Visually Exploring Random Forests

The ggRandomForests package

John Ehrlinger

Department of Quantitative Health Sciences
Lerner Research Institute
Cleveland Clinic
john.ehrlinger@gmail.com

UseR! 2014

Random Forests

Statistical Modeling: The Two Cultures

Two goals of statistical models:

- Prediction: Predict the response given future observations
- Information: Explain association of covariates to the response

L. Breiman 2001

Random Forests

L. Breiman 2001

- Ensemble of Classification/Regression Trees

randomForest R Package

- RStudio CRAN logs rank: 61

Random Forests

L. Breiman 2001

- Ensemble of Classification/Regression Trees

randomForest R Package

- RStudio CRAN logs rank: 61
- Advantages
  - Predictive Performance (A+)
  - Simple to train/tune
  - Non-parametric/non-linear
  - Built in generalization error estimates

Random Forests

L. Breiman 2001

- Ensemble of Classification/Regression Trees

randomForest R Package

- RStudio CRAN logs rank: 61
- Advantages
  - Predictive Performance (A+)
  - Simple to train/tune
  - Non-parametric/non-linear
  - Built in generalization error estimates
- Disadvantages
  - Information (F)

randomForest

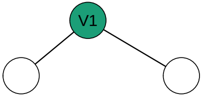
Generic randomForest algorithm

- Bootstrap Data (B)
  - Training set (b)
  - Hold out set (oob)

randomForest

Generic randomForest algorithm

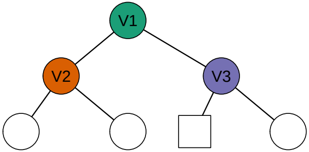
- Bootstrap Data (B)
  - Training set (b)
  - Hold out set (oob)
- A Split Rule



randomForest

Generic randomForest algorithm

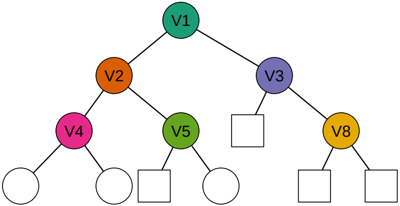
- Bootstrap Data (B)
  - Training set (b)
  - Hold out set (oob)
- A Split Rule
- A Stopping Rule



randomForest

Generic randomForest algorithm

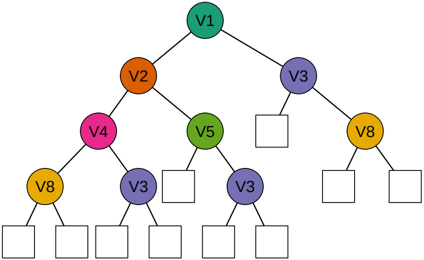
- Bootstrap Data (B)
  - Training set (b)
  - Hold out set (oob)
- A Split Rule
- A Stopping Rule



randomForest

Generic randomForest algorithm

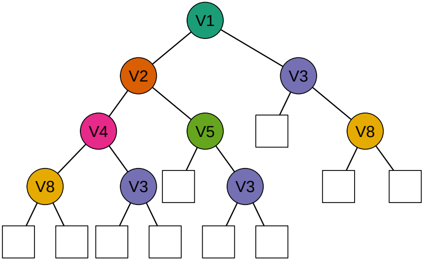
- Bootstrap Data (B)
  - Training set (b)
  - Hold out set (oob)
- A Split Rule
- A Stopping Rule
- Tree Estimates



randomForest

Generic randomForest algorithm

- Bootstrap Data (B)
  - Training set (b)
  - Hold out set (oob)
- A Split Rule
- A Stopping Rule
- Tree Estimates
- Aggregate for Forest Estimates



randomForests for Survival

Ishwaran et al., 2008

randomForestSRC package: A unified treatment for

- Survival
- Regression
- Classification

randomForests for Survival

Ishwaran et al., 2008

randomForestSRC package: A unified treatment for

- Survival
- Regression
- Classification
- Advantages
  - randomForests for Survival
  - Parallel Execution (OpenMP)
  - Minimal Depth Variable Selection

randomForests for Survival

Ishwaran et al., 2008

randomForestSRC package: A unified treatment for

- Survival
- Regression
- Classification
- Advantages
  - randomForests for Survival
  - Parallel Execution (OpenMP)
  - Minimal Depth Variable Selection
- Disadvantages
  - Some performance optimization remains
  - Graphics. . .

ggRandomForest package

Goal: Simplify graphical representation of randomForests.

