



Biomedical Data Science with **rtemis**

Efstathios (Stathis) D Gennatas, MBBS AICSM PhD
Assistant Professional Researcher
University of California, San Francisco
Efstathios.Gennatas@ucsf.edu
egenn.github.io



Introduction Statistics / Machine Learning / Data Science

Context ML for Precision Medicine

Main **rtemis:** ML for all

Goals

Implementation

Supervised Learning

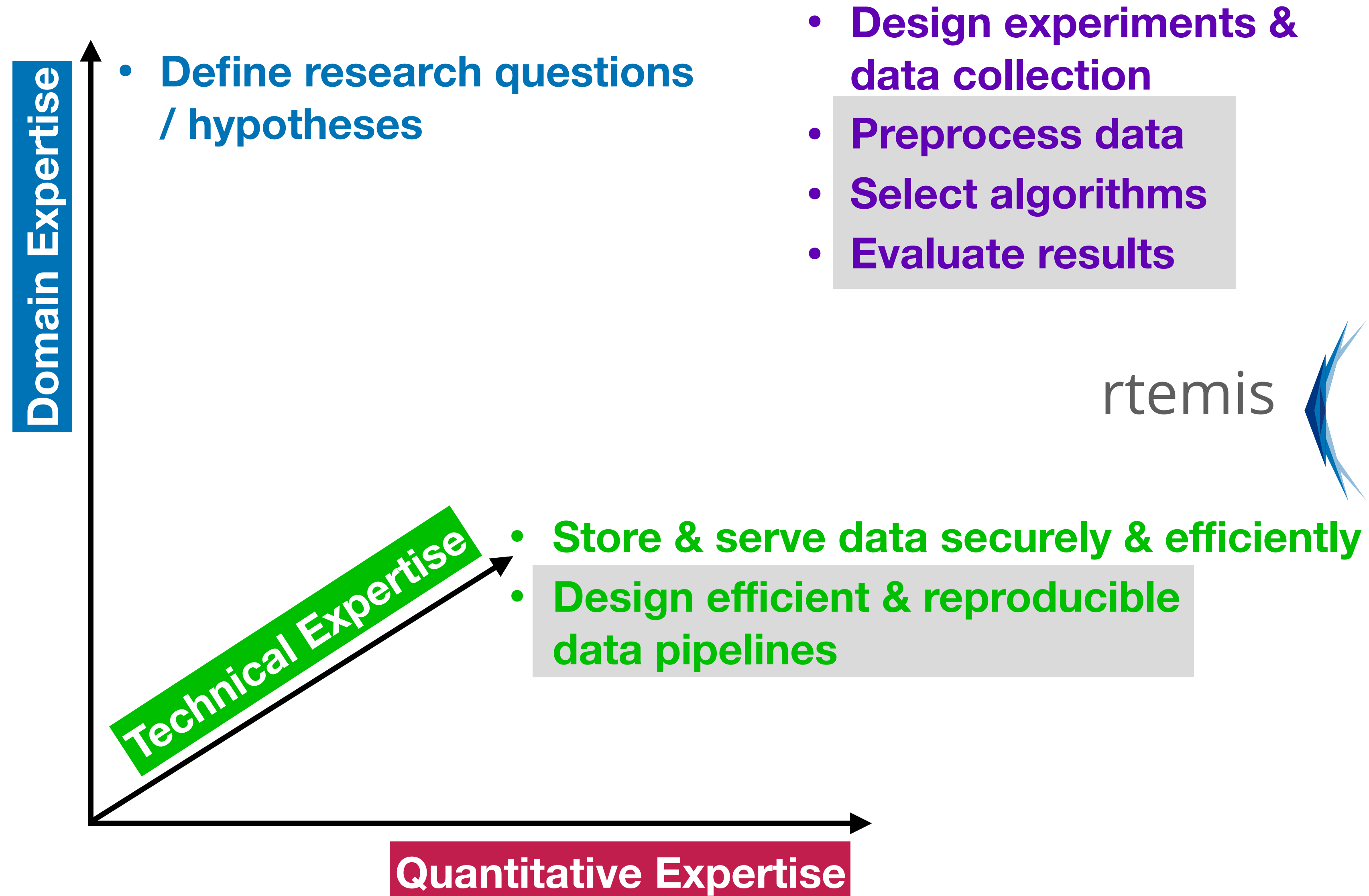
Lab Online Documentation

GitHub

Core Algorithms

Example pipeline walkthrough

[Story]



What is a statistical model?

A mathematical model, which builds a **mapping from features to outcome**

What is the purpose of a statistical model?

1. Learn something about the world

2. Predict the future

Interpretability

Accuracy

There is often a trade-off between Accuracy & Interpretability

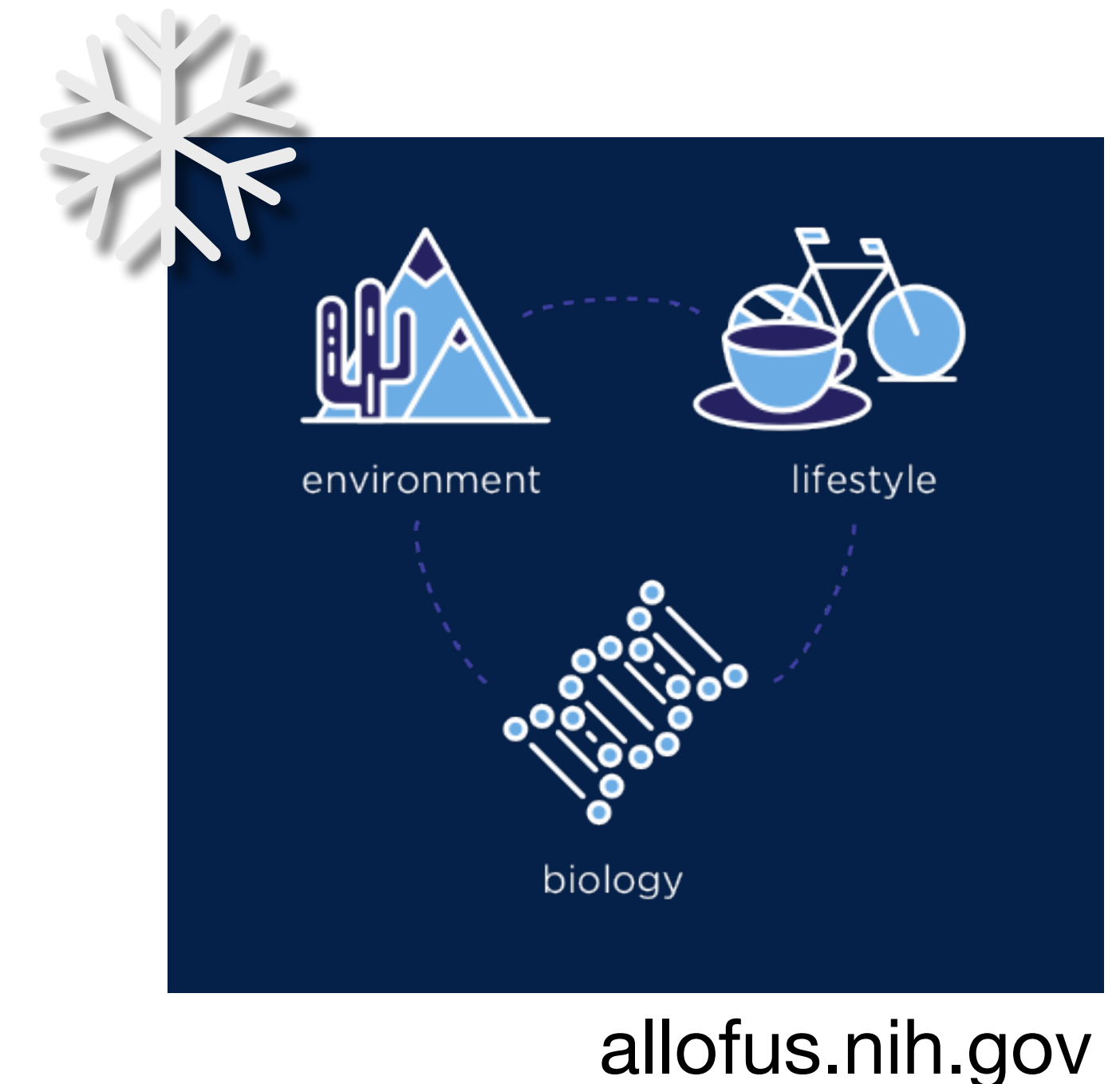
Precision Medicine: The promise

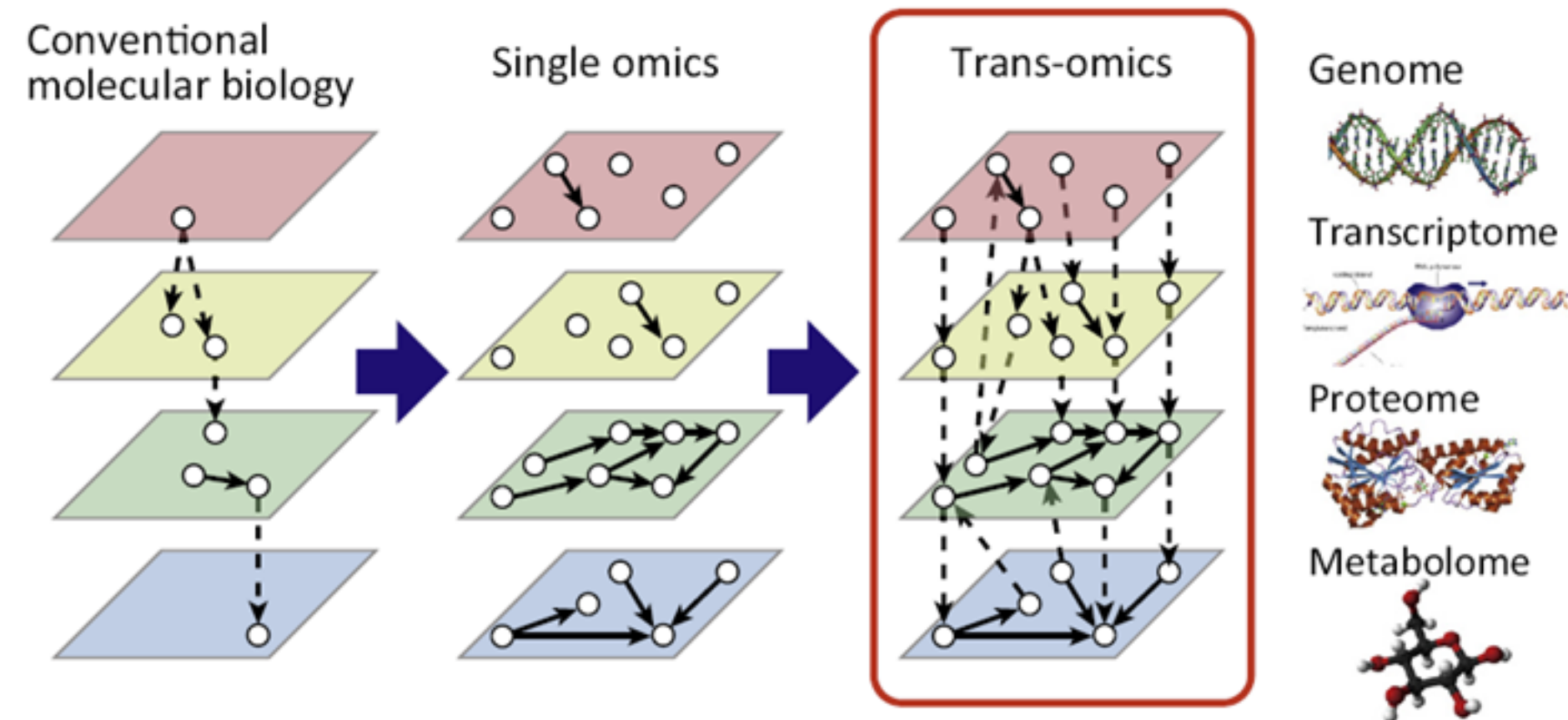
The need for precision medicine is clearer than ever today.
Each person is the result of a unique **genome + environment**,
resulting in similarly unique:

- **risk** of developing disease
- clinical **presentation**
- **response** to treatment

According to the NIH Precision Medicine Initiative,
precision medicine is "***an emerging approach for disease treatment and prevention that takes into account individual variability in genes, environment, and lifestyle for each person.***"

► i.e. **patient stratification** at each point of care





Yuki K, et al 2016

Electronic Health Records

- Demographics
- Medical, Family, Social, Drug History
- Clinical examination

Imaging

- Xray, CT, PET, MRI, Ultrasound

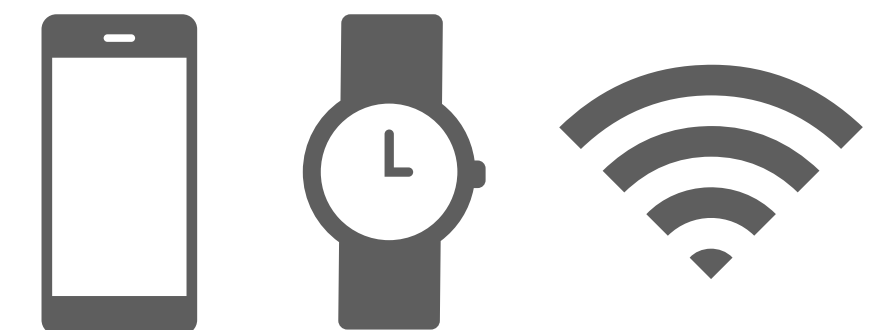


Pathology & Laboratory medicine / Systems Biology

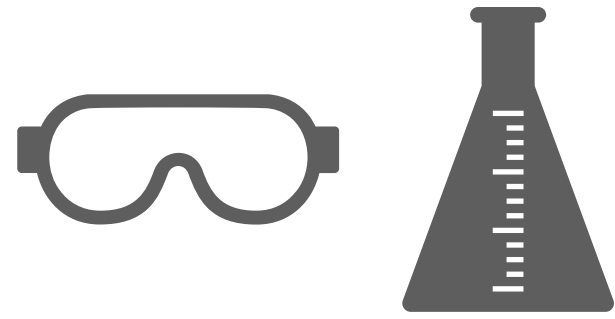
- Histology, Microbiology, Chemistry
- Genomics
- Transcriptomics
- Proteomics
- Metabonomics
- Lipidomics

Smart devices & wearables

- Activity, temperature, blood pressure, weight, sleep

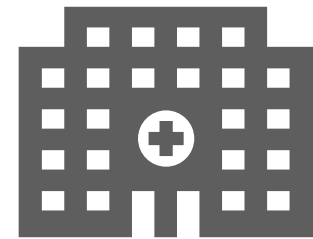


ML can benefit all aspects of medicine



Basic research: making sense of experimental data

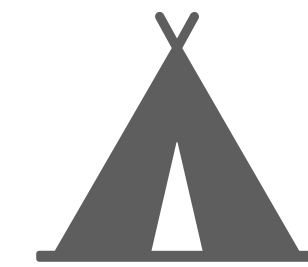
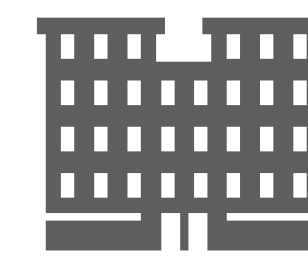
- Extract insights on disease pathophysiology
- Guide new treatment discovery



Healthcare: predicting outcomes

- Prevention
- Diagnosis
- Treatment
- Prognosis

Settings



- Everyday home & work environment
- Primary care
- Outpatient office
- Emergency Department
- Inpatient ward; Intensive Care

Teen's Life Saved by Apple Watch That Alerted Her Heart Condition

NEWS 8:23 AM PDT, May 4, 2018 - Inside Edition Staff

HEALTH

This Woman's Apple Watch Helped Detect Her Severe Thyroid Problem

Heather Hendershot had no symptoms and felt fine — but her Apple Watch indicated that she might have a health problem.

Pathways for therapeutics development:

- Basic research ► Characterization of disease mechanisms ► Design of treatment agents
- Serendipitous / trial & error discovery of pharmaceuticals

ML can help speed up basic and translational research

or

bypass it, mapping data to outcomes without a
complete understanding of the underlying pathophysiology

Toughest & most important goals in Medicine:

Disease prevention

Diagnosis of presymptomatic disease

Prediction of critical clinical events

Chronic

Acute

ML in medicine will help bring:

- Improved clinical decision **accuracy**
- Increased **efficiency** of medical services delivery
- Enhanced **access** to healthcare

Precision Medicine: The challenge

Advances in biomedical and clinical science are helping generate an increasing **volume** and **variety** of data, at an increasing **velocity**, though often of uncertain **veracity**

Along with increased requirements for **data warehousing**, **data security**, and **privacy control**, this raises the need for sophisticated analytic methods to

- **extract insights on disease pathophysiology**
- **guide new treatment discovery**
- **inform clinical decision making**

Group-level hypothesis testing is being complemented with **subject-level predictive modeling**

Mass-univariate analyses of **unimodal data** are increasingly complemented with **multivariate analyses** of **high-dimensional, multimodal data**.

