Histopathology Research Template

true

2019-11-08

Contents

```
knitr::opts_chunk$set(
    echo = TRUE, #change to TRUE
    eval = TRUE,
    message = FALSE,
    warning = FALSE,
    comment = NA,
    tidy = TRUE,
```

```
    fig.path = here::here("figs/")
)

# https://cran.r-project.org/web/packages/exploreR/vignettes/exploreR.html
# exploreR::reset()
```

1 Define Variable Types

2 Overview / Exploratory Data Analysis (EDA)

3 Information value

4 Feature imbalance

5 Memory usage

---

6 SmartEDA

Create HTML EDA report

Create a exploratory data analysis report in HTML format

```
ExpReport(Carseats,Target="Urban",label=NULL,theme="Default",op_file="test.html",op_dir=getwd(),sc=2,
sn=2,Rc="Yes")
```

Quantile-quantile plot for numeric variables

```
ExpOutQQ(CData,nlim=10,fname=NULL,Page=c(2,2),sample=4)
```

Parallel Co-ordinate plots

6.1 Defualt ExpParcoord funciton

```
ExpParcoord(CData,Group=NULL,Stsize=NULL,Nvar=c("Price","Income","Advertising","Population","Age",
"Education"))
```

6.2 With Stratified rows and selected columns only

ExpParcoord(CData,Group=“ShelveLoc”,Stsize=c(10,15,20),Nvar=c(“Price”,“Income”),Cvar=c(“Urban”,“US”)) ## Without stratification ExpParcoord(CData,Group=“ShelveLoc”,Nvar=c(“Price”,“Income”),Cvar=c(“Urban”,“US”),scale=NULL)

Exploratory analysis - Custom tables, summary statistics

Descriptive summary on all input variables for each level/combination of group variable. Also while running the analysis we can filter row/cases of the data.

```
ExpCustomStat(Carseats,Cvar=c("US","Urban","ShelveLoc"),gpby=FALSE)
ExpCustomStat(Carseats,Cvar=c("US","Urban"),gpby=TRUE,filt=NULL)
ExpCustomStat(Carseats,Cvar=c("US","Urban","ShelveLoc"),gpby=TRUE,filt=NULL)
ExpCustomStat(Carseats,Cvar=c("US","Urban"),gpby=TRUE,filt="Population>150")
ExpCustomStat(Carseats,Cvar=c("US","ShelveLoc"),gpby=TRUE,filt="Urban=='Yes' & Population>150")
```

ExpCustomStat(Carseats,Cvar=c("US","Urban","ShelveLoc","Education"),gpby=FALSE)

ExpCTable(Carseats,Target=NULL,clim=5,nlim=15,round=2,bin=NULL,per=F)

ExpCustomStat(Carseats,Cvar=c("US","Urban","ShelveLoc"),gpby=FALSE)

ExpCustomStat(Carseats,Cvar=c("US","Urban"),gpby=TRUE,filt=NULL)

ExpCustomStat(Carseats,Cvar=c("US","Urban","ShelveLoc"),gpby=TRUE,filt=NULL)

ExpCustomStat(Carseats,Cvar=c("US","Urban"),gpby=TRUE,filt="Population>150")

ExpCustomStat(Carseats,Cvar=c("US","ShelveLoc"),gpby=TRUE,filt="Urban=='Yes' & Population>150")

options(width = 150) ExpCustomStat(Carseats,Nvar=c("Population","Sales","CompPrice","Income"),stat = c('Count','mean','sum','var','sd','min','max','IQR'))

ExpCustomStat(Carseats,Nvar=c("Population","Sales","CompPrice","Income"),stat = c('min','p0.25','median','p0.75','max'))

options(width = 150) ExpCustomStat(Carseats,Nvar=c("Population","Sales","CompPrice","Income"),stat = c('Count','mean','sum','var','min','median','max'),filt="Urban=='Yes' ")

options(width=150) ExpCustomStat(Carseats,Nvar=c("Population","Sales","CompPrice","Income"),stat = c('Count','mean','sum','median','IQR'),filt="Urban=='Yes' & Population>150")

data_sam = Carseats[,] data_sam[sample(1:400,30),"Sales"] <- 999 data_sam[sample(1:400,20),"CompPrice"] <- -9 data_sam[sample(1:400,45),"Income"] <- 999 ExpCustomStat(data_sam,Nvar=c("Population","Sales","CompPrice","Income"),stat = c('Count','mean','sum','min'),filt="All %ni% c(999,-9)")