In progress:
`https://github.com/ehrlinger/ggRandomForests`

ggRandomForest package

Goal: Simplify graphical representation of randomForests.

In progress:
`https://github.com/ehrlinger/ggRandomForests`

- Extracts data.frame objects from a randomForest[SRC].
- Create ggplot graphic elements from each data.frame type.

Unified graphics for Survival, Regression and Classification Forests

Example: Heart Surgery Data

Yoon et.al. 2010

Four surgical treatments:

CABG, CABG+MVR, CABG+SVR, Transplant

- 1466 patients (observations n)
- 46 covariates (predictors p)
- randomForest imputation for missing data.
- 2 separate outcomes (response)
  - Hospital Death (binary, events=43)
  - Survival time with censoring (events=444)

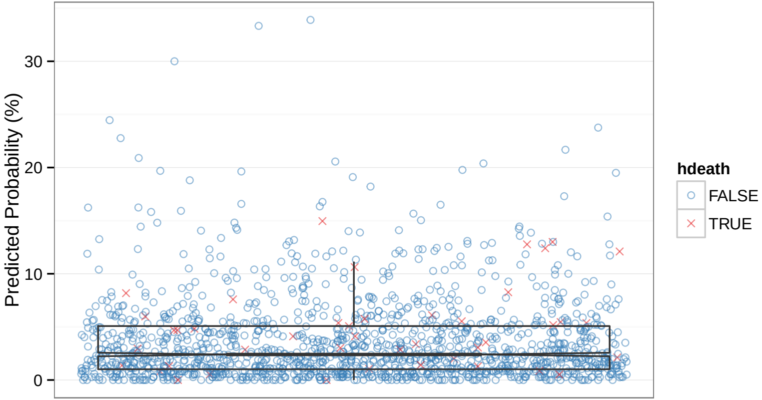
Classification Forests

```
# randomForestSRC classification forest
rf.cls = rfsrc(hdeath~., data=dta.rfc,
               ntree=ntree )

# ggRandomForests default (predicted values)
plot.ggRFsrc(rf.cls)
```

Classification - predicted probability

Hospital Death

```
plot.ggRFsrc(rf.cls)
```

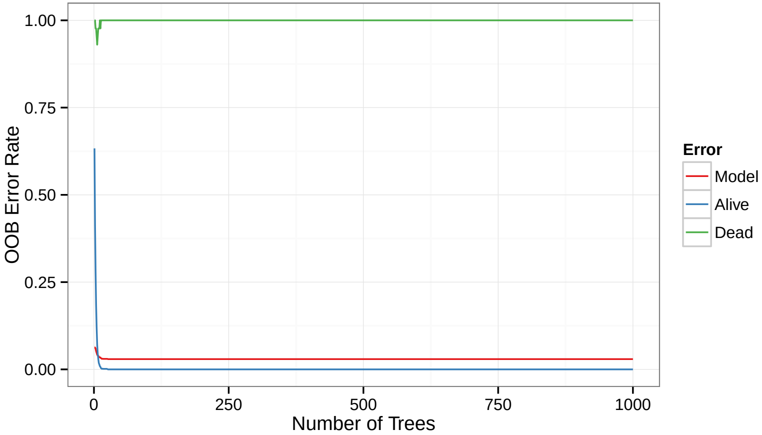


ggError function

```r
# ggRandomForest error convergence rate
gg.err = ggError(rf.cls)
plot(gg.err)

# or...
plot.ggError(rf.cls)
```