Biomedical and clinical datasets present multiple data science challenges

- Small sample sizes
- Rare cases
- Missing Data
- High noise in predictors and outcomes
- Hidden confounders
- Lack of controls
- Bias in clinical practice
- Need for continuous model update

Biomedical and clinical datasets present multiple data science challenges

- Small sample sizes
- Rare cases
- Missing Data
- High noise in predictors and outcomes
- Hidden confounders
- Lack of controls
- Bias in clinical practice
- Need for continuous model update

Mostly affects training performance

Mostly affects testing performance

An evaluation of machine-learning methods for predicting pneumonia mortality

Gregory F. Cooper^{a,*}, Constantin F. Aliferis^a, Richard Ambrosino^a,
John Aronis^b, Bruce G. Buchanan^b, Richard Caruana^c,
Michael J. Fine^d, Clark Glymour^e, Geoffrey Gordon^c,
Barbara H. Hanusa^d, Janine E. Janosky^f, Christopher Meek^e,
Tom Mitchell^c, Thomas Richardson^e, Peter Spirtes^e

1997

Predicting dire outcomes of patients with community acquired pneumonia

Gregory F. Cooper^{a,*}, Vijoy Abraham^b, Constantin F. Aliferis^c, John M. Aronis^d,
Bruce G. Buchanan^e, Richard Caruana^f, Michael J. Fine^g, Janine E. Janosky^h,
Gary Livingstonⁱ, Tom Mitchell^j, Stefano Monti^k, Peter Spirtes^{j,l}

2005

Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission

Rich Caruana
Microsoft Research
rcaruana@microsoft.com

Yin Lou
LinkedIn Corporation
ylou@linkedin.com

Johannes Gehrke
Microsoft
johannes@microsoft.com

2015

Paul Koch
Microsoft Research
paulkoch@microsoft.com

Marc Sturm
NewYork-Presbyterian Hospital
mas9161@nyp.org

Noémie Elhadad
Columbia University
noemie.elhadad@columbia.edu

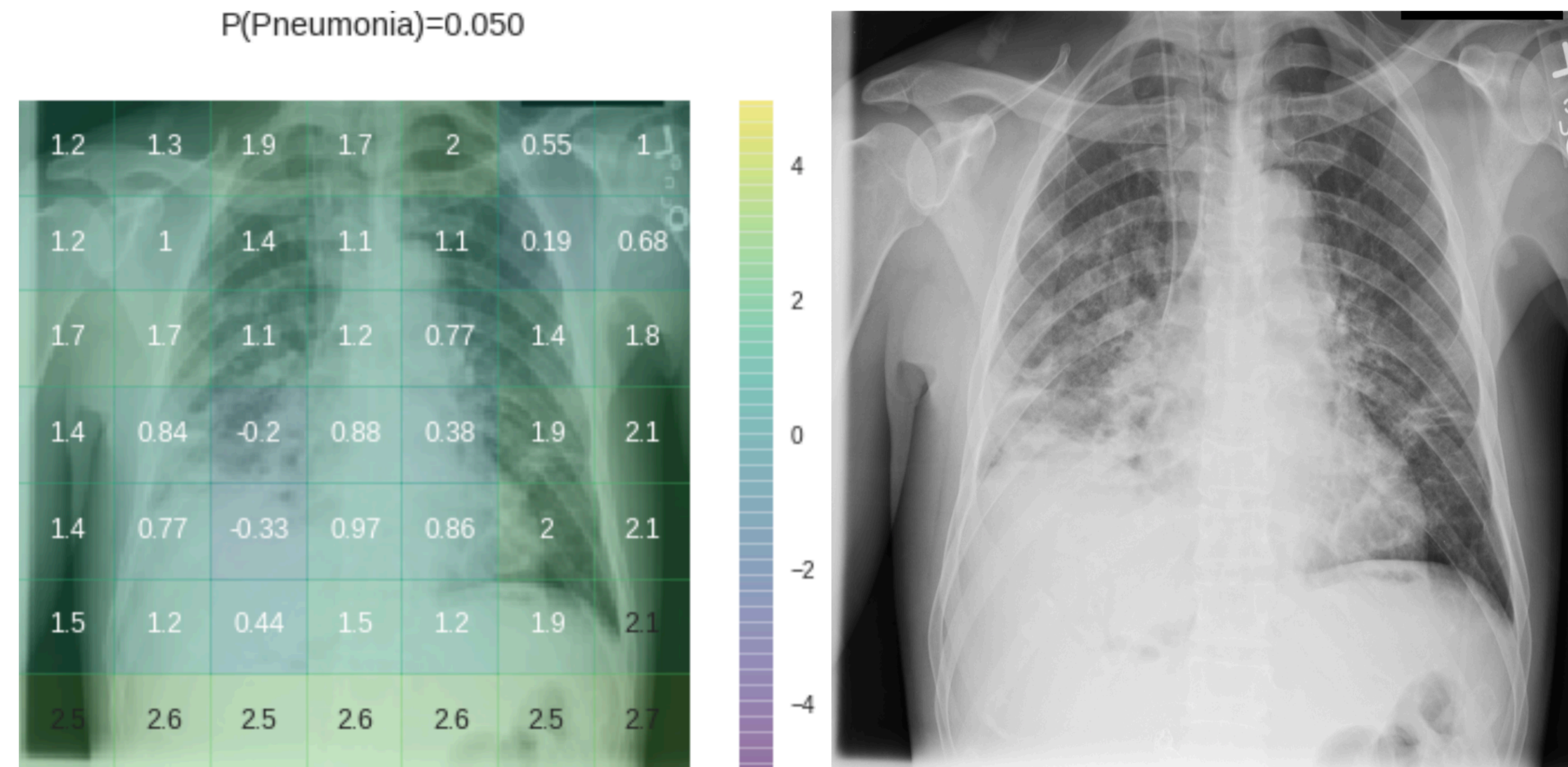
Hidden confounders are a common and big problem in medical data

Model predicting mortality risk of patients presenting to ER with pneumonia

- Multiple algorithms compared, black box neural network performed the best
- Rule-based model revealed a common problem
- Patients with asthma labeled as low risk of mortality from pneumonia
- Interpretation can be misleading unless reviewed by expert:
 - Missing confounders: patients received more aggressive treatment, faster
- Models worked just as expected: minimized cross-validation loss using correlation structure of data

Out of context, and sample, model predictions can be misleading and fatal

“Superhuman” neural network fails to detect pneumonia by weighing image artifacts over pathology

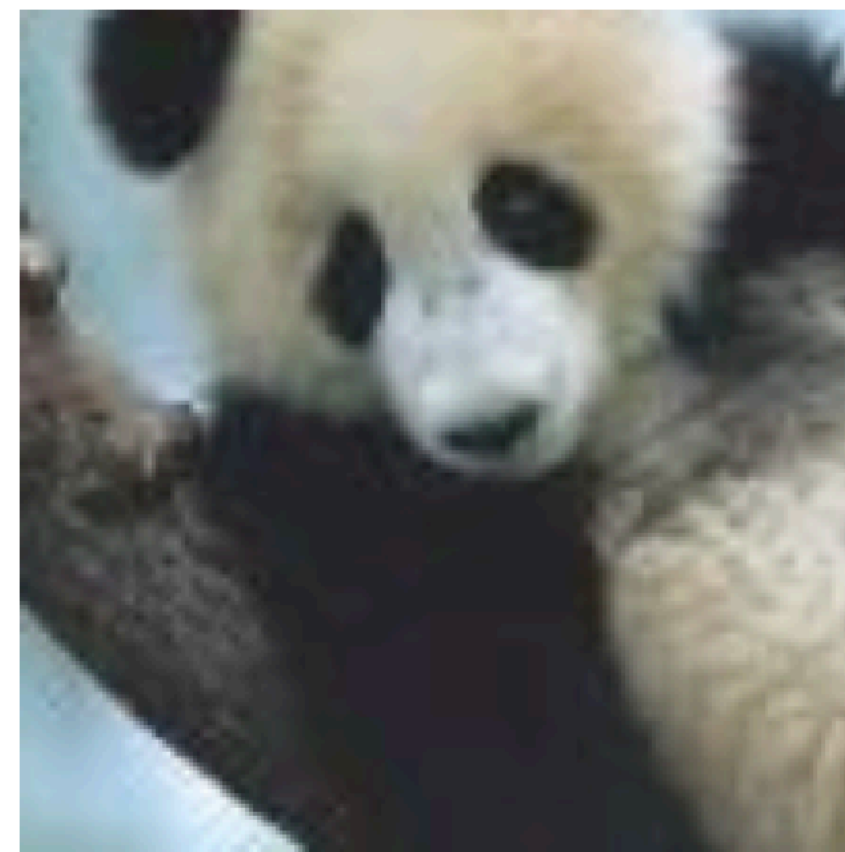


Patient with pneumonia: heatmap of CNN on left, original image on right

<https://medium.com/@jrzech/what-are-radiological-deep-learning-models-actually-learning-f97a546c5b98>

<https://arxiv.org/pdf/1807.00431.pdf>

Neural networks can perform well without necessarily learning something “real”



x

“panda”

57.7% confidence

$+ .007 \times$

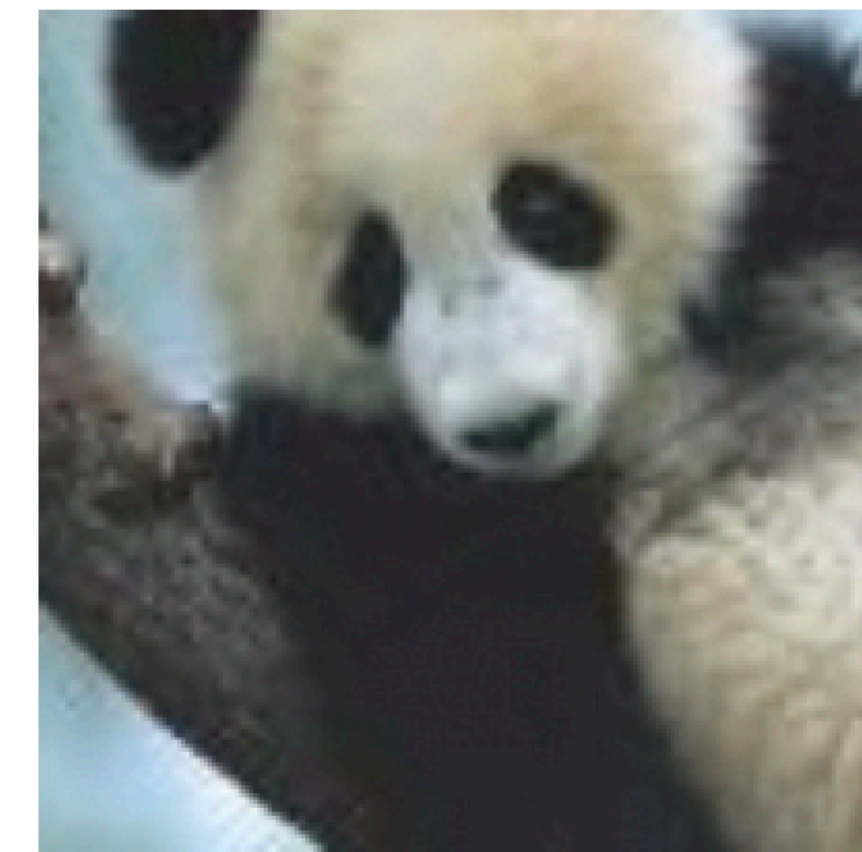


$\text{sign}(\nabla_x J(\theta, x, y))$

“nematode”

8.2% confidence

$=$



$x + \epsilon \text{sign}(\nabla_x J(\theta, x, y))$

“gibbon”

99.3 % confidence

Adversarial attack on GoogLeNet trained on ImageNet

Goodfellow IJ et al 2015

Ethics of ML in Medicine: How good is good enough?

Imagine a scenario where:

- Best group of clinicians gets accuracy of 80%
- Best ML models get accuracy of 85%
- Is ML automatically >> clinicians?
- What if they outperform clinicians in 10% of cases but make otherwise preventable errors in 5% of cases. Is this ethically acceptable?

- Learning algorithms are tools designed to perform a job - map predictors to outcomes - given certain assumptions are met
- Using the wrong tool for a job, or using a tool when its assumptions are not met, or expecting it to perform a function it was not designed to perform is futile

ML for biomedical and clinical applications

- Biomedical research is increasingly **quantitative** and **multimodal**
- **Powerful machine learning** (ML) tools have been widely available for over two decades
- Increases in **computational power** have made deployment of ML tools widely accessible
- Still, the ***majority of biomedical research falls short*** when it comes to using the best available tools

What stops researchers / analysts /... from using latest and greatest data analytic tools?

- **Poor ML education**
- **Tools are difficult to use**
 - Inconsistent input / output among software, poor documentation
 - Bad or missing defaults
 - Need for careful tuning & testing, ++
- **Algorithms are often not interpretable**
 - Good prediction accuracy, at best
 - Relative variable importance
 - Performance - interpretability tradeoff; No mechanistic model of how variables interact to produce / modulate outcome

Introducing ***rtemis*** Machine Learning and visualization:

An ML framework for research, education, and applied data science

1a. **mplot3.*** Static Graphics (base)

x: vector

```
> mplot3.x(x, c('index', 'line', 'histogram', 'density', 'qqline'))
```

x: data frame

```
> mplot3.box(x)
> mplot3.bar(x)
```

x & y: vectors

```
> mplot3.xy(x, y, fit = 'gam')
```

X & Y: matrices

```
> mplot3.heat(cor(X, Y))
```

Classifier estimate and true labels

```
> mplot3.roc(probability, labels)
```

decision tree

```
> mplot3.tree(tree)
```

1b. **dplot3.*** Dynamic Graphics (plotly)

```
> dplot3.x(x, 'density')
> dplot3.xy(x, y, fit = 'ppr')
> dplot3.heat(cor(X, Y))
```

2. **u.*** Unsupervised Learning: Clustering

```
> clustSelect() List available algorithms
```

x: matrix

```
> u.KMEANS(x, k=2)
```

3. **d.*** Unsupervised Learning: Decomposition

```
> decomSelect() List available algorithms
```

x: matrix

```
> d.NMF(x, k=12)
```

4. **s.*** Supervised Learning: Classification, Regression, Survival Analysis

```
> modSelect() List available algorithms
```

*x: matrix; y: vector/Surv** **y is factor: Classification*
y is numeric: Regression
y is Surv object: Survival

```
> s.GBM(x, y)
```

Automatic hyperparameter tuning

```
> s.GBM(x, y, shrinkage=c(.001, .01), interaction.depth=2:5)
```

Single function to preprocess, decompose, train, tune, and test

```
> elevate(x, y, 'gbm')
```

5. **x.*** Cross-Decomposition

```
> xdecomSelect() List available algorithms
```

x: matrix; z: matrix

```
> x.CCA(x, z, k=4)
```

6. **meta**-Modeling

Model Stacking

x: matrix; y: vector

```
> metaMod(x, y, base.mods=c('lgb', 'h2odl'), meta.mod='gam')
```

Modality Stacking

x: list of matrices; y: vector

```
> metaFeat(x, y, base.mods='xgblin', meta.mod='gam')
```

Group-Weighted Stacking

x: matrix; y: vector

```
> metaGroup(x, y, base.mods='xgblin', group=group, meta.mod='mdb')
```

R6 Class system

One class for each model family:

rtClust

rtDecom

rtMod, rtModCV, rtModBag

rtXDecom

- Objects contain both attributes and methods
- Support S3 generics: print, plot, summary, predict, etc

rtemis is for

Research

New machine learning algorithm development,
with a focus on:

- Interpretable models
- Meta models
- Novel representational and multimodal learning approaches

Education

Go from theory to application painlessly
Learn ML concepts and practices using focused vignettes

Applied Data Science

Apply advanced ML pipelines
on your data correctly and effortlessly

Example: Tune, train, and test a predictive model in *rtemis*

R code

```
data <- read.csv("data.csv")
```

Description

Read data into R

```
mod <- elevate(data)
```

Run **10-fold cross-validated random forest**
Classification or Regression


```
mod <- elevate(data, mod = 'gbm')
```

Run 10-fold cross-validated **gradient boosting**
with early stopping

```
mod <- elevate(data, mod = 'gbm',  
               interaction.depth = c(2, 5, 10),  
               shrinkage = c(.01, .005))
```

Run 10-fold cross-validated gradient boosting
with early stopping and automatic hyperparameter tuning
by grid search

$= (3 * 2 * 5 + 1) * 10 = 310$ models


```
mod <- elevate(data,  
  .preprocess = rtset.preprocess(impute = TRUE),  
  mod = 'gbm',  
  interaction.depth = c(2, 5, 10),  
  shrinkage = c(.01, .005))
```

Impute missing data and
run 10-fold cross-validated gradient boosting
with early stopping and automatic hyperparameter tuning
by grid search

```
mod <- elevate(data,  
  .preprocess = rtset.preprocess(impute = TRUE),  
  .decompose = rtset.decompose('ICA', 20),  
  mod = 'gbm',  
  interaction.depth = c(2, 5, 10),  
  shrinkage = c(.01, .005))
```

Impute missing data,
decompose to 20 ICA dimensions,
run 10-fold cross-validated gradient boosting
with early stopping and automatic hyperparameter tuning
by grid search

```
mod <- elevate(data,  
  .preprocess = rtset.preprocess(impute = TRUE),  
  .decompose = rtset.decompose('ICA', 20),  
  .resample = rtset.resampler('strat.sub', 25),  
  mod = 'gbm',  
  interaction.depth = c(2, 5, 10),  
  shrinkage = c(.01, .005))
```

Impute missing data,
decompose to 20 ICA dimensions,
run **25 stratified subsamples** using gradient boosting
with early stopping and automatic hyperparameter tuning
by grid search


```
mod <- elevate(data,  
  .preprocess = rtset.preprocess(impute = TRUE),  
  .decompose = rtset.decompose('ICA', 20),  
  .resample = rtset.resampler('strat.sub', 25),  
  mod = 'gbm',  
  interaction.depth = c(2, 5, 10),  
  shrinkage = c(.01, .005),  
  outdir = '/Drive/Project/Results')
```

Impute missing data,
decompose to 20 ICA dimensions,
run 25 stratified subsamples using gradient boosting
with early stopping and automatic hyperparameter tuning
by grid search,
and **save all to output directory** (log.txt, .rds model, PDF plots)

```
> sonar.rf <- elevate(Sonar)
[2019-06-05 14:09:04 elevate] Hello, egenn

[[ Classification Input Summary ]]
  Training features: 208 x 60
  Training outcome: 208 x 1

[2019-06-05 14:09:04 resLearn] Training Random Forest (ranger) on 10 stratified subsamples...
|+++++| 100% elapsed = 03s

[[ elevate RANGER ]]
  N repeats = 1
  N resamples = 10
  Resampler = strat.sub
  Mean Balanced Accuracy of 10 test sets in each repeat = 0.84

[2019-06-05 14:09:07 elevate] Run completed in 0.06 minutes (Real: 3.30; User: 4.82; System: 0.20)
```