options(width = 150) ExpCustomStat(Carseats,Nvar=c("Population","Sales","CompPrice","Education","Income"),stat = c('Count','mean','sum','var','sd','IQR','median'),filt=c("ShelveLoc=='Good'Urban=='Yes'Price>=150A

options(width = 150) ExpCustomStat(Carseats,Cvar = c("Urban","ShelveLoc"), Nvar=c("Population","Sales"), stat = c('Count','Prop','mean','min','P0.25','median','p0.75','max'),gpby=FALSE)

options(width = 150) ExpCustomStat(Carseats,Cvar = c("Urban","US","ShelveLoc"), Nvar=c("CompPrice","Income"), stat = c('Count','Prop','mean','sum','PS','min','max','IQR','sd'), gpby = TRUE)

options(width = 150) ExpCustomStat(Carseats,Cvar = c("Urban","US","ShelveLoc"), Nvar=c("CompPrice","Income"), stat = c('Count','Prop','mean','sum','PS','median','IQR'), gpby = TRUE,filt="Urban=='Yes' ")

options(width = 150) data_sam = Carseats[,] data_sam[sample(1:400,30),"Sales"] <- 888 data_sam[sample(1:400,20),"CompPrice"] <- 999 data_sam[sample(1:400,45),"Income"] <- 999 ExpCustomStat(data_sam,Cvar = c("Urban","US","ShelveLoc"), Nvar=c("Sales","CompPrice","Income"), stat = c('Count','Prop','mean','sum','PS'), gpby = TRUE,filt="All %ni% c(888,999)")

ExpCustomStat(Carseats,Cvar = c("Urban","US"), Nvar=c("Population","Sales","CompPrice"), stat = c('Count','Prop','mean','sum','var','IQR'), filt=c("ShelveLoc=='Good'Urban=='Yes'Price>=150"))

options(width = 150) ExpCustomStat(Carseats,Cvar = c("Urban"), Nvar=c("Population","Sales"), stat = c('Count','Prop'),gpby=TRUE,dcast=TRUE)

##Frequency table for categorical variables ExpCustomStat(Carseats,Cvar=c("US","Urban","ShelveLoc"),gpby=FALSE)

##Crosstabulation between categorical variables ExpCustomStat(Carseats,Cvar=c("US","Urban"),gpby=TRUE,filt=NULL ExpCustomStat(Carseats,Cvar=c("US","Urban","ShelveLoc"),gpby=TRUE,filt=NULL)

##Adding filters for custom tables ExpCustomStat(Carseats,Cvar=c("US","Urban"),gpby=TRUE,filt="Population>150") ExpCustomStat(Carseats,Cvar=c("US","ShelveLoc"),gpby=TRUE,filt="Urban=='Yes' & Population>150")

6.3 Numeric variable summary

ExpCustomStat(Carseats,Nvar=c("Population","Sales","CompPrice","Income"),stat = c('Count','mean','sum','var','min','m
ExpCustomStat(Carseats,Nvar=c("Population","Sales","CompPrice","Income"),stat = c('min','p0.25','median','p0.75','max'))

6.4 Adding filters for complete data (like base Subset)

ExpCustomStat(Carseats,Nvar=c("Population","Sales","CompPrice","Income"),stat = c('Count','mean','sum','var'),filt="Urban=='Yes' ") ExpCustomStat(Carseats,Nvar=c("Population","Sales","CompPrice","Income"),stat = c('Count','mean','sum'),filt="Urban=='Yes' & Population>150")

6.5 Filter unique value from all the numeric variables

ExpCustomStat(data_sam,Nvar=c("Population","Sales","CompPrice","Income"),stat = c('Count','mean','sum','min'),filt='
%ni% c(999,-9)")

6.6 Adding filters at variable level

ExpCustomStat(Carseats,Nvar=c("Population","Sales","CompPrice","Education","Income"),stat = c('Count','mean','sum','var','sd','IQR','median'),filt=c("ShelveLoc=='Good'^Urban=='Yes'^Price>=150^^US=='Yes' "))

##Numerical summaries by category ##Variable summary report (One group variable) ExpCustomStat(Carseats,Cvar = c("Urban","ShelveLoc"), Nvar=c("Population","Sales"), stat = c('Count','Prop','mean','min','P0.25','median','p0.75','max'),gpby=FALSE)