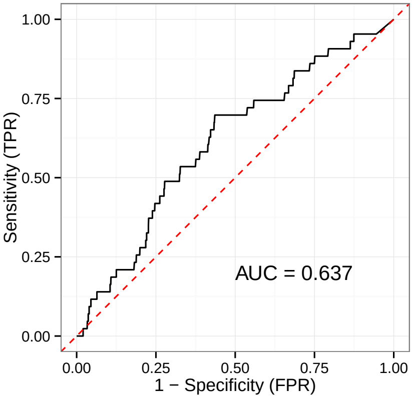
ggError function

```
plot.ggError(rf.cls)
```



ROC curves

```
plot.ggROC(rf.cls)
```



Random Forests for Survival

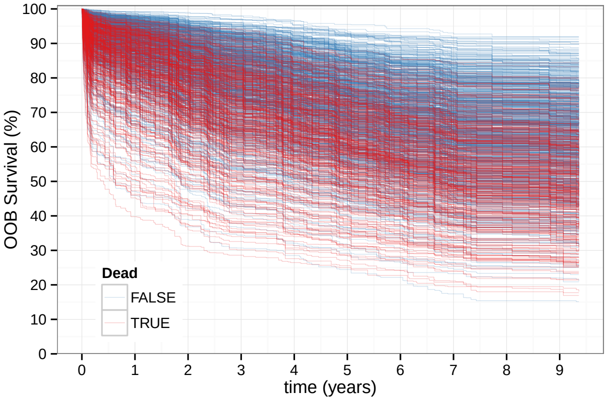
```
# randomForestSRC survival forest
rf.surv = rfsrc(Surv(ivdead,dead)~.,
                data = dta.rfs,
                ntree = ntree)

# ggRandomForests default (predicted survival)
plot.ggRFsrc(rf.surv)
```

Alternatively:

```
# ggRFsrc data object
srvData = ggRFsrc(rf.surv)
plot(srvData)
```

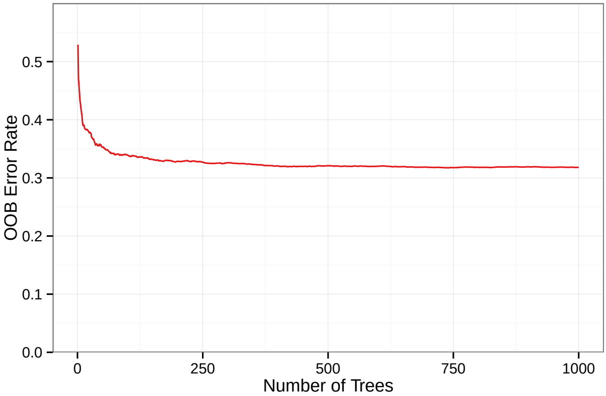
Random Forests for Survival

```
plot.ggRFsrc(rf.surv)
```



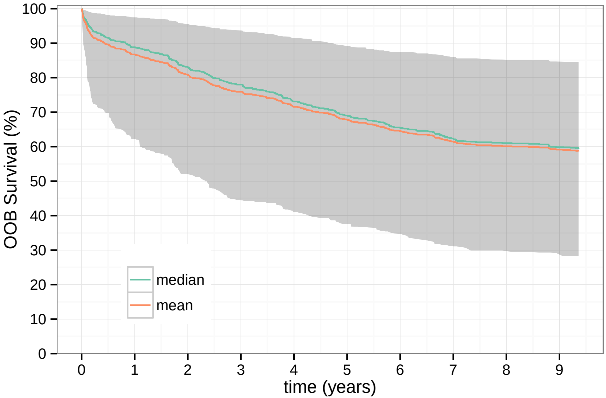
ggError Function

```
plot.ggError(rf.surv)
```