R6 Objects hold all relevant attributes and functions

```
sonar.rf$
```

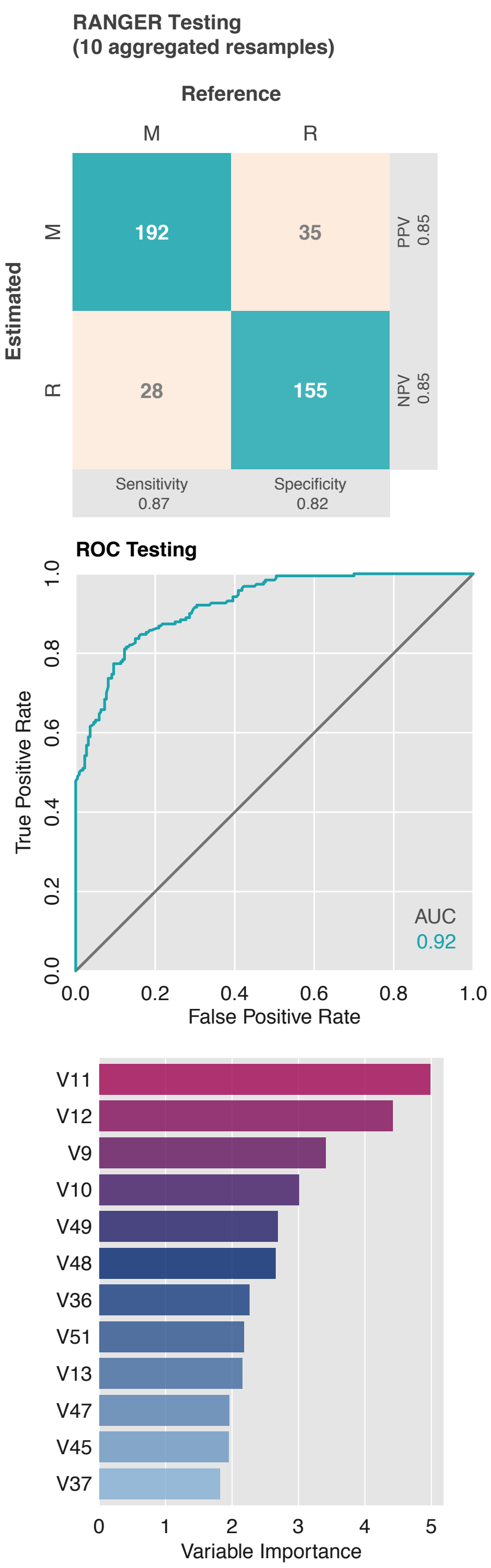
parameters	
plot	{sonar.rf}
plotFitted	{sonar.rf}
plotPredicted	{sonar.rf}
plotROC	{sonar.rf}
plotROCFitted	{sonar.rf}
plotROCpredicted	{sonar.rf}

```
> class(sonar.rf)
[1] "rtModCVclass" "rtModCV"      "R6"
```

```
sonar.rf$plot( )
```

```
sonar.rf$plotROC( )
```

```
sonar.rf$plotVarImp( )
```



Go to file/function

Addins

Palette.R ×rtCol_DEV.R ×mplot3.bar.R ×rtemisKey.R ×rtMod.R ×referenceSemantics.R ×unsupervisedLearning_tests.R ×

Source on SaveRunSource

24 res <- resample(y)

25 dat.train <- data.frame(x[res\$Resample01,], y = y[res\$Resample01])

26 dat.test <- data.frame(x[-res\$Resample01,], y = y[-res\$Resample01])

27 dat.train[20, 4] <- dat.train[30, 8] <- dat.train[60, 20] <- NA

28

29 dat.ridge <- elevate(dat.train, dat.test,

30 .preprocess = rtset.preprocess(impute = TRUE),

31 mod = "GLMNET", lambda = .1, alpha = 0)

32 dat.ridge\$plot()

31:61# elevate RegressionR Script

ConsoleTerminalJobs

/Applications/rtemis/

> dat.ridge <- elevate(dat.train, dat.test,

+ .preprocess = rtset.preprocess(impute = TRUE),

+ mod = "GLMNET", lambda = .1, alpha = 0)

[2019-06-05 14:16:43 elevate] Hello, egenn

[[Regression Input Summary]]

Training features: 301 x 40

Training outcome: 301 x 1

[2019-06-05 14:16:43 preprocess.default] Imputing missing values using missForest...

missForest iteration 1 in progress...done!

missForest iteration 2 in progress...done!

missForest iteration 3 in progress...done!

[2019-06-05 14:16:54 preprocess.default] Done

[2019-06-05 14:16:54 resLearn] Training Elastic Net on 10 stratified subsamples...

|+++++| 100% elapsed = 01s

[[elevate GLMNET]]

N repeats = 1

N resamples = 10

Resampler = strat.sub

Mean MSE of 10 resamples in each repeat = 1.30

Mean MSE reduction in each repeat = 96.47%

[2019-06-05 14:16:55 elevate] Run completed in 0.21 minutes (Real: 12.45; User: 8.98; System: 0.50)

> dat.ridge\$plot()

EnvironmentHistoryConnectionsBuildGit

Import Dataset

Global Environment

Data

dat.ridgeEnvironment

dat.test99 obs. of 41 variables

dat.train301 obs. of 41 variables

resList of 10

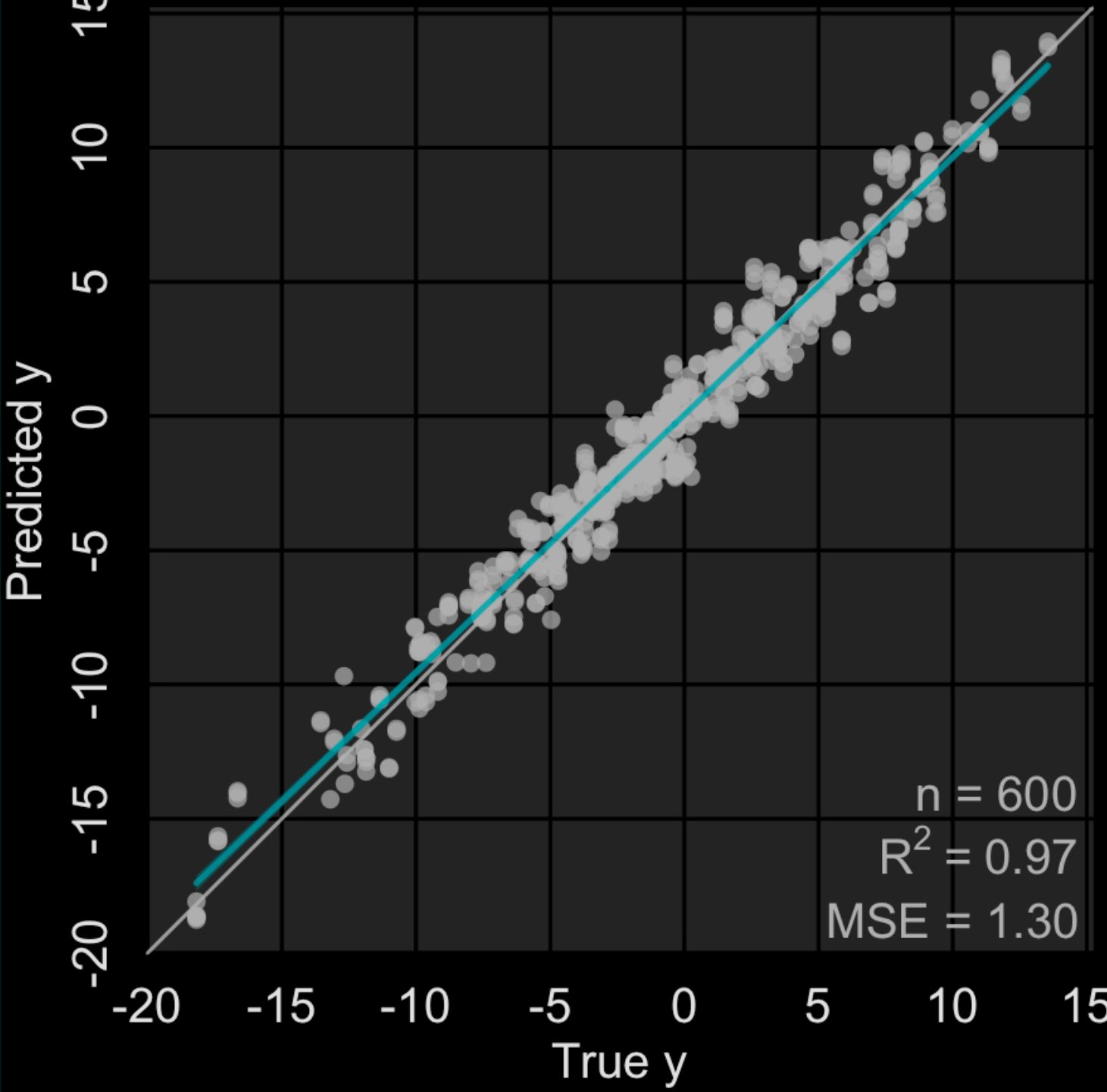
num [1:400, 1:40] 0.739 0.515 -1...

FilesPlotsPackagesHelpViewer

ZoomExport

GLMNET Testing

(10 aggregated resamples)



n = 600

R² = 0.97

MSE = 1.30

```
> dat.ridge$describe()
```

```
Regression was performed using Elastic Net. Data was preprocessed by imputing missing values using missForest. Model generalizability was assessed using 10 stratified subsamples. The mean R-squared across all resamples was 0.96.
```

ML in R: **caret** vs **rtemis**

caret

```
set.seed(998)
inTraining <- createDataPartition(Sonar$Class,
                                   p = .75,
                                   list = FALSE)

training <- Sonar[ inTraining,]
testing  <- Sonar[-inTraining,]

fitControl <- trainControl(method = "repeatedcv",
                           number = 10,
                           repeats = 10)

gbmGrid <- expand.grid(interaction.depth = c(1, 5, 9),
                      n.trees = (1:30)*50, !!
                      shrinkage = 0.1,
                      n.minobsinnode = 20)

gbmFit <- train(Class ~ ., data = training,
               method = "gbm",
               trControl = fitControl,
               tuneGrid = gbmGrid)

predicted <- predict(gbmFit, testing)      +++
```

rtemis

```
mod.gbm <- elevate(Sonar, 'gbm',
                   shrinkage = .1,
                   minobsinnode = 20,
                   interaction.depth = c(1, 5, 9),
                   max.trees = 1500,
                   resampler = "kfold",
                   n.resamples = 10,
                   n.repeats = 10, seed = 998)
```

No nested resampling: manually loop over this, extract error, train full model, etc.; No early stopping

<https://topepo.github.io/caret/model-training-and-tuning.html>

rtemis Design Principles

Efficient

Minimize user time: flexible, high-level functions

Minimize machine time: parallel and distributed computation

Minimize user error by minimizing user code input

Accessible

User friendly and intuitive;

minimize need to consult the manual

Transparent analysis pipelines, informative messaging,
error reporting, and logging

Modular

Single platform for supervised & unsupervised
learning, visualization with cross-integration;
easy to build new pipelines and procedures

rtemis Implementation

Homogenized I/O	For each type of model (supervised learning, clustering, decomposition, etc)
R6 objects	Reference semantics; hold attributes and functions
R6 + S3 methods	R generics for all objects (summary, print, plot, predict, etc.)
Parallel execution	On by default: forking (default) or PSOCK for Linux, macOS; PSOCK for Windows

Each type of model gets its own R6 class

- **rtMod & rtModClass**: Supervised Learning
- **rtModBag**: Bagged learners
- **rtModCV & rtModCVclass**: Cross-validated learners
- **rtDecom**: Decomposition
- **rtClust**: Clustering
- **rtXDecom**: Cross-decomposition
- **rtMeta**: Meta models

R6 Objects contain **attributes** and **methods**

```
mod <- s.PPR(x.train, y.train, x.test, y.test)
mod$
```

◆ plotFitted	{mod}
◆ plotPredicted	{mod}
◆ predicted	
◆ print	{mod}
◆ question	
◆ se.fit	
◆ se.prediction	

```
> methods(class = 'rtMod')
```

```
[1] coef      fitted    plot      predict   print     residuals summary
see '?methods' for accessing help and source code
```


rtemis Algorithms: Supervised Learning

Alias	Description	Class	Reg	Surv
ADABOOST	Adaptive Boosting	T	F	F
ADDTBOOST	Boosting of Additive Trees	T	T	F
ADDTREE	Additive Tree	T	T	F
BART	Bayesian Additive Regression Trees	T	T	F
BAYESGLM	Bayesian GLM	T	T	F
BRUTO	BRUTO Additive Model	F	T	F
CART	Classification and Regression Trees	T	T	T
CFOREST	Conditional Random Forest	T	T	T
CTREE	Conditional Inference Trees	T	T	T
C50	C5.0 Decision Tree	T	F	F
DN	Deep Learning using deepnet	T	T	F
ET	Extra Trees	T	T	F
EVTREE	Evolutionary Learning of Globally Optimal Trees	T	T	F
GAM	Generalized Additive Model	T	T	F
GBM	Gradient Boosting Machine	T	T	T
GLM	Generalized Linear Model	T	T	F
GLMNET	Elastic Net	T	T	T
GLS	Generalized Least Squares	F	T	F
H2ODL	H2O Deep Learning	T	T	F
H2OGBM	H2O Gradient Boosting Machine	T	T	F
H2ORF	H2O Random Forest	T	T	F
KNN	k-Nearest Neighbor	T	T	F
LDA	Linear Discriminant Analysis	T	F	F
LGB	Light Gradient Boosting Machine	T	T	F

Alias	Description	Class	Reg	Surv
LM	Ordinary Least Squares Regression	F	T	F
LOESS	Local Polynomial Regression	F	T	F
LOGISTIC	Logistic Regression	T	F	F
MARS	Multivariate Adaptive Regression Splines	T	T	F
MLGBM	Spark MLlib Gradient Boosting	T	T	F
MLGLM	Spark MLlib GLM	T	T	F
MLRF	Spark MLlib Random Forest	T	T	F
MULTINOM	Multinomial Logistic Regression	T	F	F
MXN	Deep Learning using MXNET	T	T	F
NBAYES	Naive Bayes	T	F	F
NW	Nadaraya-Watson Kernel Regression	F	T	F
POLY	Polynomial Regression	F	T	F
POLYMARS	Multivariate Adaptive Polynomial Spline Regression	T	T	F
PPR	Projection Pursuit Regression	F	T	F
PPTREE	Projection Pursuit Trees	T	F	F
QRNN	Quantile Neural Network Regression	F	T	F
RF	Random Forest	T	T	F
RFSRC	Random Forest (Survival, Regression, Classification)	T	T	T
RULEFEAT	Rulefit (GBM + Lasso)	T	T	F
RLM	Robust Linear Model	F	T	F
SGD	Stochastic Gradient Descent	F	T	F
SIG	One Sigmoid Unit Regression	F	T	F
SPLS	Sparse Partial Least Squares	F	T	F
SVM	Support Vector Machine	T	T	F
TLS	Total Least Squares	F	T	F
XGB	Extreme Gradient Boosting	T	T	F
XGBLIN	Extreme Gradient Boosting of Linear Models	F	T	F

Unsupervised Learning

rtemis Algorithms: Clustering

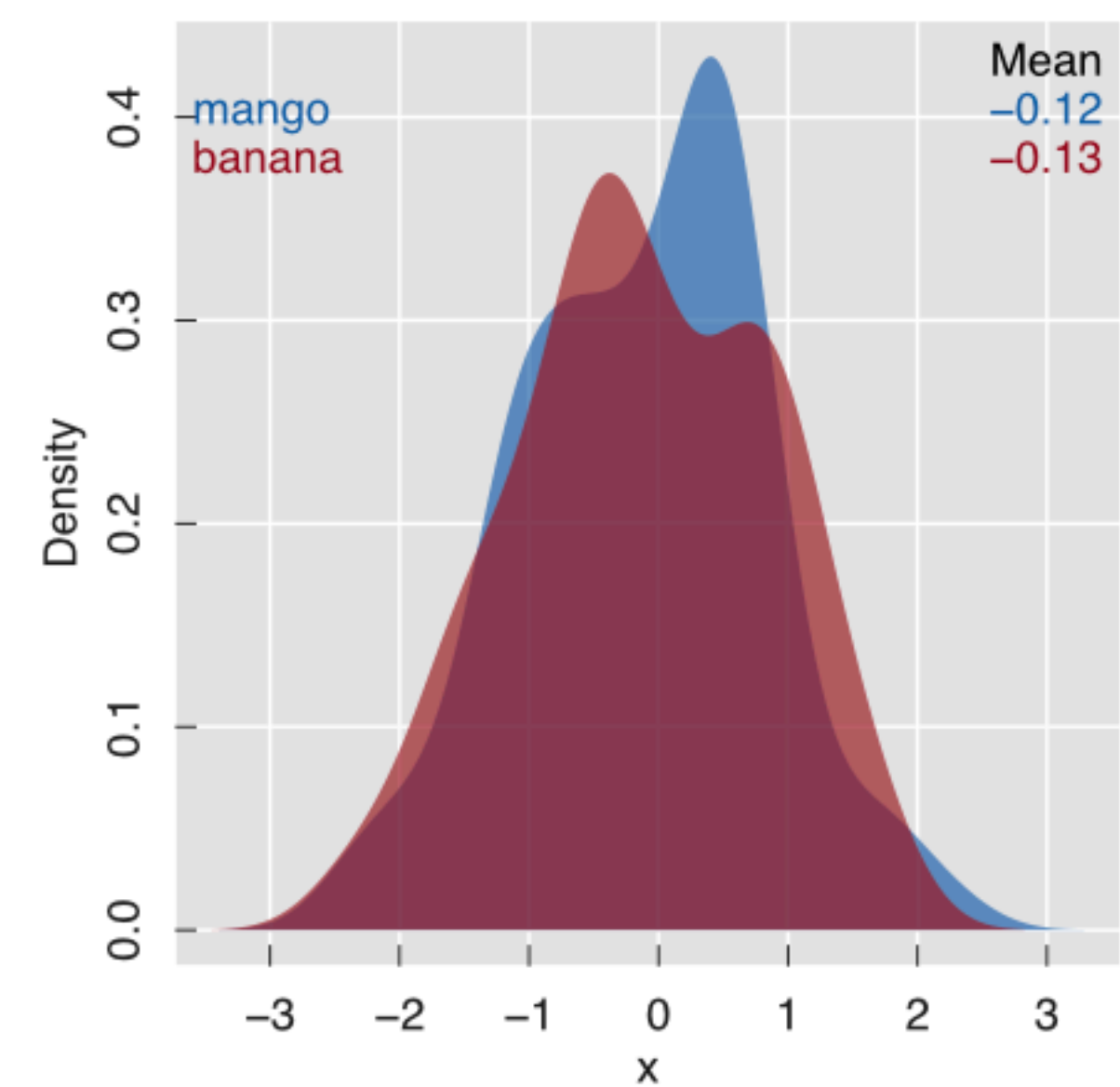
Alias	Description
CMEANS	Fuzzy C-means Clustering
EMC	Expectation-maximization Clustering
HARDCL	Hard Competitive Learning
HOPACH	Hierarchical Ordered Partitioning And Collapsing Hybrid
H2OKMEANS	H2O K-Means Clustering
KMEANS	K-Means Clustering
NGAS	Neural Gas Clustering
PAM	Partitioning Around Medoids
PAMK	Partitioning Around Medoids with k estimation
SPEC	Spectral Clustering