##Variable summary report (More than One group variable) ExpCustomStat(Carseats,Cvar = c("Urban","US","ShelveLoc"), Nvar=c("CompPrice","Income"), stat = c('Count','Prop','mean','sum','PS','min','max','IQR','sd'), gpby = TRUE)

##Variable summary report (More than One group variable) with filter ExpCustomStat(Carseats,Cvar = c("Urban","US","ShelveLoc"), Nvar=c("CompPrice","Income"), stat = c('Count','Prop','mean','sum','PS','P0.25','median','p0.75'), gpby = TRUE,filt="Urban=='Yes' ") ExpCustomStat(data_sam,Cvar = c("Urban","US","ShelveLoc"), Nvar=c("Sales","CompPrice","Income"), stat = c('Count','Prop','mean','sum','PS'), gpby = TRUE,filt="All %ni% c(888,999)") ExpCustomStat(Carseats,Cvar = c("Urban","US"), Nvar=c("Population","Sales","CompPrice"), stat = c('Count','Prop','mean','sum','var','min','max'), filt=c("ShelveLoc=='Good'^Urban=='Yes'^

iris %>% mutate(sumVar = rowSums(.[1:4]))

iris %>% mutate(sumVar = rowSums(select(., contains("Sepal")))) %>% head

iris %>% mutate(sumVar = select(., contains("Sepal")) %>% rowSums()) %>% head

iRenameColumn.R

iSelectColumn.R

```
<= 22 Low
>= 23 & <= 41 Average
>=42 High
```

12 Descriptive Statistics

12.1 Table 1

Report Data

12.2 Categorical Variables

12.2.1 Split-Group Stats Categorical

12.2.2 Grouped Categorical

12.3 Continious Variables

12.3.1 Split-Group Stats Continious

12.3.2 Grouped Continious

13 Cross Tables

14 Plots

14.1 Categorical Variables

15 Plots

15.1 Continious Variables

16 Hypothesis Tests

16.1 Tests of Normality

16.2 Categorical

16.2.1 Chi-Square Cramer Association Predictive Power

16.3 Continious

16.4 Odds

17 ROC

18 Decision Tree

19 Survival Analysis

20 Pairwise comparison

21 Multivariate Analysis Survival

---

22 KM plot

explanatory = c(“perfor.factor”) dependent = “Surv(time, status)” colon_s %>% surv.plot(dependent, explanatory, xlab=“Time (days)”, pval=TRUE, legend=“none”)

Notes

Use Hmisc::label() to assign labels to variables for tables and plots.

label(colon_s$age.factor) = “Age (years)”

Export dataframe tables directly or to R Markdown using knitr::kable().

Note wrapper summary.missing() can be useful. Wraps mice::md.pattern.

colon_s %>% summary.missing(dependent, explanatory)

Where a multivariable model contains a subset of the variables specified in the full univariable set, this can be specified.

explanatory = c("age.factor", "sex.factor", "obstruct.factor", "perfor.factor") explanatory.multi = c("age.factor", "obstruct.factor") dependent = 'mort_5yr' colon_s %>% summarizer(dependent, explanatory, explanatory.multi)

Random effects.

e.g. lme4::glmer(dependent ~ explanatory + (1 | random_effect), family="binomial")

explanatory = c("age.factor", "sex.factor", "obstruct.factor", "perfor.factor") explanatory.multi = c("age.factor", "obstruct.factor") random.effect = "hospital" dependent = 'mort_5yr' colon_s %>% summarizer(dependent, explanatory, explanatory.multi, random.effect)

metrics=TRUE provides common model metrics.

colon_s %>% summarizer(dependent, explanatory, explanatory.multi, metrics=TRUE)

Cox proportional hazards

e.g. survival::coxph(dependent ~ explanatory)

explanatory = c("age.factor", "sex.factor", "obstruct.factor", "perfor.factor") dependent = "Surv(time, status)"

colon_s %>% summarizer(dependent, explanatory)

Rather than going all-in-one, any number of subset models can be manually added on to a summary.factorlist() table using summarizer.merge(). This is particularly useful when models take a long-time to run or are complicated.