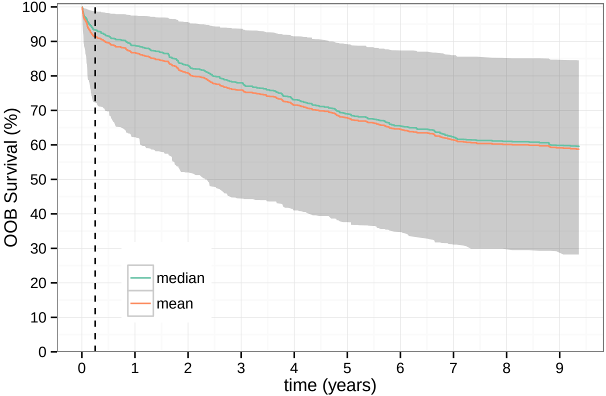
Survival Forests

```
plot.ggRFsrc(rf.surv, se=.95)
```



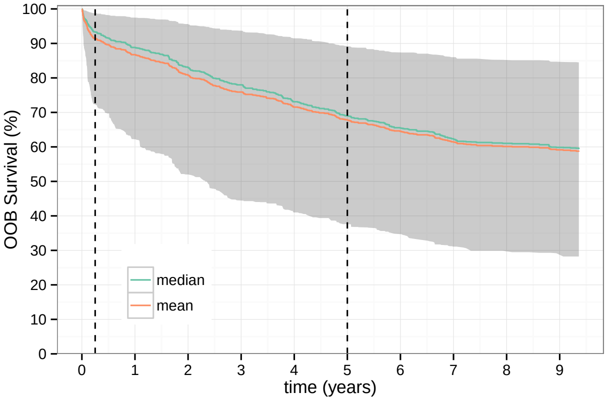
Survival Forests

```
plot.ggRFsrc(rf.surv, se=.95)
```

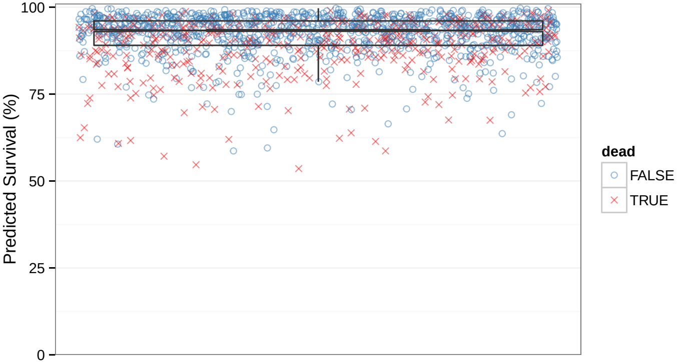


Survival Forests

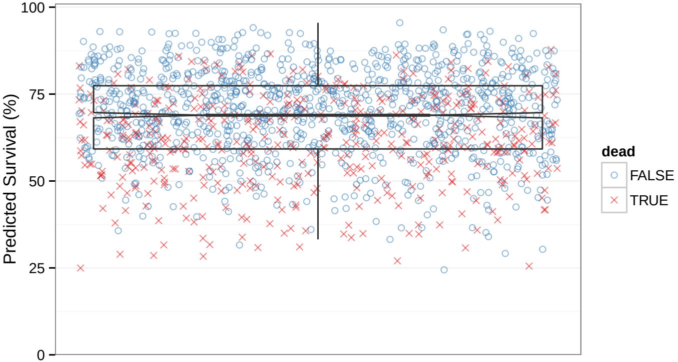
```
plot.ggRFsrc(rf.surv, se=.95)
```



Survival Forests (3 month)



Survival Forests (5 year)



But how do randomForests predict?

We want the good prediction ... and information too!

But how do randomForests predict?

We want the good prediction . . . and information too!