rtemis Algorithms: **Decomposition**

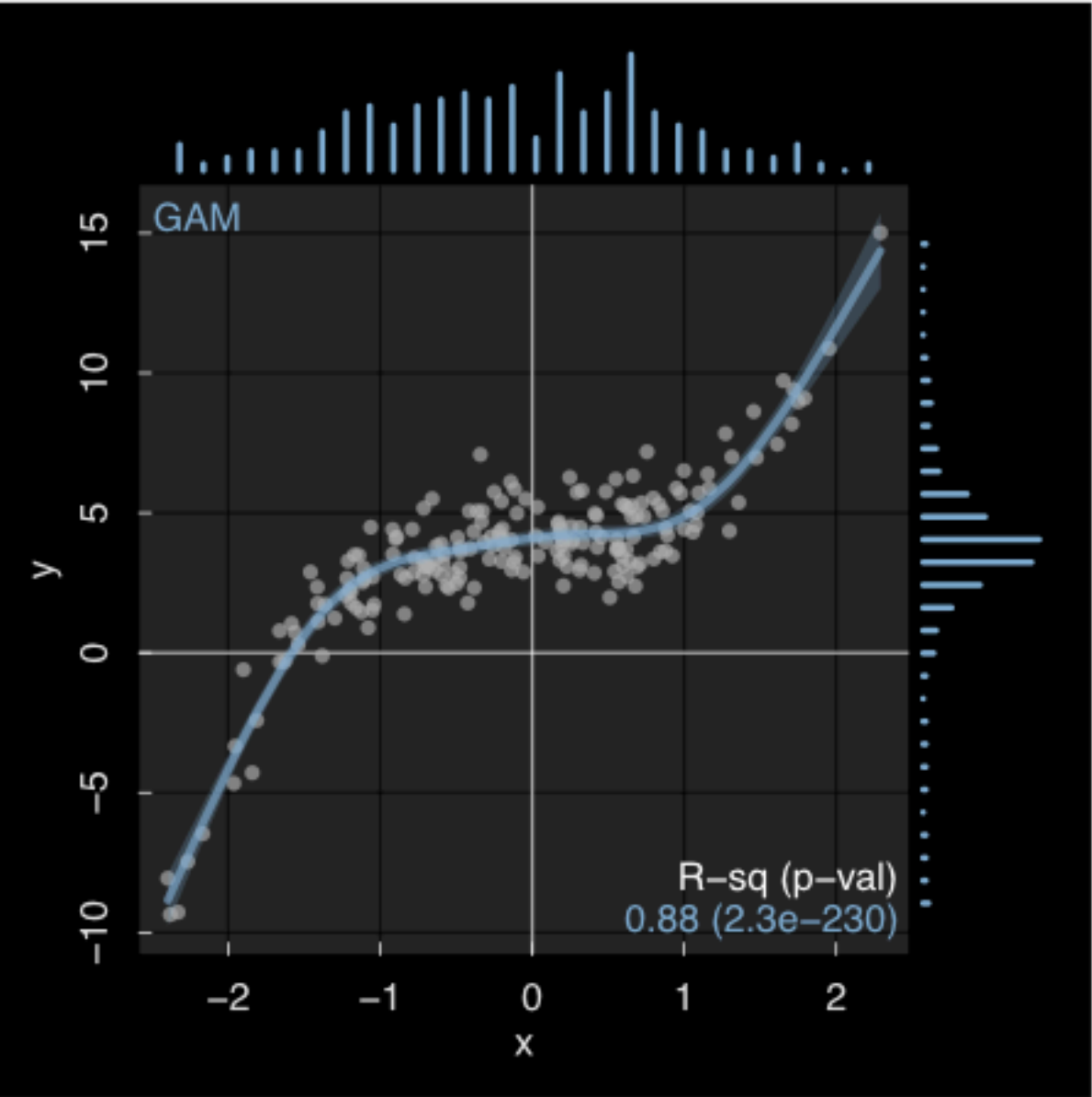
Alias	Description
CUR	CUR Matrix Approximation
H2OAE	H2O Autoencoder
H2OGLRM	H2O Generalized Low-Rank Model
ICA	Independent Component Analysis
ISOMAP	ISOMAP
KPCA	Kernel Principal Component Analysis
LLE	Locally Linear Embedding
MDS	Multidimensional Scaling
NMF	Non-negative Matrix Factorization
PCA	Principal Component Analysis
SPCA	Sparse Principal Component Analysis
SVD	Singular Value Decomposition
TSNE	t-distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection

Visualization

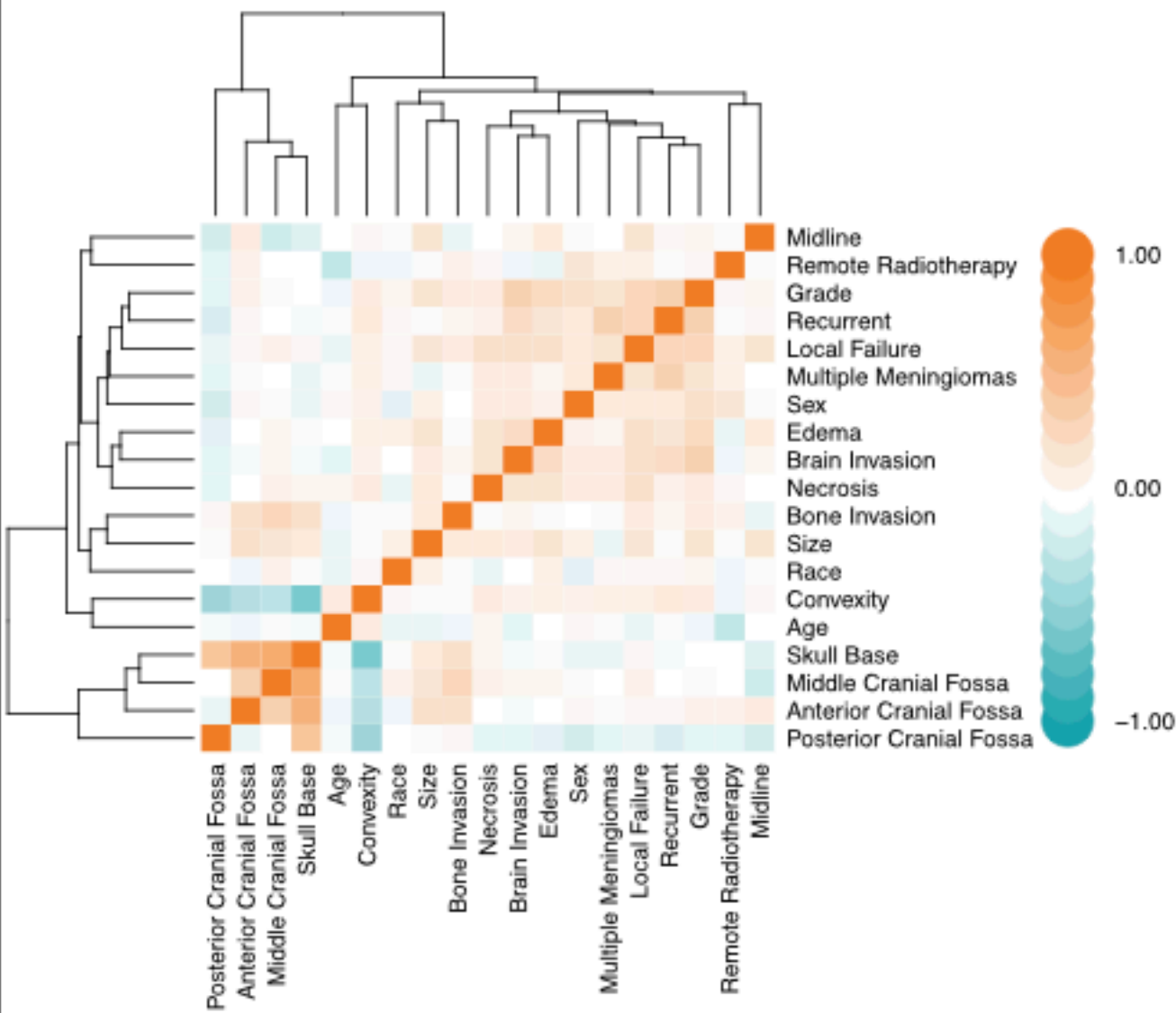
mplot3.x



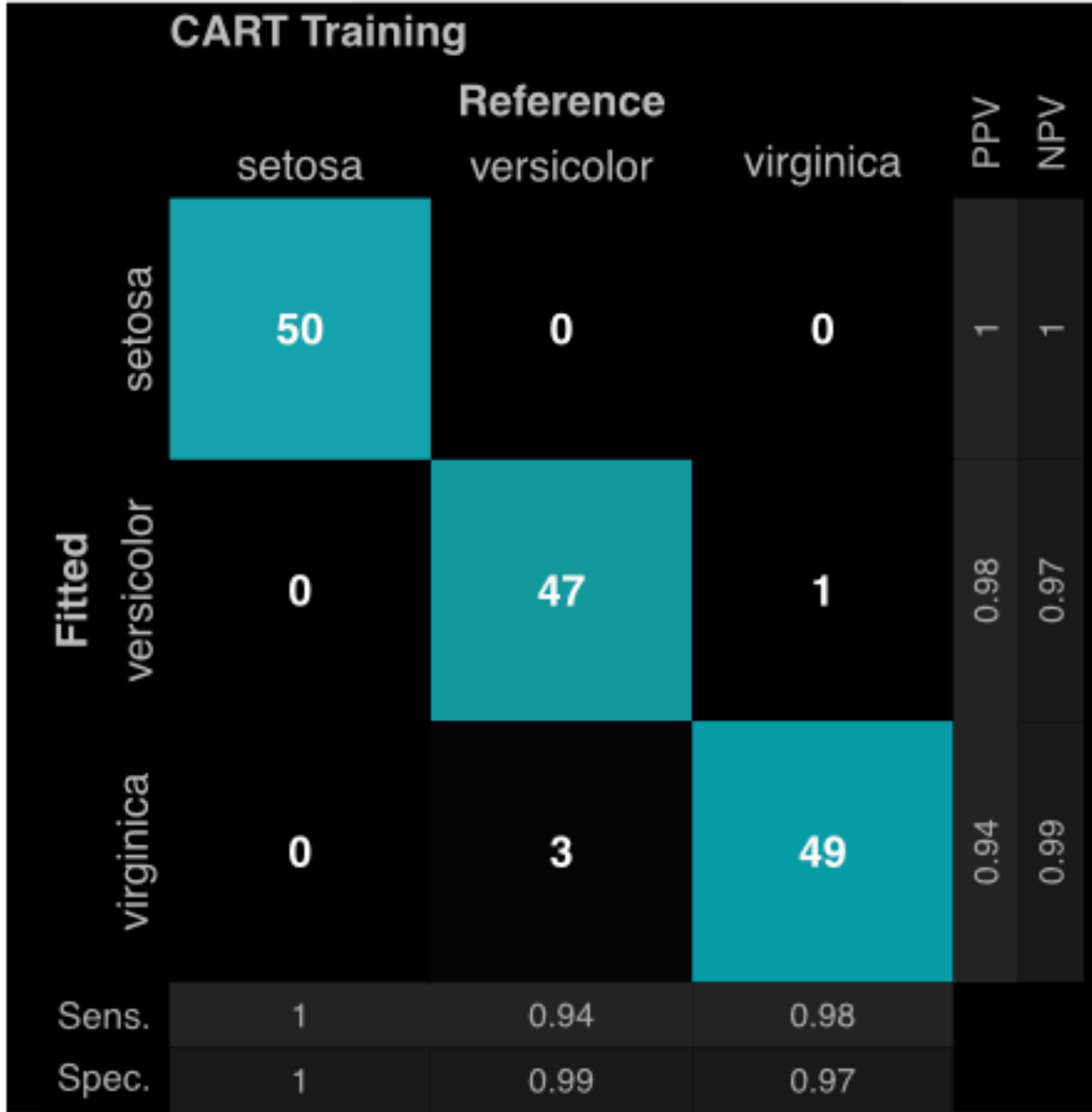
mplot3.xym



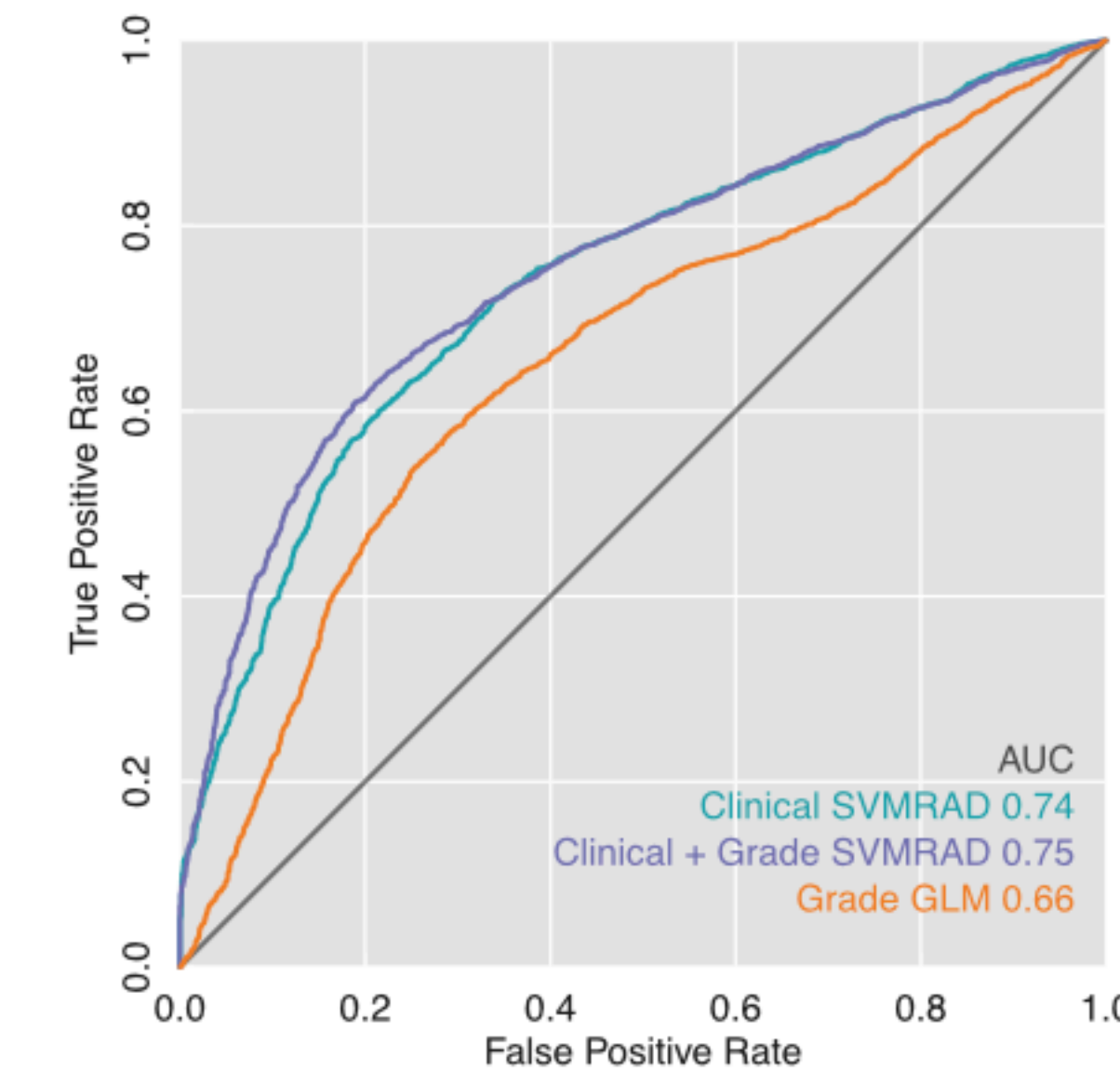
mplot3.heat



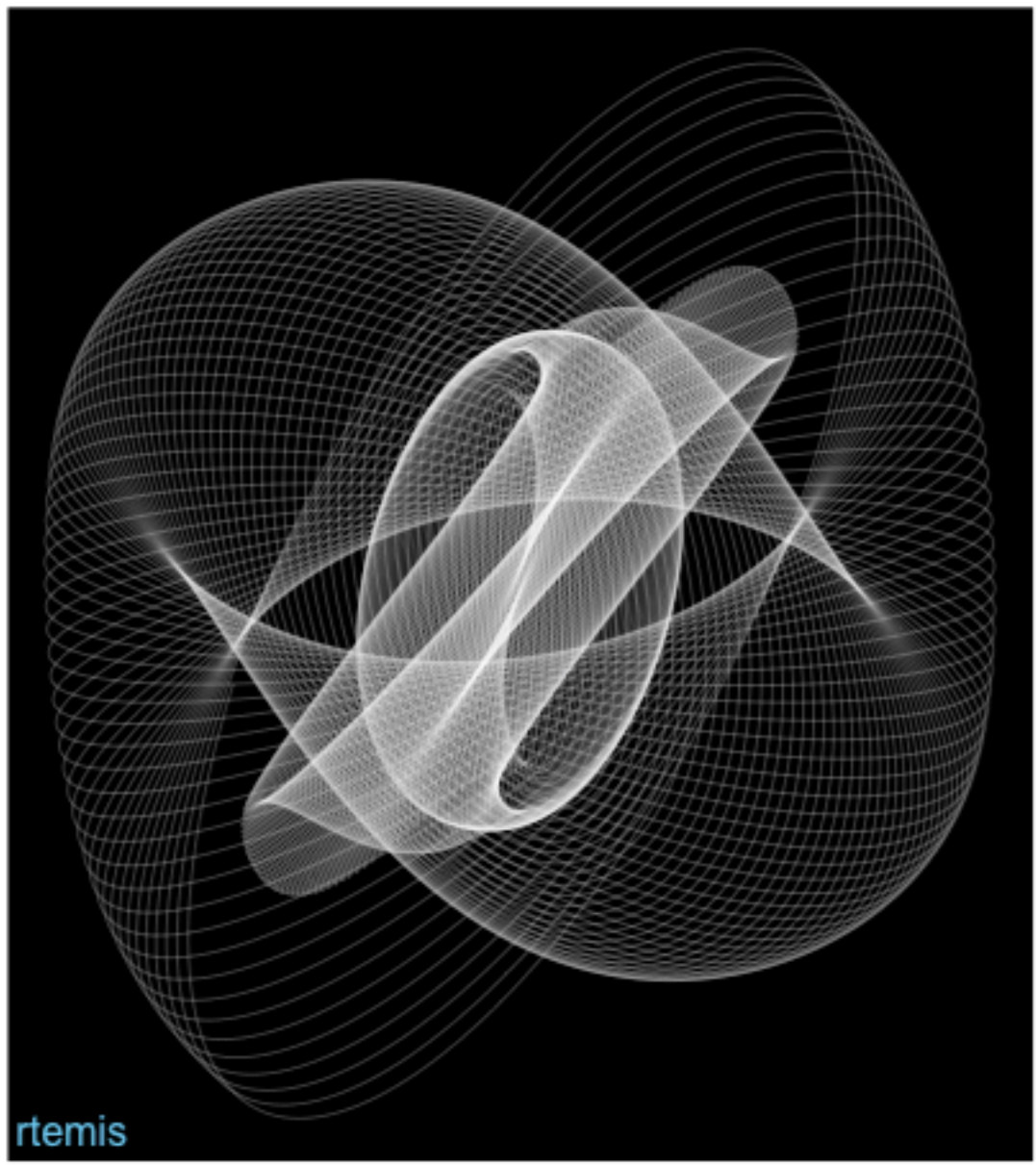
mplot3.conf



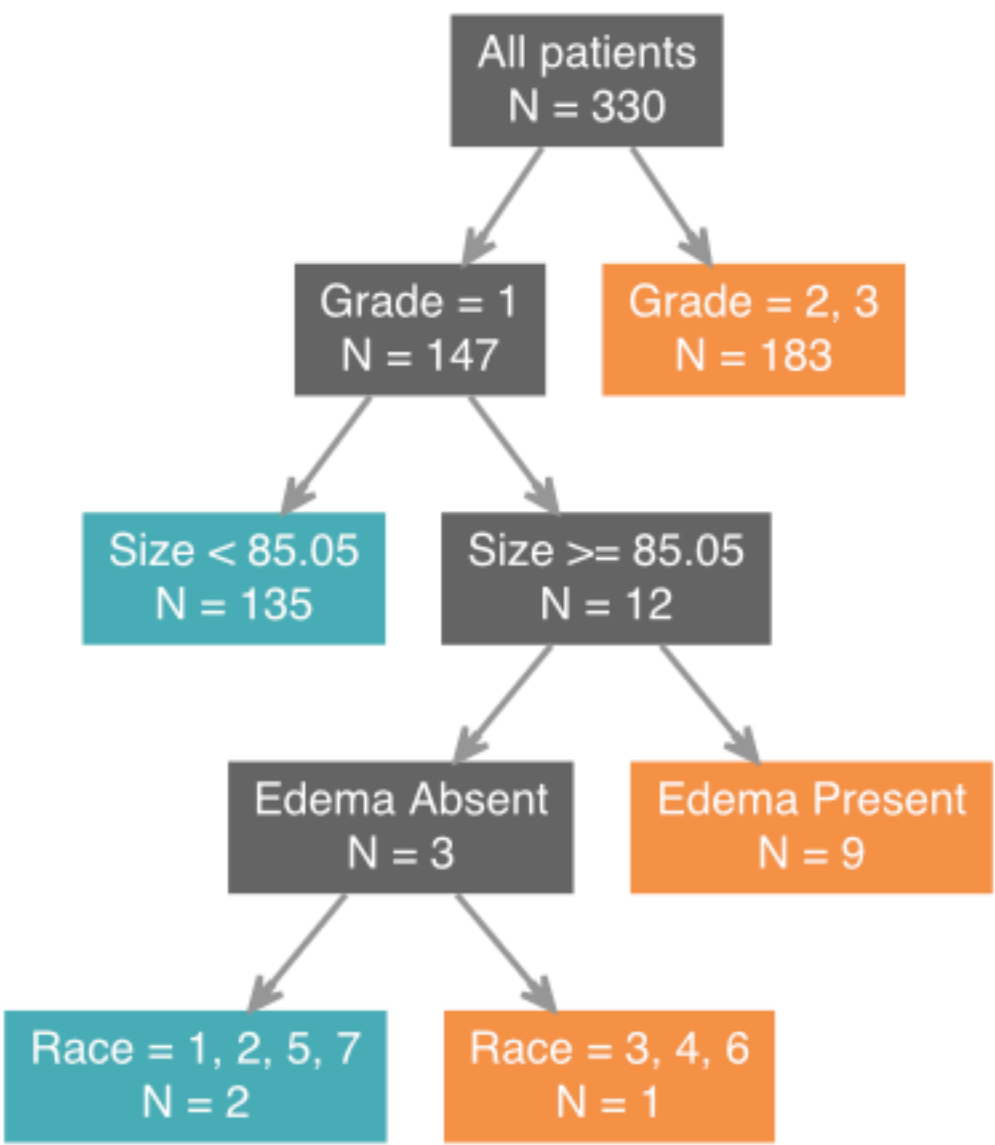
mplot3.roc



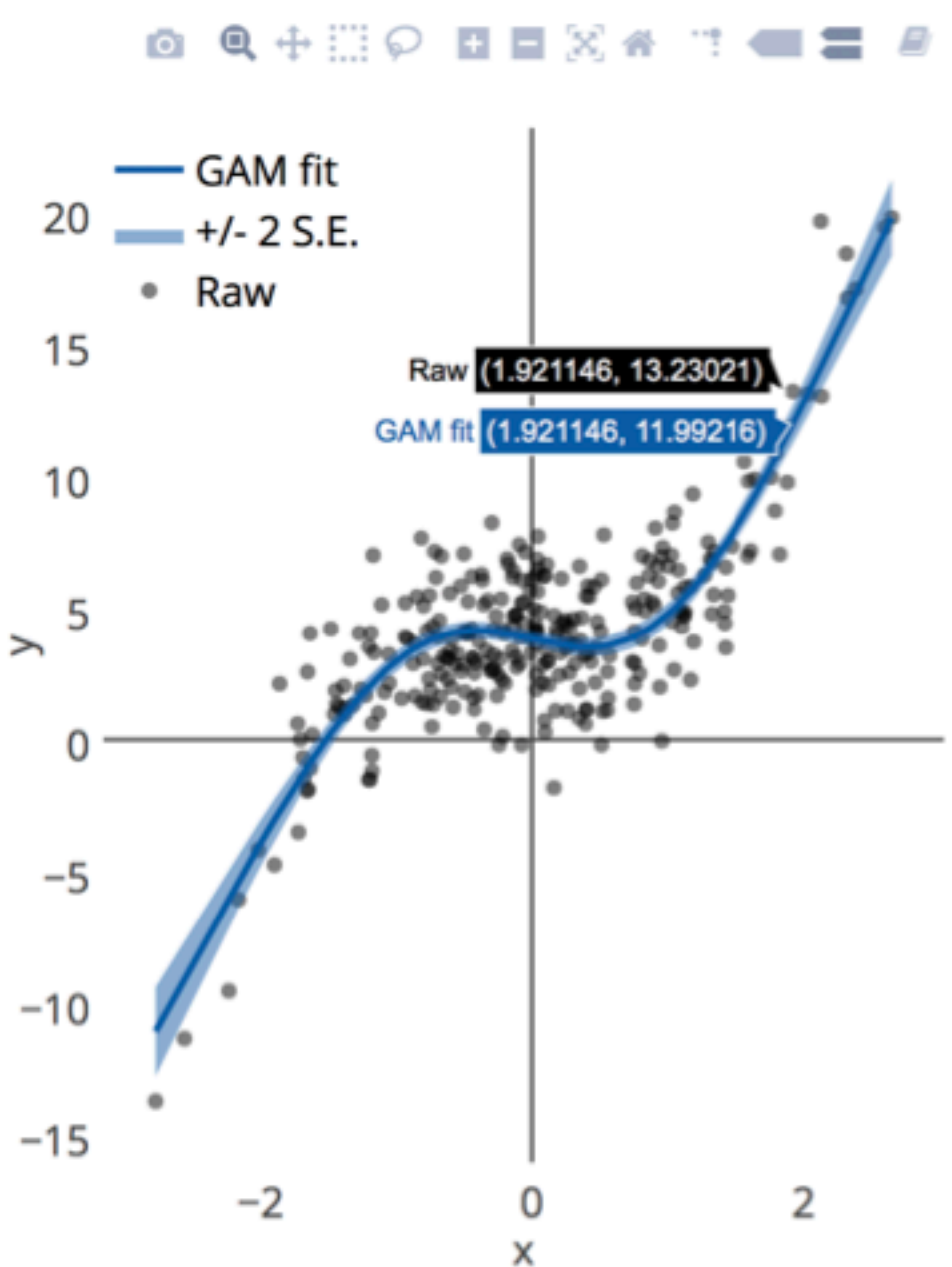
mplot3.harmonograph

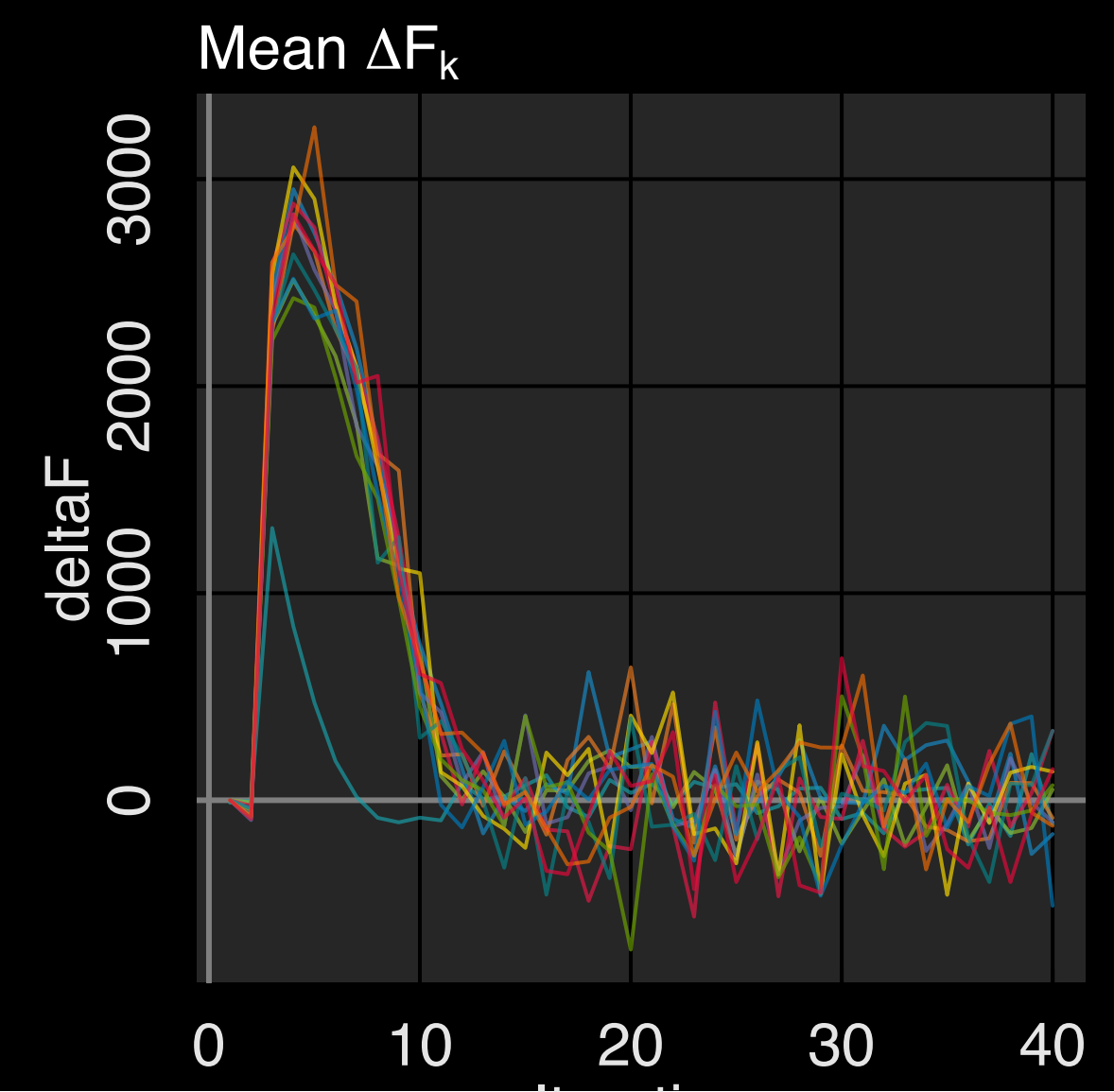
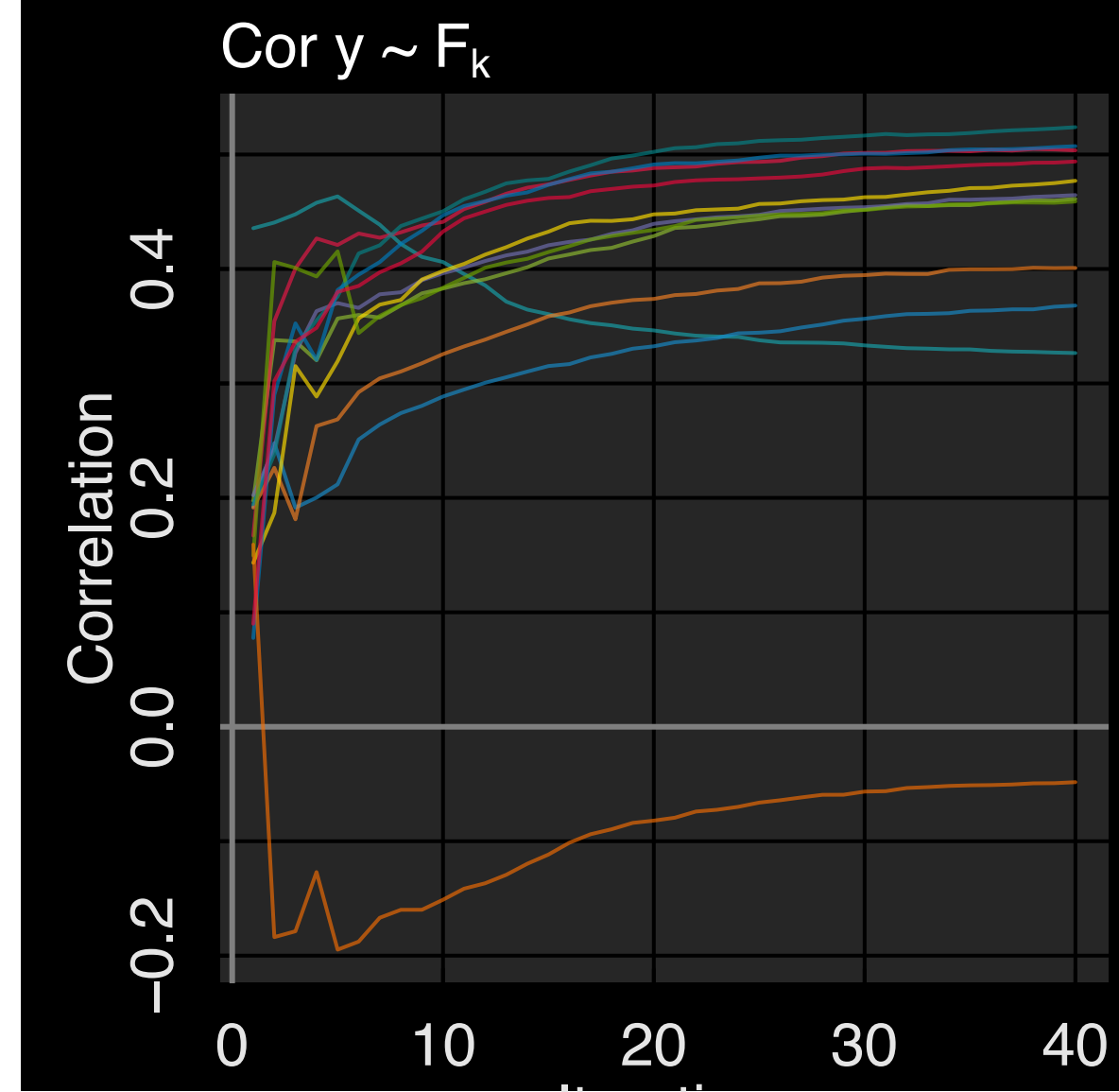