Note requirement for glm.id=TRUE. fit2df is a subfunction extracting most common models to a dataframe.

explanatory = c("age.factor", "sex.factor", "obstruct.factor", "perfor.factor") explanatory.multi = c("age.factor", "obstruct.factor") random.effect = "hospital" dependent = 'mort_5yr'

23 Separate tables

colon_s %>% summary.factorlist(dependent, explanatory, glm.id=TRUE) -> example.summary

colon_s %>% glmuni(dependent, explanatory) %>% fit2df(estimate.suffix=" (univariable)") -> example.univariable

colon_s %>% glmmulti(dependent, explanatory) %>% fit2df(estimate.suffix=" (multivariable)") -> example.multivariable

colon_s %>% glmmixed(dependent, explanatory, random.effect) %>% fit2df(estimate.suffix=" (multilevel") -> example.multilevel

24 Pipe together

example.summary %>% summarizer.merge(example.univariable) %>% summarizer.merge(example.multivariable) %>% summarizer.merge(example.multilevel) %>% select(-c(glm.id, index)) -> example.final example.final

Cox Proportional Hazards example with separate tables merged together.

explanatory = c("age.factor", "sex.factor", "obstruct.factor", "perfor.factor") explanatory.multi = c("age.factor", "obstruct.factor") dependent = "Surv(time, status)"

25 Separate tables

colon_s %>% summary.factorlist(dependent, explanatory, glm.id=TRUE) -> example2.summary

colon_s %>% coxphuni(dependent, explanatory) %>% fit2df(estimate.suffix=" (univariable)") -> example2.univariable

colon_s %>% coxphmulti(dependent, explanatory.multi) %>% fit2df(estimate.suffix=" (multivariable)") -> example2.multivariable

26 Pipe together

example2.summary %>% summarizer.merge(example2.univariable) %>% summarizer.merge(example2.multivariable) %>% select(-c(glm.id, index)) -> example2.final example2.final

27 OR plot

explanatory = c("age.factor", "sex.factor", "obstruct.factor", "perfor.factor") dependent = 'mort_5yr' colon_s %>% or.plot(dependent, explanatory) # Previously fitted models (`glmmulti()` or `glmmixed()`) can be provided directly to `glmfit`

28 HR plot (not fully tested)

explanatory = c("age.factor", "sex.factor", "obstruct.factor", "perfor.factor") dependent = "Surv(time, status)" colon_s %>% hr.plot(dependent, explanatory, dependent_label = "Survival") # Previously fitted models (`coxphmulti`) can be provided directly using `coxfit`

29 ANOVA

Some Text ile sağkalım açısından bir ilişki bulunmamıştır (p = 0.22).

Some Text ile sağkalım açısından bir ilişki bulunmamıştır (p = 0.22).

Some Text

İstatistik Metod:, , Sürekli verilerin ortalama, standart sapma, median, minimum ve, maksimum değerleri verildi. Kategorik veriler ve gruplanan sürekli, veriler için frekans tabloları oluşturuldu. Genel sağkalım, analizinde ölüm tarihi ve son başvuru tarihi hasta dosyalarından, elde edildi. Sağkalım analizinde Kaplan-Meier grafikleri„ Log-rank testi ve Cox-Regresyon testleri uygulandı. Analizler, R-project (version 3.6.0) ve RStudio ile survival ve finalfit, paketleri kullanılarak yapıldı. p değeri 0.05 düzeyinde anlamlı, olarak kabul edildi., , R Core Team (2019). R: A language and environment for statistical, computing. R Foundation for Statistical Computing, Vienna„ Austria. URL https://www.R-project.org/., , Therneau T (2015). A Package for Survival Analysis in S. version, 2.38, https://CRAN.R-project.org/package=survival, , Terry M. Therneau, Patricia M. Grambsch (2000). Modeling Survival, Data: Extending the Cox Model. Springer, New York. ISBN, 0-387-98784-3., , Ewen Harrison, Tom Drake and Riinu Ots (2019). finalfit: Quickly, Create Elegant Regression Results Tables and Plots when Modelling., R package version 0.9.6. https://github.com/ewenharrison/finalfit