- Which Variables contribute?
  - Variable Importance (VIMP)
  - Minimal Depth

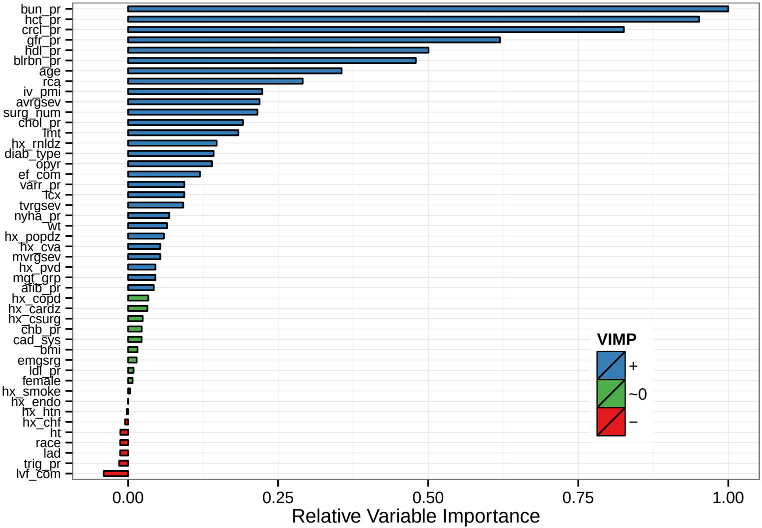
But how do randomForests predict?

We want the good prediction . . . and information too!

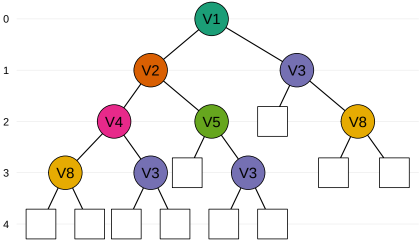
- Which Variables contribute?
  - Variable Importance (VIMP)
  - Minimal Depth
- How do Variables contribute?
  - Variable Dependence plots
  - Partial Dependence plots

Variable Importance

```
vimp.plt=plot.ggVimp(rf.surv)
```



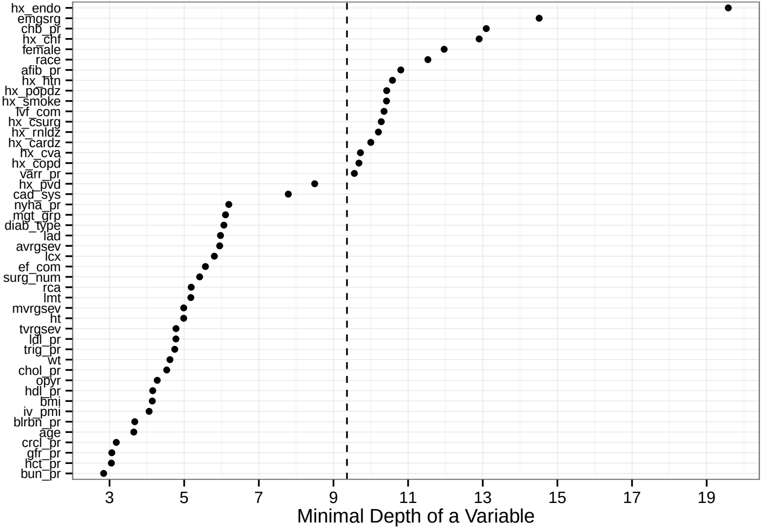
Minimal Depth



- Average (minimal) split distance from the root node (0) over the entire forest
- Measure of how a variable segregates the population