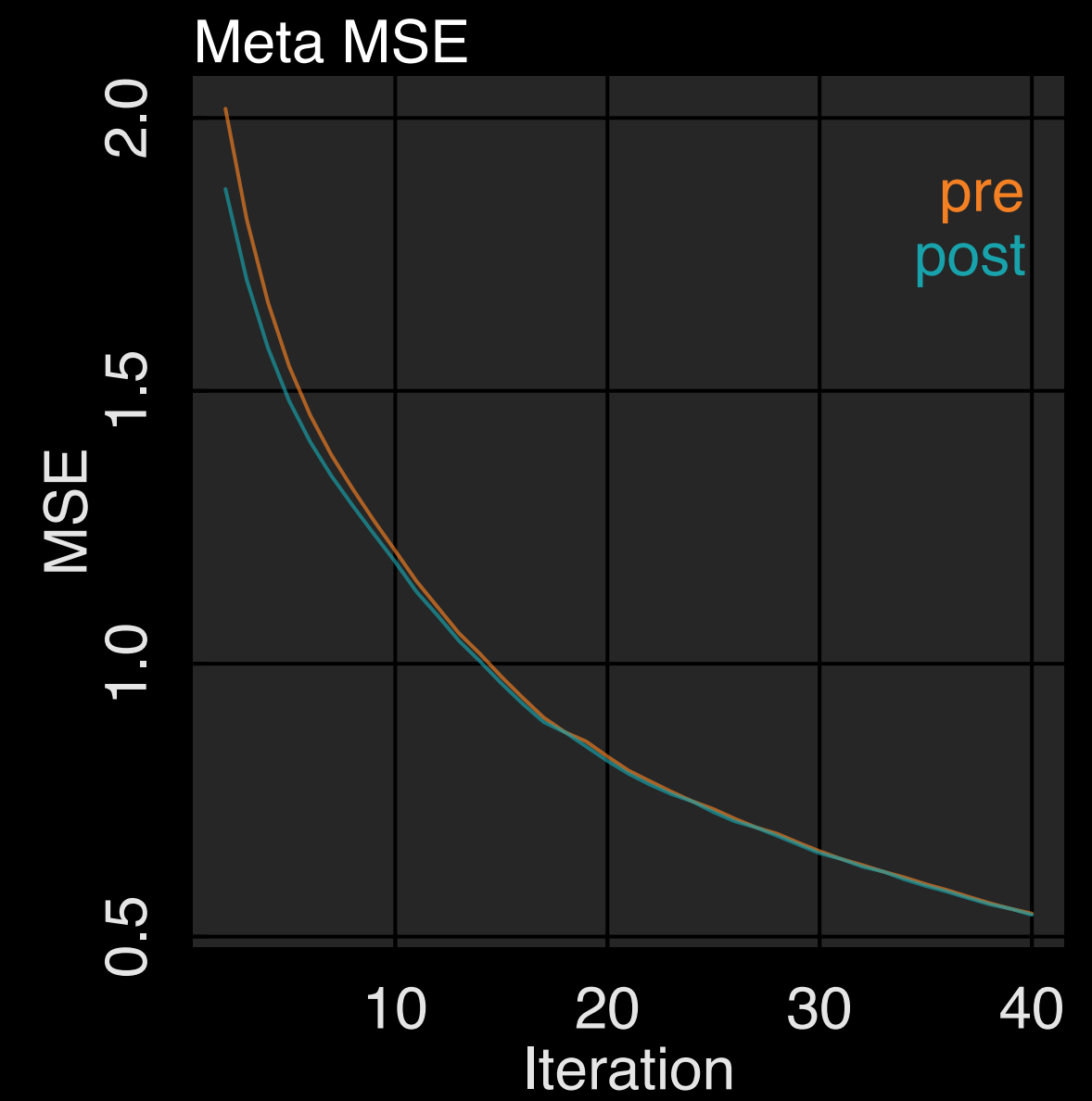
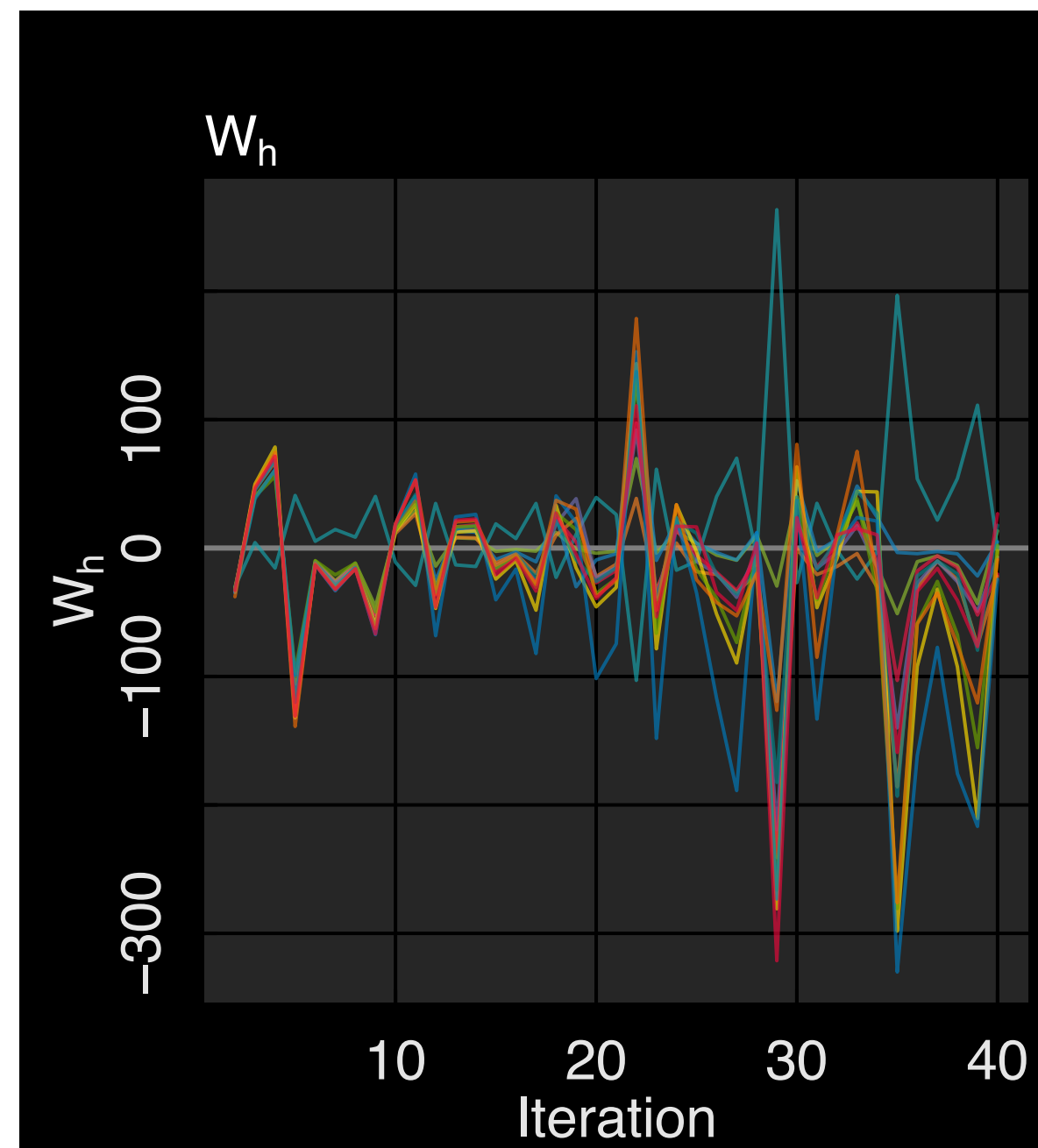
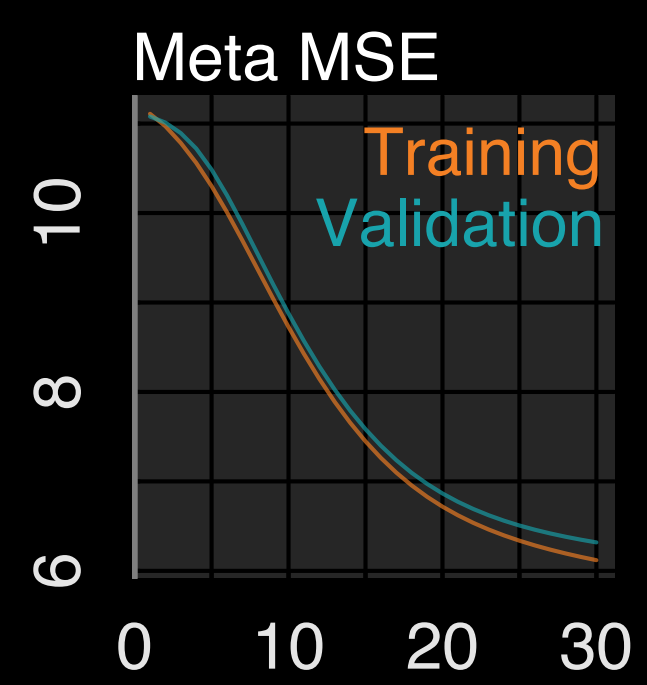
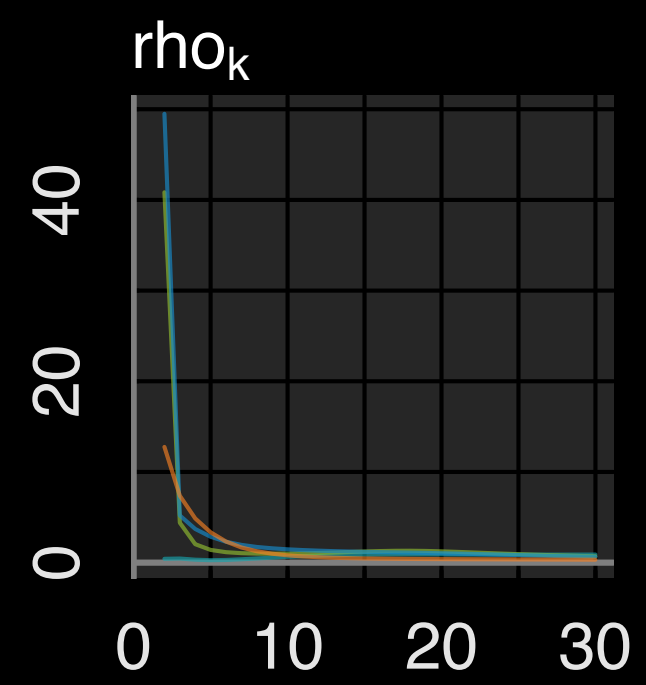
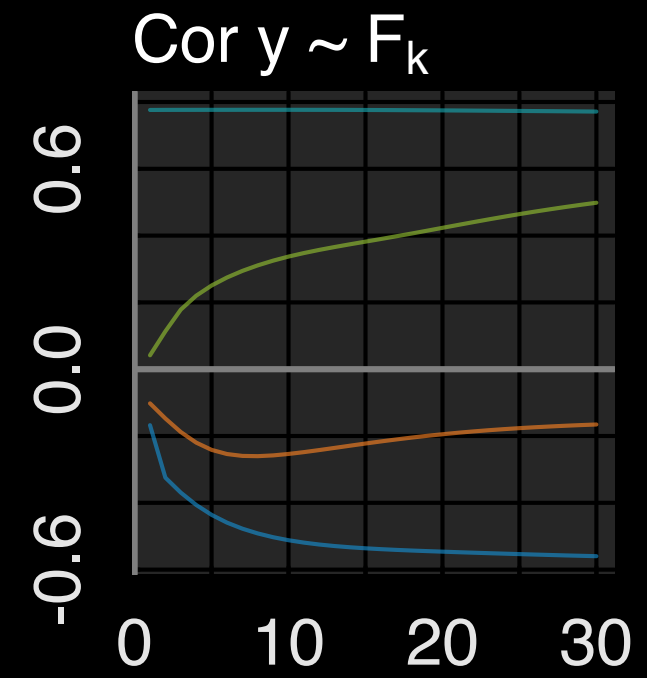
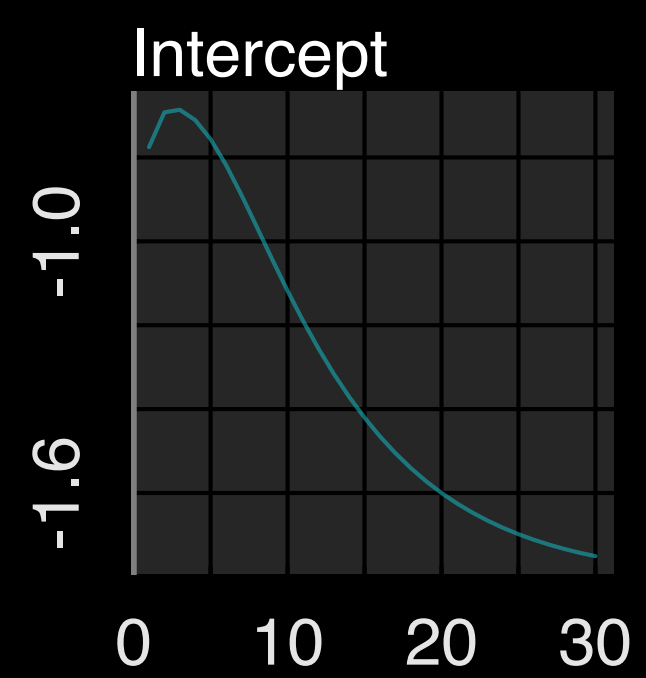
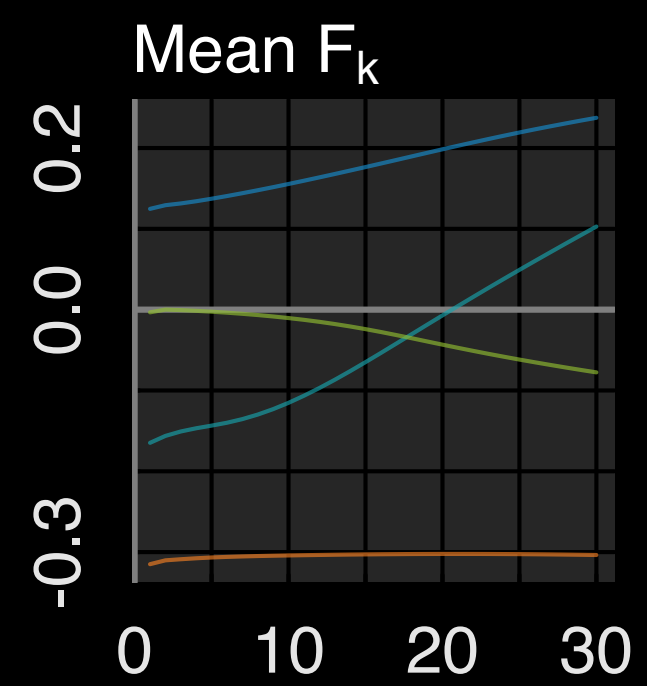
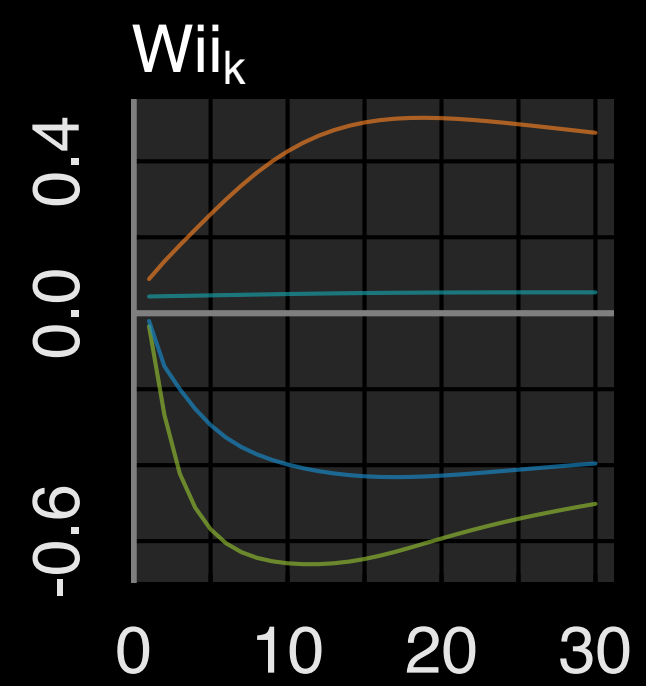
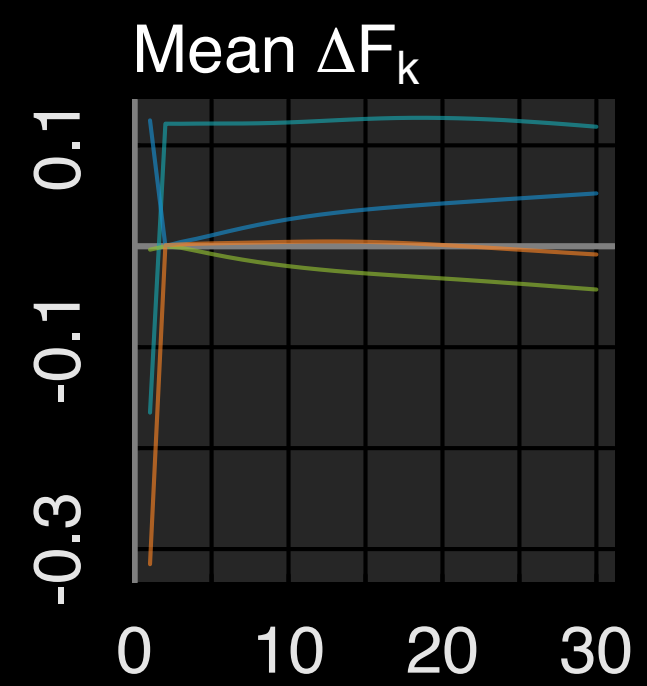
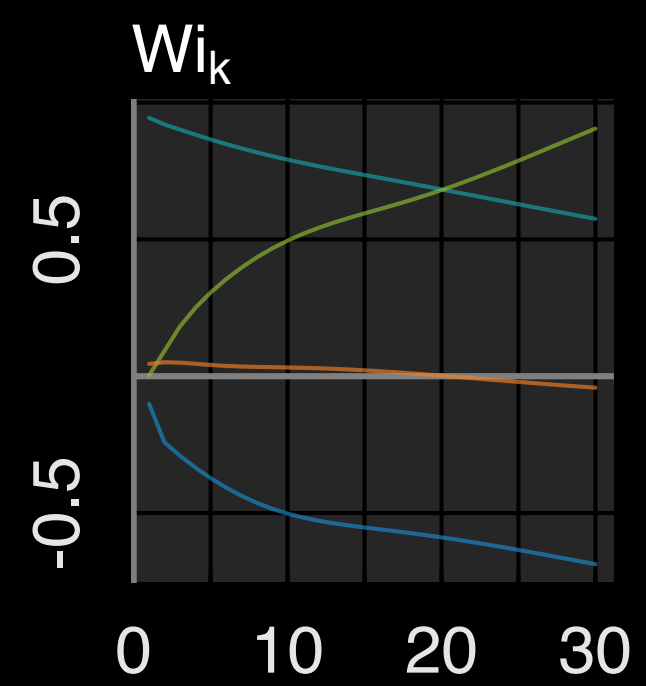