İstatistik Metod:, , Sürekli verilerin ortalama, standart sapma, median, minimum ve, maksimum değerleri verildi. Kategorik veriler ve gruplanan sürekli, veriler için frekans tabloları oluşturuldu. Genel sağkalım, analizinde ölüm tarihi ve son başvuru tarihi hasta dosyalarından, elde edildi. Sağkalım analizinde Kaplan-Meier grafikleri„ Log-rank testi ve Cox-Regresyon testleri uygulandı. Analizler, R-project (version 3.6.0) ve RStudio ile survival ve finalfit, paketleri kullanılarak yapıldı. p değeri 0.05 düzeyinde anlamlı, olarak kabul edildi., , R Core Team (2019). R: A language and environment for statistical, computing. R Foundation for Statistical Computing, Vienna„ Austria. URL https://www.R-project.org/., , Therneau T (2015). A Package for Survival Analysis in S. version, 2.38, https://CRAN.R-project.org/package=survival, , Terry M. Therneau, Patricia M. Grambsch (2000). Modeling Survival, Data: Extending the Cox Model. Springer, New York. ISBN, 0-387-98784-3., , Ewen Harrison, Tom Drake and Riinu Ots (2019). finalfit: Quickly, Create Elegant Regression Results Tables and Plots when Modelling., R package version 0.9.6. https://github.com/ewenharrison/finalfit

Text Here

Text Here

30 Save Final Data

```
saved data after analysis to/Users/serdarbalciold/histopathology-template/data/histopathology-
template2019-11-08.xlsx : 2019-11-08 23:38:35
```

31 Final Data Summary

32 Software and Libraries Used

```
To cite R in publications use:

  R Core Team (2019). R: A language and environment for
  statistical computing. R Foundation for Statistical Computing,
  Vienna, Austria. URL https://www.R-project.org/.

A BibTeX entry for LaTeX users is

  @Manual{,
    title = {R: A Language and Environment for Statistical Computing},
    author = {{R Core Team}},
    organization = {R Foundation for Statistical Computing},
    address = {Vienna, Austria},
    year = {2019},
    url = {https://www.R-project.org/},
  }

We have invested a lot of time and effort in creating R, please
cite it when using it for data analysis. See also
'citation("pkgname")' for citing R packages.
```

The jamovi project (2019). jamovi. (Version 0.9) [Computer Software]. Retrieved from https://www.jamovi.org. R Core Team (2018). R: A Language and envionment for statistical computing. [Computer software]. Retrieved from https://cran.r-project.org/. Fox, J., & Weisberg, S. (2018). car: Companion to Applied Regression. [R package]. Retrieved from https://cran.r-project.org/package=car.

32.1 data[order(data$References),]

Ewen Harrison, Tom Drake and Riinu Ots (2019). finalfit: Quickly Create Elegant Regression Results Tables and Plots when Modelling. R package version 0.9.6. https://github.com/ewenharrison/finalfit

Hadley Wickham and Jennifer Bryan (2019). readxl: Read Excel Files. R package version 1.3.1. https://CRAN.R-project.org/package=readxl

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2019). dplyr: A Grammar of Data Manipulation. R package version 0.8.3. https://CRAN.R-project.org/package=dplyr

Makowski, D. & Lüdecke, D. (2019). The report package for R: Ensuring the use of best practices for results reporting. CRAN. Available from https://github.com/easystats/report. doi: .

Patil I (2018). *ggstatsplot: 'ggplot2' Based Plots withStatistical Details*. doi: 10.5281/zenodo.2074621 (URL:https://doi.org/10.5281/zenodo.2074621), <URL:https://CRAN.R-project.org/package=ggstatsplot>.

Rinker, T. W. (2018). wakefield: Generate Random Data. version 0.3.3. Buffalo, New York. https://github.com/trinker/wakefield