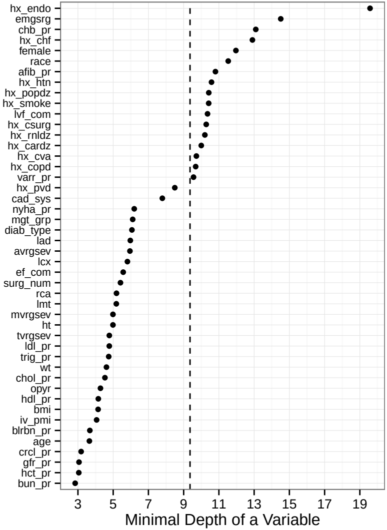
Minimal Depth

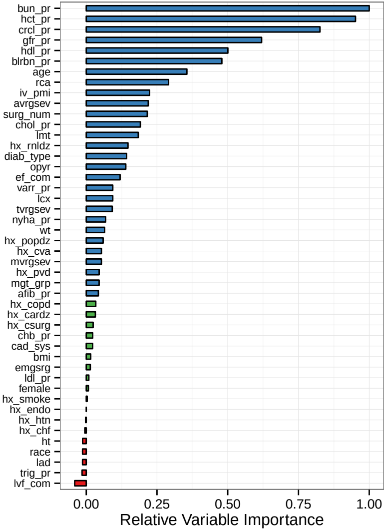
```r
md.plt=plot.ggMinimalDepth(rf.surv)
```



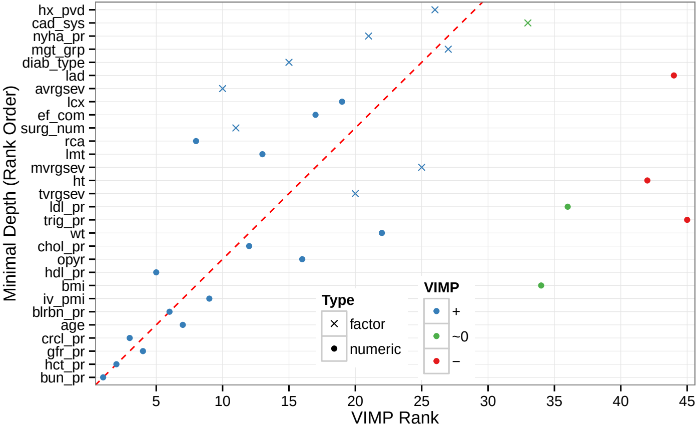
Minimal Depth and VIMP

```
grid.arrange(md.plt, vimp.plt)
```

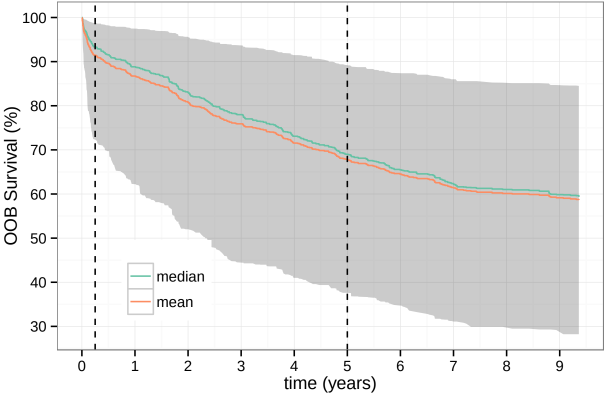




Minimal Depth and VIMP

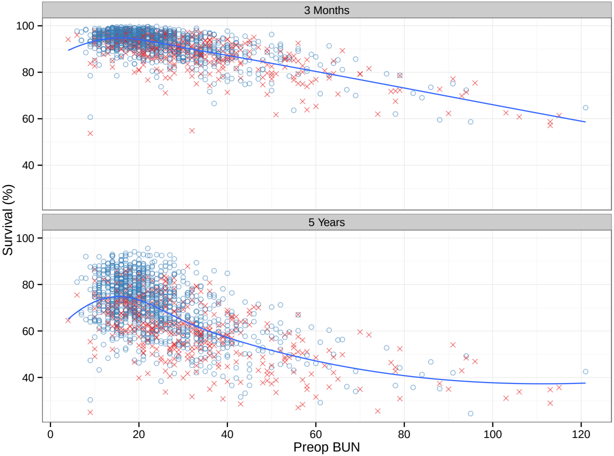


How do variables contribute?



