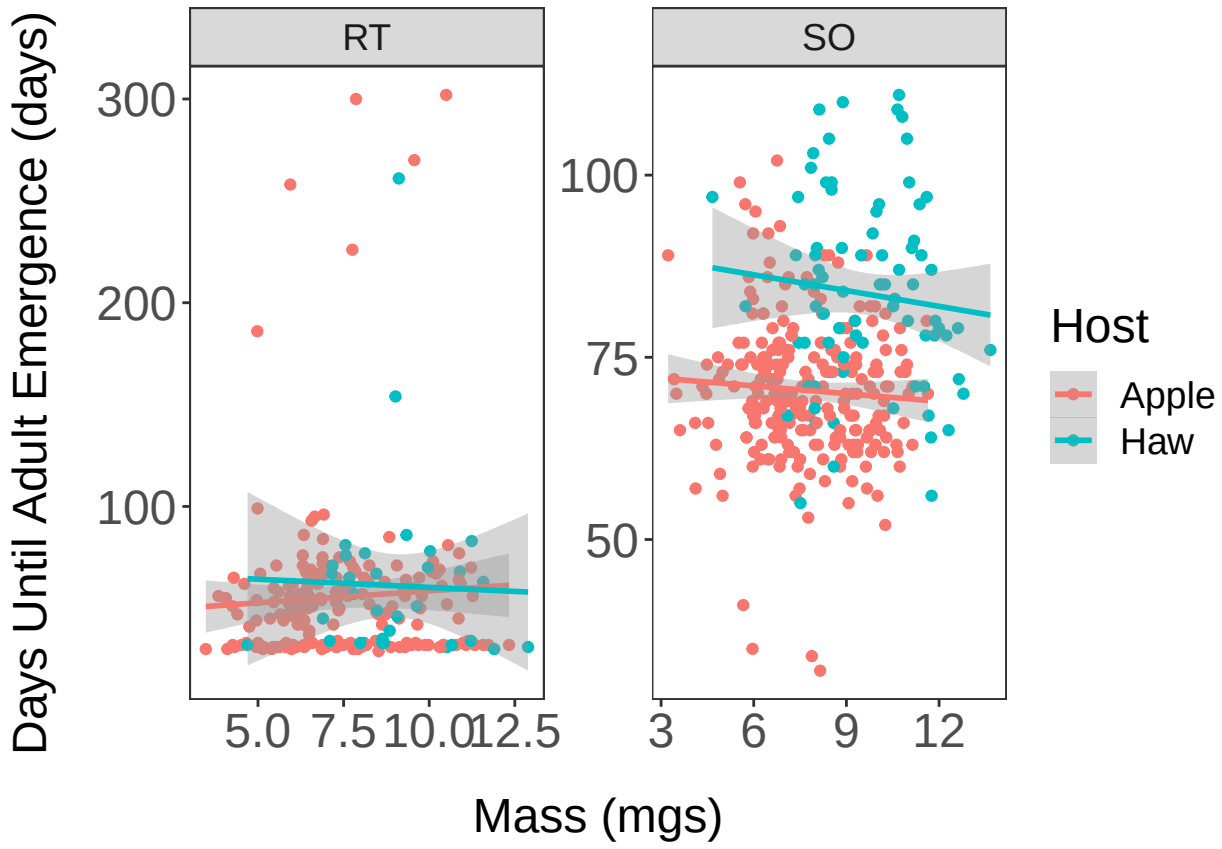
Scatter plots to display regressions

For illustrative purposes, I'm plotting mass with eclosion days just to show how to implement scatter plots.

```
ggplot(ff1,aes(x=mass_day10,y=new.eclosions,colour=Host))+
  geom_point()+stat_smooth(method="lm")+
#fits regression lines with standard errors
#and takes into account the Host when colour is specified
```

```
  facet_wrap(~treatment,scale="free")+
  ylab("Days Until Adult Emergence (days)")+
  xlab("Mass (mgs)")+T
```

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



Survival analysis

The data are best suited for a surival analysis because we have time of events. I'll fit a cox hazard proportional regression model to identify interactions between treatment and host fruit on eclosion timing.

```
# I need to convert Host into a factor
ff1$hfac<-factor(ifelse(ff1$Host=="Apple","1","0"))


#####################################

coxmod<-coxph(Surv(new.eclosions, eclfac) ~ Host*treatment,data=ff1)
summary(coxmod)
```

```
## Call:
## coxph(formula = Surv(new.eclosions, eclfac) ~ Host * treatment,
##     data = ff1)
```

```
##
##   n= 555, number of events= 555
##
##                          coef exp(coef) se(coef)      z Pr(>|z|)
## HostHaw             -0.4273    0.6523   0.1864 -2.292   0.0219 *
## treatmentSO         -0.7010    0.4961   0.1002 -6.997 2.62e-12 ***
## HostHaw:treatmentSO -0.2808    0.7552   0.2278 -1.233   0.2176
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## HostHaw                0.6523      1.533    0.4527    0.9400
## treatmentSO            0.4961      2.016    0.4077    0.6037
## HostHaw:treatmentSO    0.7552      1.324    0.4833    1.1801
##
## Concordance= 0.711  (se = 0.011 )
## Likelihood ratio test= 122  on 3 df,   p=<2e-16
## Wald test            = 117.4  on 3 df,   p=<2e-16
## Score (logrank) test = 128  on 3 df,   p=<2e-16
```

There are two main effects: host fruit and treatment. Compared to Apple, the Hawthorne fruit has a 34.77% lower eclosion (hazard) rate, which supports the ECDF figure where fruit flies on apple fruit had earlier eclosion timing. The simulated overwintering effect had a 53.39% lower eclosion (hazard) rate than the room temperature treatment. This makes sense because overwintering costs energy and energy levels become depleted afterwards, so development to eclosion is slower.