Sam Firke (2019). janitor: Simple Tools for Examining and Cleaning Dirty Data. R package version 1.2.0. https://CRAN.R-project.org/package=janitor


```
To cite package 'tidyverse' in publications use:

  Hadley Wickham (2017). tidyverse: Easily Install and Load the
  'Tidyverse'. R package version 1.2.1.
  https://CRAN.R-project.org/package=tidyverse

A BibTeX entry for LaTeX users is
```

```
  @Manual{,
    title = {tidyverse: Easily Install and Load the 'Tidyverse'},
    author = {Hadley Wickham},
    year = {2017},
    note = {R package version 1.2.1},
    url = {https://CRAN.R-project.org/package=tidyverse},
  }


To cite package 'readxl' in publications use:

  Hadley Wickham and Jennifer Bryan (2019). readxl: Read Excel
  Files. R package version 1.3.1.
  https://CRAN.R-project.org/package=readxl

A BibTeX entry for LaTeX users is

  @Manual{,
    title = {readxl: Read Excel Files},
    author = {Hadley Wickham and Jennifer Bryan},
    year = {2019},
    note = {R package version 1.3.1},
    url = {https://CRAN.R-project.org/package=readxl},
  }


To cite package 'janitor' in publications use:

  Sam Firke (2019). janitor: Simple Tools for Examining and
  Cleaning Dirty Data. R package version 1.2.0.
  https://CRAN.R-project.org/package=janitor

A BibTeX entry for LaTeX users is

  @Manual{,
    title = {janitor: Simple Tools for Examining and Cleaning Dirty Data},
    author = {Sam Firke},
    year = {2019},
    note = {R package version 1.2.0},
    url = {https://CRAN.R-project.org/package=janitor},
  }


To cite in publications use:

  Makowski, D. & Lüdecke, D. (2019). The report package for R:
  Ensuring the use of best practices for results reporting. CRAN.
  Available from https://github.com/easystats/report. doi: .

A BibTeX entry for LaTeX users is

  @Article{,
    title = {The report package for R: Ensuring the use of best practices for results reporting},
    author = {{Makowski} and {Dominique} and {Lüdecke} and {Daniel}},
    journal = {CRAN},
```

```
    year = {2019},
    note = {R package},
    url = {https://github.com/easystats/report},
  }


To cite package 'finalfit' in publications use:

  Ewen Harrison, Tom Drake and Riinu Ots (2019). finalfit: Quickly
  Create Elegant Regression Results Tables and Plots when
  Modelling. R package version 0.9.6.
  https://github.com/ewenharrison/finalfit

A BibTeX entry for LaTeX users is

  @Manual{,
    title = {finalfit: Quickly Create Elegant Regression Results Tables and Plots when
Modelling},
    author = {Ewen Harrison and Tom Drake and Riinu Ots},
    year = {2019},
    note = {R package version 0.9.6},
    url = {https://github.com/ewenharrison/finalfit},
  }


Patil I (2018). _ggstatsplot: 'ggplot2' Based Plots with
Statistical Details_. doi: 10.5281/zenodo.2074621 (URL:
https://doi.org/10.5281/zenodo.2074621), <URL:
https://CRAN.R-project.org/package=ggstatsplot>.

A BibTeX entry for LaTeX users is

  @Manual{,
    title = {ggstatsplot: 'ggplot2' Based Plots with Statistical Details},
    author = {Indrajeet Patil},
    year = {2018},
    url = {https://CRAN.R-project.org/package=ggstatsplot},
    doi = {10.5281/zenodo.2074621},
  }
```