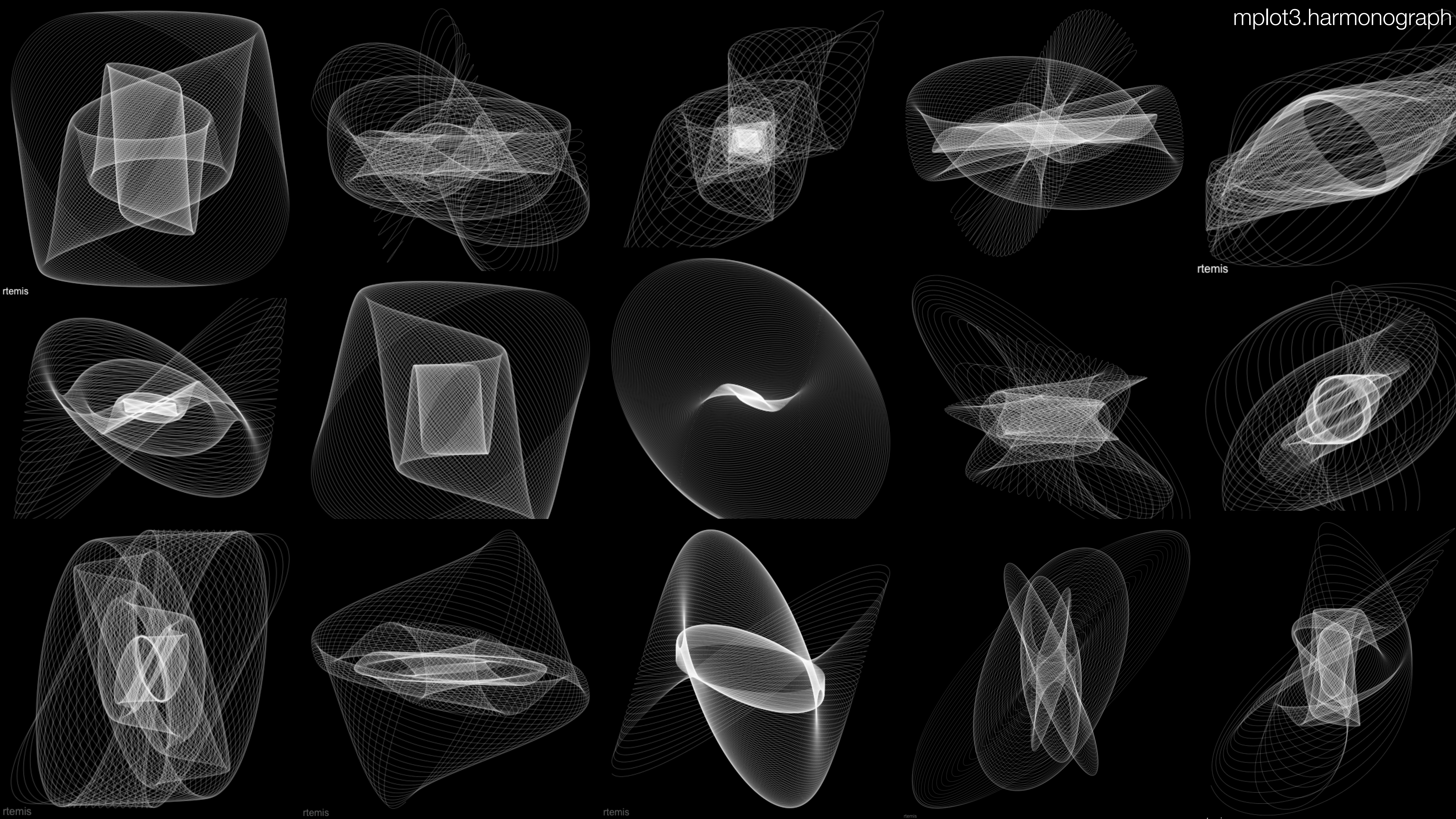
mplot3.cart



dplot3







Static graphics

“base” graphics

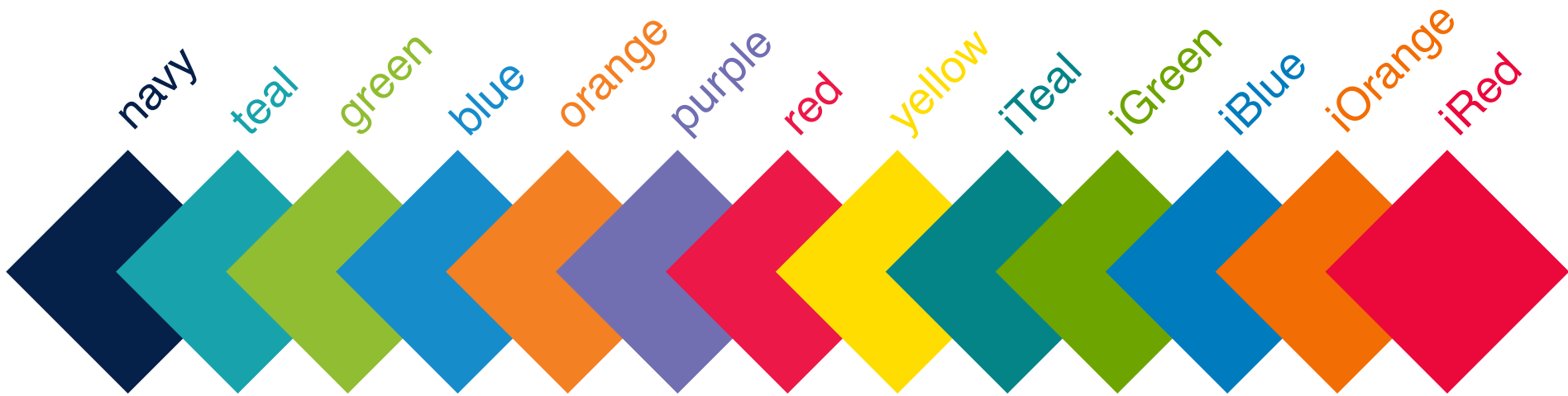
- Very fast (vs. ggplot2)
- Easy for user to extend

Dynamic graphics

plotly

- High quality open source APIs
- Active development

UCSF



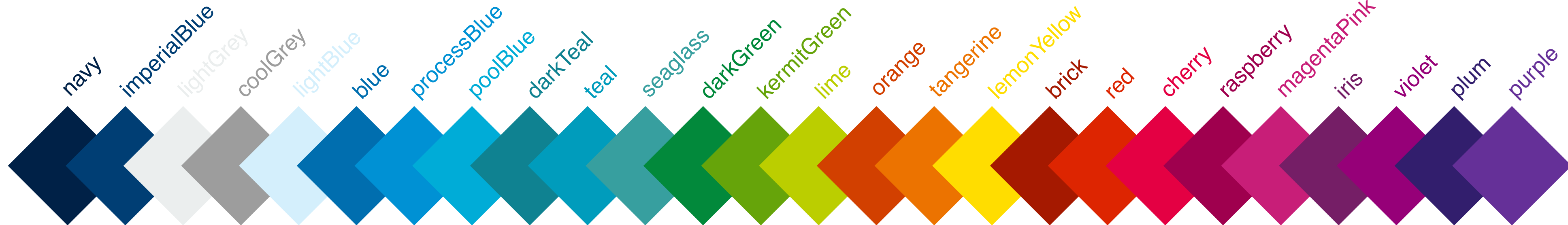
Berkeley



Stanford



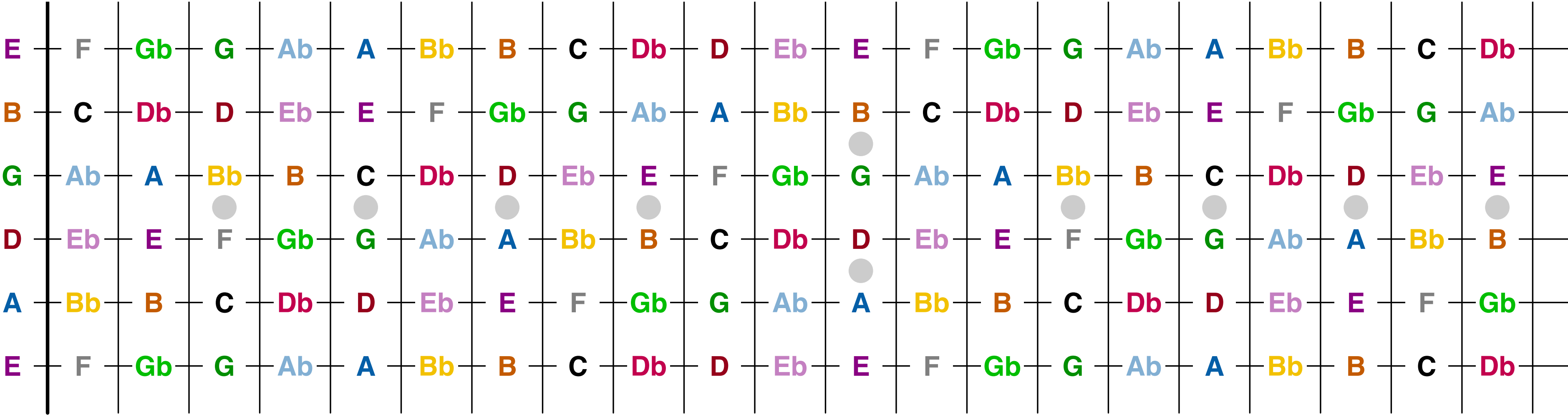
Imperial



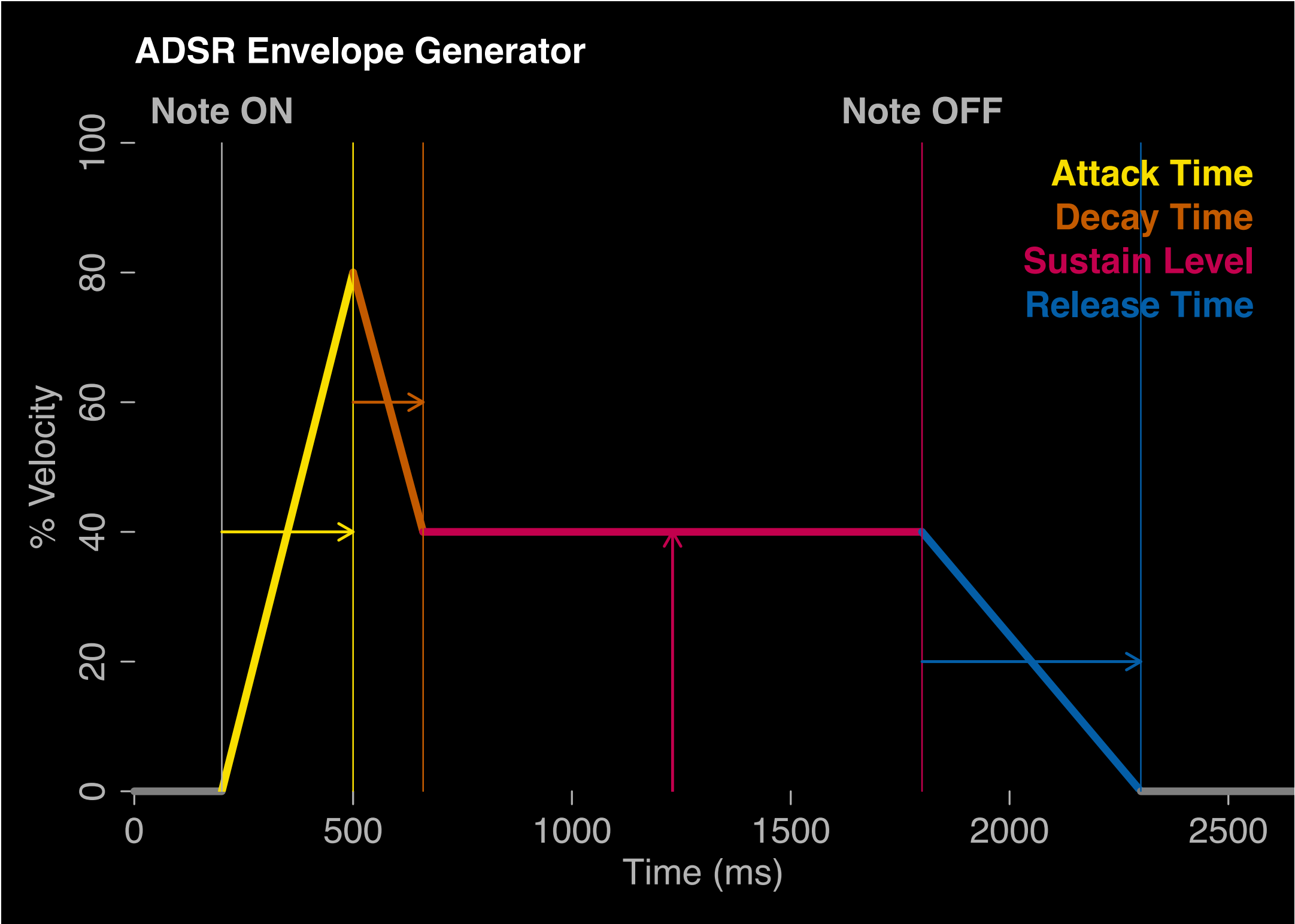
Penn



mplot3.fret



mplot3.adsr



Web Apps powered by **rtemis**

Setup

Plot type

☐ index

☐ histogram

☐ density

☒ scatter

Var 1

ageAtGo1Scan

Var 2

mprage_mars_gmd_R_Ains

Scatter fit

☐ none

☐ lm

☒ gam

☒ 2 * Standard Error of fit

Group

sex

Filter

GMDV.exclude

Age range

8

23.5

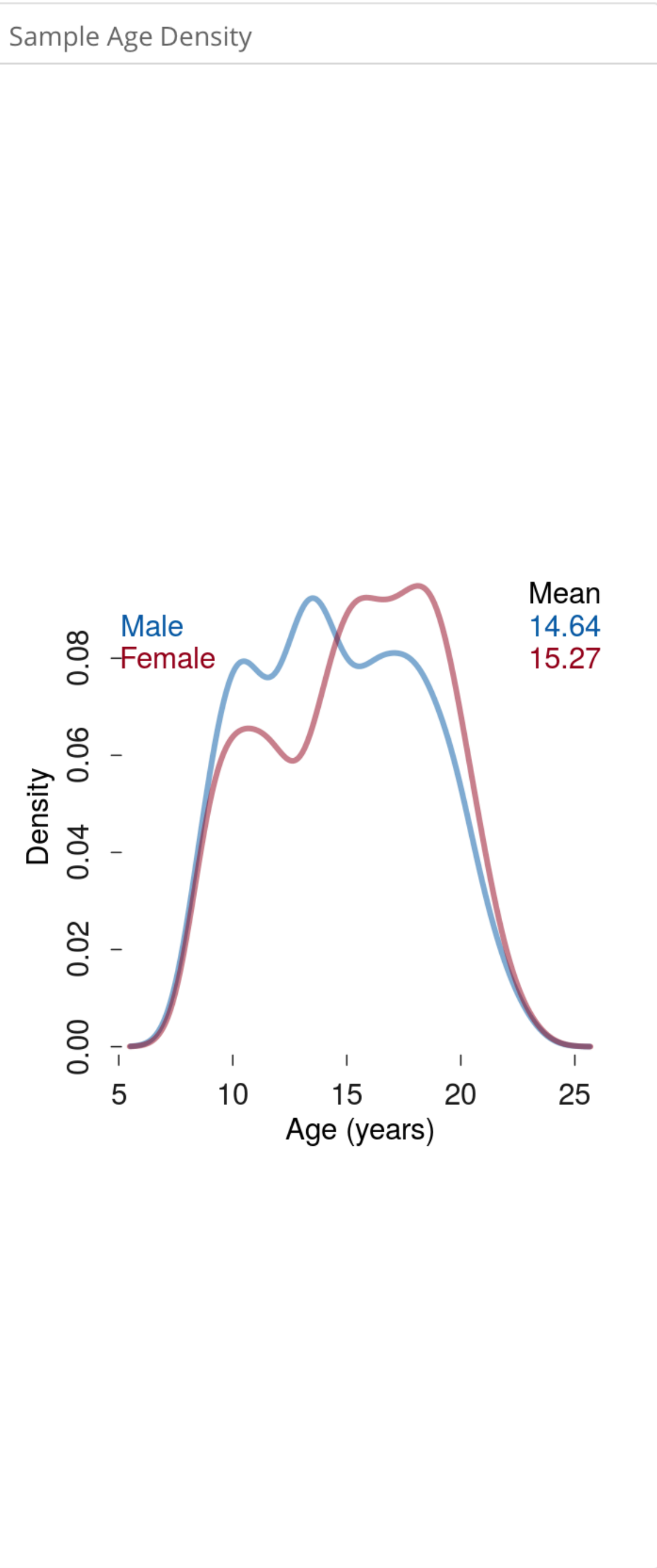
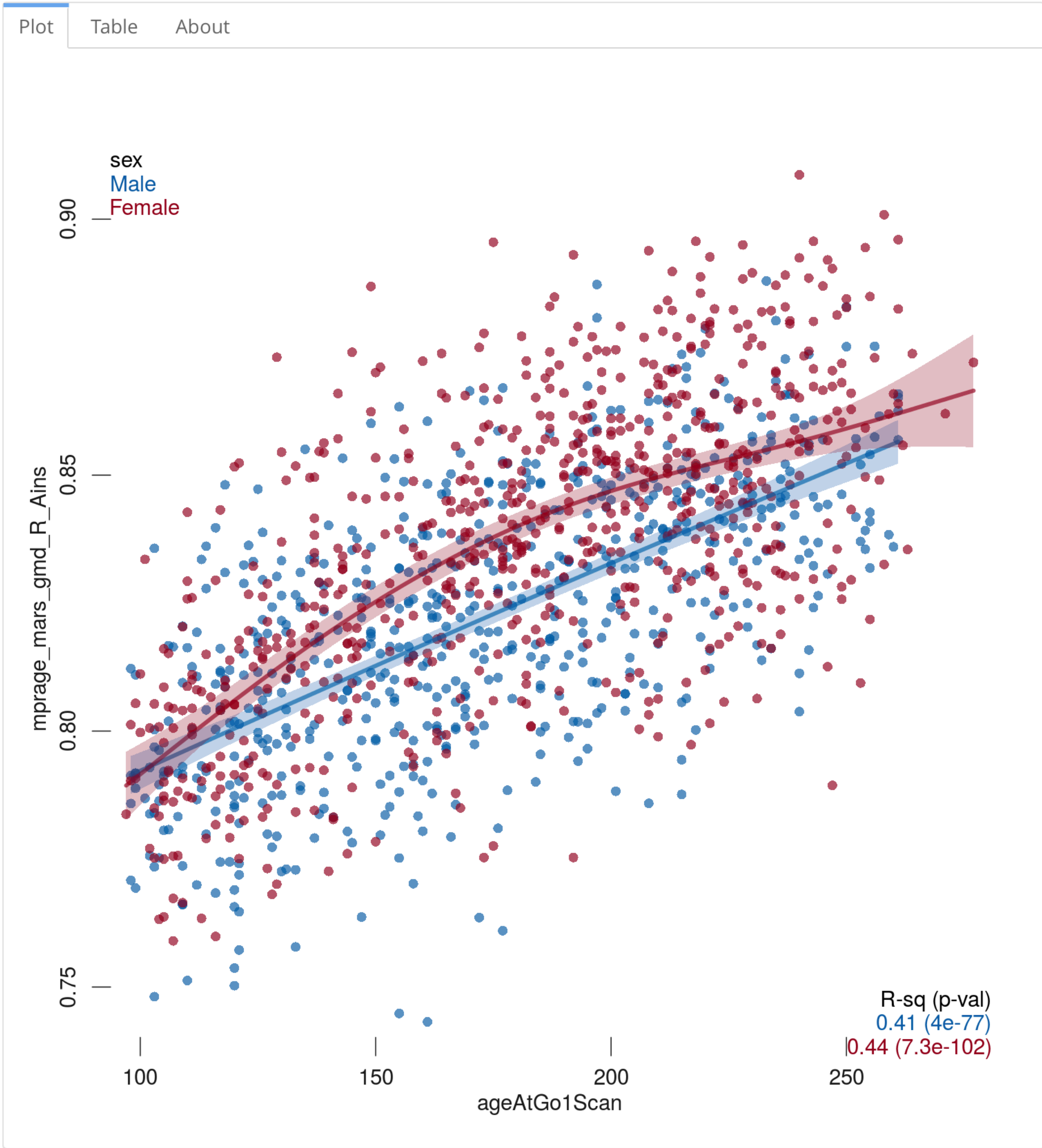
Sex