Machine Learning -> predicting host fruit from organismal features

Is it possible to predict host fruit based on the measurements we made? We'll create data partitions for training and then testing the model.

```
ff2<-ff1%>%
  dplyr::filter(treatment=="SO")%>%
  dplyr::select(hfac,mass_day10,resp_day15,new.eclosions)
  #just picking out variables that we want
intrain <- createDataPartition(y = ff2$hfac, p= 0.7, list = FALSE)
training <- ff2[intrain,]
testing <- ff2[-intrain,]

trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
#The "method" parameter defines the resampling method,
#in this demo we'll be using the repeatedcv or the repeated cross-validation method.

#The next parameter is the "number",
#this basically holds the number of resampling iterations.

#The "repeats " parameter contains the sets to
#compute for our repeated cross-validation.
#We are using setting number =10 and repeats =

svm_Linear <- train(hfac ~., data = training, method = "svmLinear",
trControl=trctrl, preProcess = c("center", "scale"), tuneLength = 10)
```

```
svm_Linear
```

```
## Support Vector Machines with Linear Kernel
##
## 227 samples
##   3 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (3), scaled (3)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 204, 204, 204, 204, 204, 205, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8359025  0.49038
##
## Tuning parameter 'C' was held constant at a value of 1
```

```
##let's see how well it predicts

test_pred <- predict(svm_Linear, newdata = testing)
confusionMatrix(table(test_pred, testing$hfac))
```

```
## Confusion Matrix and Statistics
##
##
## test_pred  0  1
##         0 12  3
##         1 12 69
##
##                Accuracy : 0.8438
##                  95% CI : (0.7554, 0.9098)
##     No Information Rate : 0.75
##     P-Value [Acc > NIR] : 0.01886
##
##                   Kappa : 0.5238
##
##  Mcnemar's Test P-Value : 0.03887
##
##             Sensitivity : 0.5000
##             Specificity : 0.9583
##          Pos Pred Value : 0.8000
##          Neg Pred Value : 0.8519
##              Prevalence : 0.2500
##          Detection Rate : 0.1250
##    Detection Prevalence : 0.1562
##       Balanced Accuracy : 0.7292
##
##        'Positive' Class : 0
##
```

Session info

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] survival_3.5-7   caret_6.0-94     lattice_0.21-8   ggbeeswarm_0.7.2
##  [5] lubridate_1.9.2  forcats_1.0.0    stringr_1.5.0    dplyr_1.1.2
##  [9] purrr_1.0.2      readr_2.1.4      tidyr_1.3.0      tibble_3.2.1
## [13] ggplot2_3.4.3    tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.0     timeDate_4022.108    vipor_0.4.5
##  [4] farver_2.1.1         fastmap_1.1.1        pROC_1.18.4
##  [7] digest_0.6.33        rpart_4.1.19         timechange_0.2.0
## [10] lifecycle_1.0.3      magrittr_2.0.3       kernlab_0.9-32
## [13] compiler_4.3.1       rlang_1.1.1          tools_4.3.1
## [16] utf8_1.2.3           yaml_2.3.7           data.table_1.14.8
## [19] knitr_1.43           labeling_0.4.2       plyr_1.8.8
## [22] withr_2.5.0          nnet_7.3-19          grid_4.3.1
## [25] stats4_4.3.1         fansi_1.0.4          e1071_1.7-13
## [28] colorspace_2.1-0     future_1.33.0        globals_0.16.2
## [31] scales_1.2.1         iterators_1.0.14     MASS_7.3-60
## [34] cli_3.6.1            rmarkdown_2.24       generics_0.1.3
## [37] rstudioapi_0.15.0    future.apply_1.11.0  reshape2_1.4.4
## [40] tzdb_0.4.0           proxy_0.4-27         splines_4.3.1
## [43] parallel_4.3.1       vctrs_0.6.3          hardhat_1.3.0
## [46] Matrix_1.6-1         hms_1.1.3            beeswarm_0.4.0
## [49] listenv_0.9.0        foreach_1.5.2        gower_1.0.1
## [52] recipes_1.0.7        glue_1.6.2           parallelly_1.36.0
## [55] codetools_0.2-19     stringi_1.7.12       gtable_0.3.4
## [58] munsell_0.5.0        pillar_1.9.0         htmltools_0.5.6
## [61] ipred_0.9-14         lava_1.7.2.1         R6_2.5.1
```

```
## [64] evaluate_0.21        highr_0.10           class_7.3-22
## [67] Rcpp_1.0.11          nlme_3.1-163         prodlim_2023.03.31
## [70] mgcv_1.9-0           xfun_0.40            pkgconfig_2.0.3
## [73] ModelMetrics_1.2.2.2
```