33 Session Info

```
R version 3.6.0 (2019-04-26)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS  10.15.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] wakefield_0.3.3   ggstatsplot_0.1.2 finalfit_0.9.6    report_0.1.0
[5] janitor_1.2.0     readxl_1.3.1      dplyr_0.8.3

loaded via a namespace (and not attached):
  [1] estimability_1.3          SparseM_1.77
  [3] coda_0.19-3               tidyr_1.0.0
  [5] ggplot2_3.2.1             acepack_1.4.1
  [7] knitr_1.25                multcomp_1.4-10
  [9] data.table_1.12.6         rpart_4.1-15
 [11] inline_0.3.15             generics_0.0.2
 [13] callr_3.3.2               cowplot_1.0.0
 [15] TH.data_1.0-10            mice_3.6.0
 [17] future_1.14.0             webshot_0.5.1
 [19] xml2_1.2.2                httpuv_1.5.2
 [21] StanHeaders_2.19.0        assertthat_0.2.1
 [23] WRS2_1.0-0                xfun_0.10
 [25] hms_0.5.1                 evaluate_0.14
 [27] promises_1.0.1            DEoptimR_1.0-8
 [29] htmlwidgets_1.5.1         mcmc_0.9-6
 [31] reshape_0.8.8             stats4_3.6.0
 [33] paletteer_0.2.1           purrr_0.3.3
 [35] ellipsis_0.3.0            rcompanion_2.3.0
 [37] backports_1.1.5           insight_0.6.0
 [39] ggcorrplot_0.1.3          MCMCpack_1.4-4
 [41] libcoin_1.0-5             jmvcore_1.0.0
 [43] vctrs_0.2.0               quantreg_5.51
 [45] here_0.1                  sjlabelled_1.1.1
 [47] abind_1.4-5               metaBMA_0.6.2
 [49] robustbase_0.93-5         checkmate_1.9.4
 [51] emmeans_1.4.1             prettyunits_1.0.2
 [53] mnormt_1.5-5              cluster_2.1.0
 [55] lazyeval_0.2.2            crayon_1.3.4
 [57] pkgconfig_2.0.3           nlme_3.1-141
 [59] statsExpressions_0.1.1    nnet_7.3-12
 [61] rlang_0.4.1               globals_0.12.4
 [63] lifecycle_0.1.0           mitml_0.3-7
 [65] miniUI_0.1.1.1            groupedstats_0.0.9
```

```
 [67] skimr_1.0.7               LaplacesDemon_16.1.1
 [69] MatrixModels_0.4-1        sandwich_2.5-1
 [71] EMT_1.1                   modelr_0.1.5
 [73] cellranger_1.1.0          rprojroot_1.3-2
 [75] matrixStats_0.55.0        broomExtra_0.0.5
 [77] lmtest_0.9-37             Matrix_1.2-17
 [79] loo_2.1.0                 mc2d_0.1-18
 [81] carData_3.0-2             boot_1.3-23
 [83] zoo_1.8-6                 pan_1.6
 [85] base64enc_0.1-3           processx_3.4.1
 [87] viridisLite_0.3.0         rjson_0.2.20
 [89] parameters_0.2.0.1        ggExtra_0.9
 [91] stringr_1.4.0             multcompView_0.1-7
 [93] coin_1.3-1                robust_0.4-18.1
 [95] readr_1.3.1               ggsignif_0.6.0
 [97] scales_1.0.0              magrittr_1.5
 [99] plyr_1.8.4                compiler_3.6.0
[101] rstantools_2.0.0          kableExtra_1.1.0
[103] RColorBrewer_1.1-2        lme4_1.1-21
[105] rrcov_1.4-7               cli_1.1.0
[107] listenv_0.7.0             pbapply_1.4-2
[109] ps_1.3.0                  TMB_1.7.15
[111] Brobdingnag_1.2-6         htmlTable_1.13.2
[113] formatR_1.7               Formula_1.2-3
[115] MASS_7.3-51.4             mgcv_1.8-29
[117] tidyselect_0.2.5          stringi_1.4.3
[119] forcats_0.4.0             yaml_2.2.0
[121] latticeExtra_0.6-28       ggrepel_0.8.1
[123] bridgesampling_0.7-2      grid_3.6.0
[125] manipulate_1.0.1          tools_3.6.0
[127] parallel_3.6.0            rio_0.5.16
[129] rstudioapi_0.10           foreign_0.8-72
[131] gridExtra_2.3             pairwiseComparisons_0.1.1
[133] digest_0.6.22             shiny_1.3.2
[135] explore_0.5.1             nortest_1.0-4
[137] jmv_0.9.6.1               Rcpp_1.0.2
[139] car_3.0-3                 broom_0.5.2
[141] metafor_2.1-0             ez_4.4-0
[143] BayesFactor_0.9.12-4.2    performance_0.3.0
[145] later_0.8.0               writexl_1.1
[147] httr_1.4.1                psych_1.8.12
[149] sjstats_0.17.6            colorspace_1.4-1
[151] rvest_0.3.4               splines_3.6.0
[153] expm_0.999-4              fit.models_0.5-14
[155] xtable_1.8-4              nloptr_1.2.1
[157] rstan_2.19.2              zeallot_0.1.0
[159] modeltools_0.2-22         R6_2.4.0
[161] broom.mixed_0.2.4         Hmisc_4.2-0
[163] pillar_1.4.2              htmltools_0.4.0
[165] mime_0.7                  glue_1.3.1
[167] minqa_1.2.4               DT_0.9
[169] codetools_0.2-16          jomo_2.6-9
[171] pkgbuild_1.0.6            pcaPP_1.9-73
[173] mvtnorm_1.0-11            furrr_0.1.0
```

```
[175] lattice_0.20-38          tibble_2.1.3
[177] curl_4.2                 DescTools_0.99.28
[179] gtools_3.8.1             logspline_2.1.13
[181] zip_2.0.4                openxlsx_4.1.0.1
[183] survival_2.44-1.1        rmarkdown_1.16
[185] munsell_0.5.0            rsample_0.0.5
[187] sjmisc_2.8.1             haven_2.1.1
[189] reshape2_1.4.3           gtable_0.3.0
[191] bayestestR_0.3.0
```

34 Notes

Last update on 2019-11-08 23:38:35

Serdar Balci, MD, Pathologist
drserdarbalci@gmail.com
https://rpubs.com/sbalci/CV