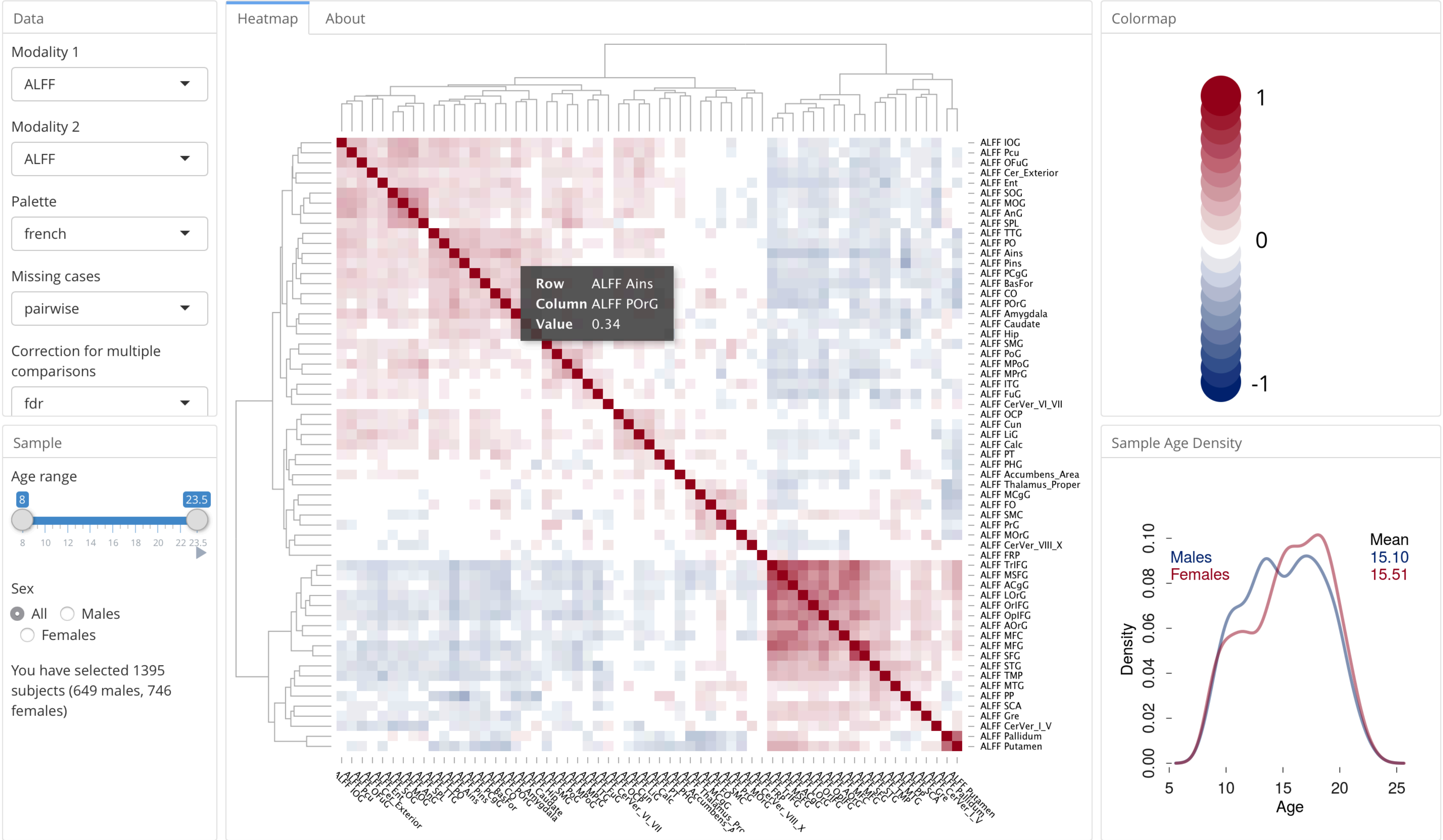
☒ All

☐ Males

☐ Females

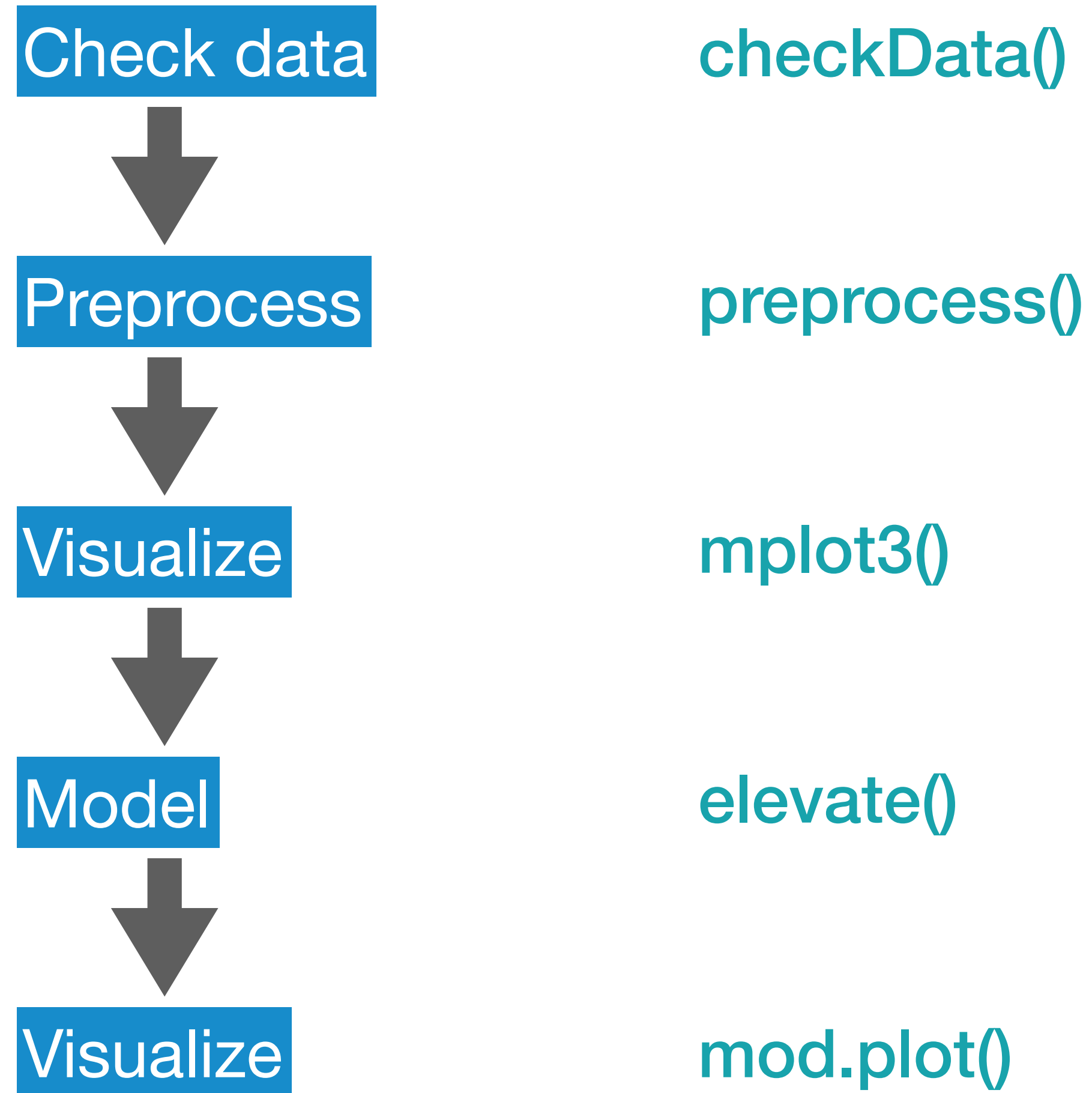
You have selected 1523 subjects (715 males, 808 females)





ML pipelines with **rtemis**

Data analysis pipeline



Supervised Learning in rtemis

Train a **single model**

```
s.CART(dat.train, dat.test, maxdepth = 3)
```

Tune and train a model
(internal resampling)

```
s.CART(dat.train, dat.test, maxdepth = 3:5)
```

Tune, train, and test multiple models
(nested resampling)

```
elevate(dat, mod = 'cart', maxdepth = 3:5)
```


Training and testing a predictive model

10-fold crossvalidation: Training and testing drawn from same population/distribution

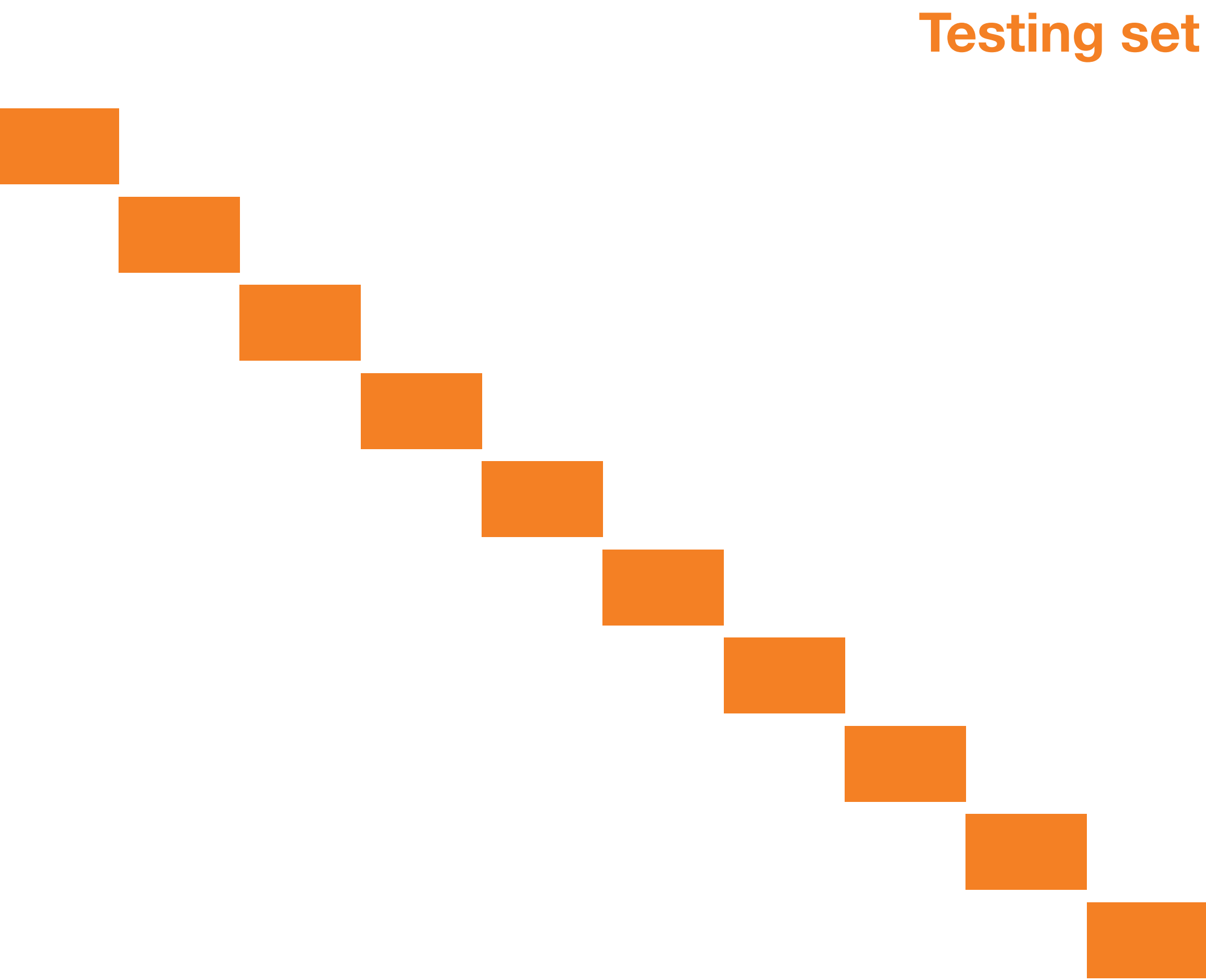
Step 1. Training

Train 10 models on different subsets of the full data



Step 2. Testing

Assess performance on 10 testing subsets and average



Crossvalidation is no panacea

On the Dangers of Cross-Validation. An Experimental Evaluation

R. Bharat Rao

IKM CKS Siemens Medical Solutions USA

Glenn Fung

IKM CKS Siemens Medical Solutions USA

Romer Rosales

IKM CKS Siemens Medical Solutions USA

It is generally accepted that increasing model complexity increases the likelihood of overfitting the data; it is interesting to note that the same applies to overfitting, as it were in the "cross validation space". Note

External test set: Training and testing drawn from separate populations/distribution

Step 1. Training

Train model on training set

Step 2. Testing

Assess performance on external dataset

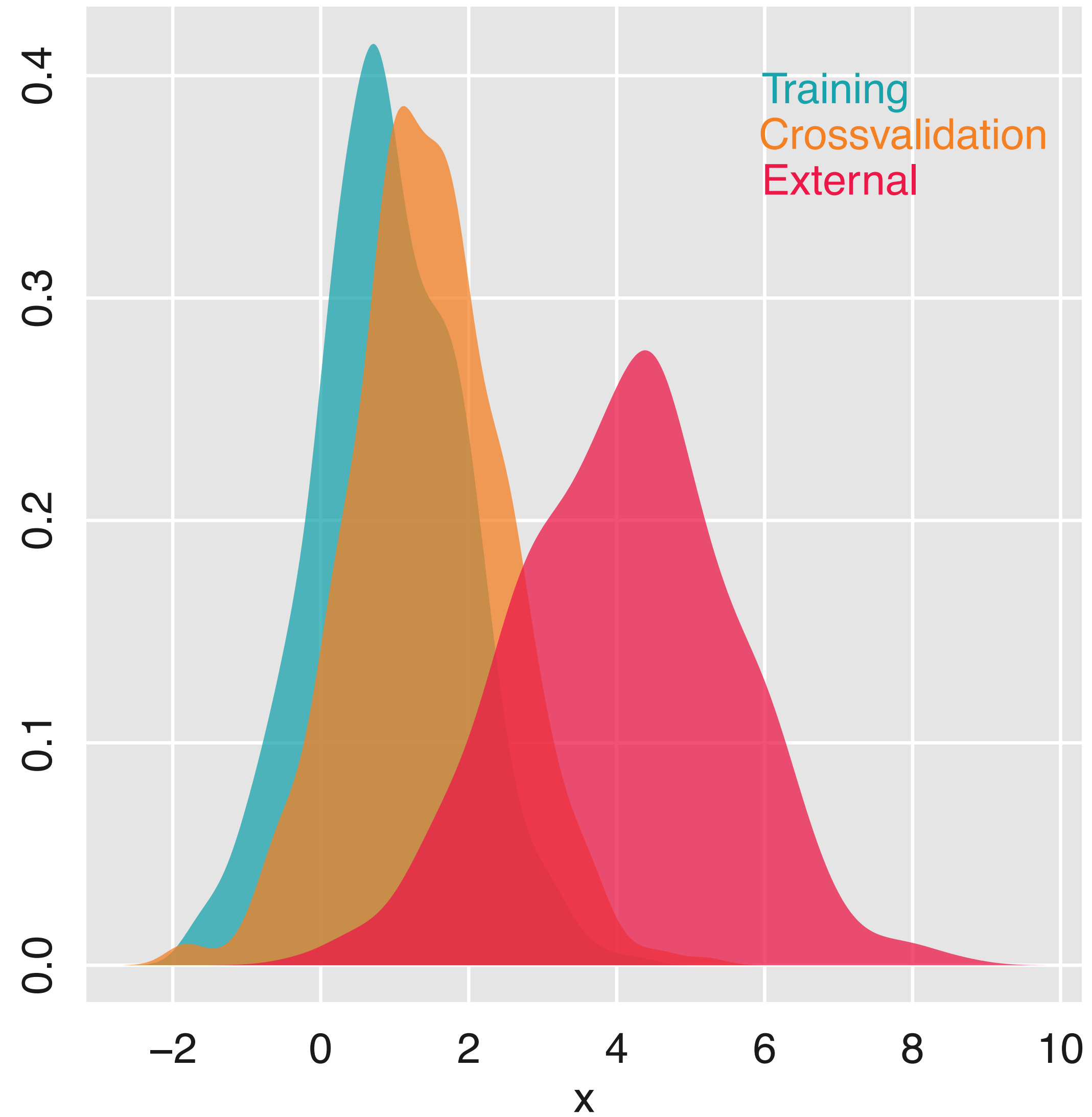
Training set

External test set



Dataset available at training

Why does the population/distribution matter?



[Demo]

rtemis
rtemis.netlify.com



powered by:



MLC



H₂O.ai



plotly

...and many more

Thank you

github.com/egenn/rtemis
rtemis.netlify.com

UCSF