Variable Dependence Plot

```
plot.ggVariable(rf.surv, vars="bun_pr",
                time=c(.25, 5))
```

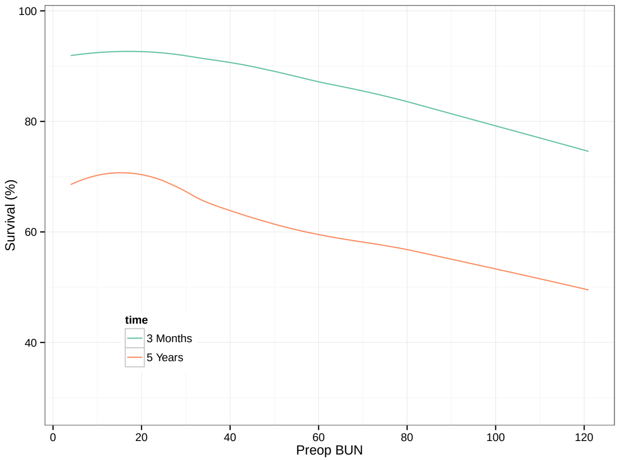


Partial Variable Dependence

```
# randomForestSRC partial plots
rf.part = plot.variable(rf.surv,
                        xvar.names = "bun_pr",
                        partial=TRUE,
                        time=c(.25,5),
                        show.plots = FALSE)

# ggRandomForests plot function
plot.ggPartial(rf.part)
```

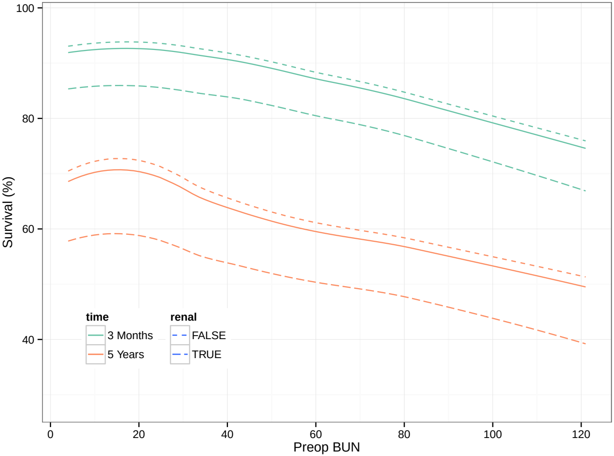
Partial Variable Dependence

```
plot.ggPartial(rf.part,...)
```

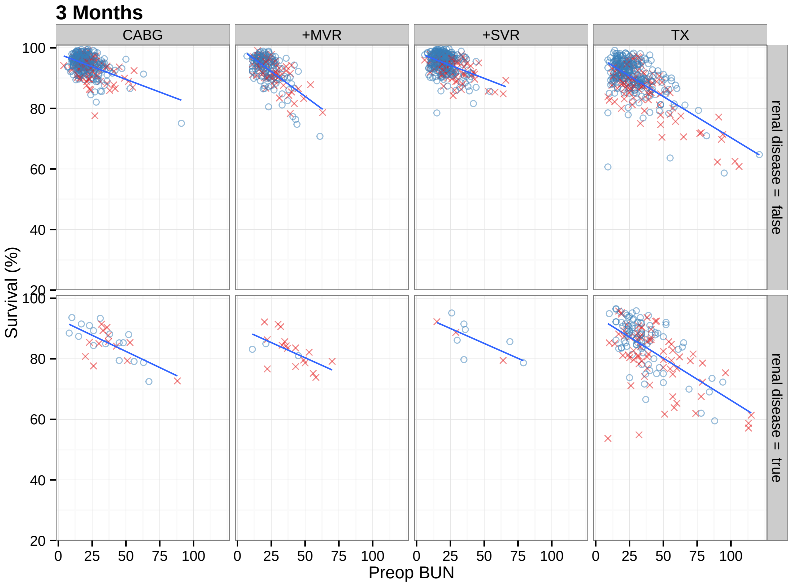


Partial Variable Dependence

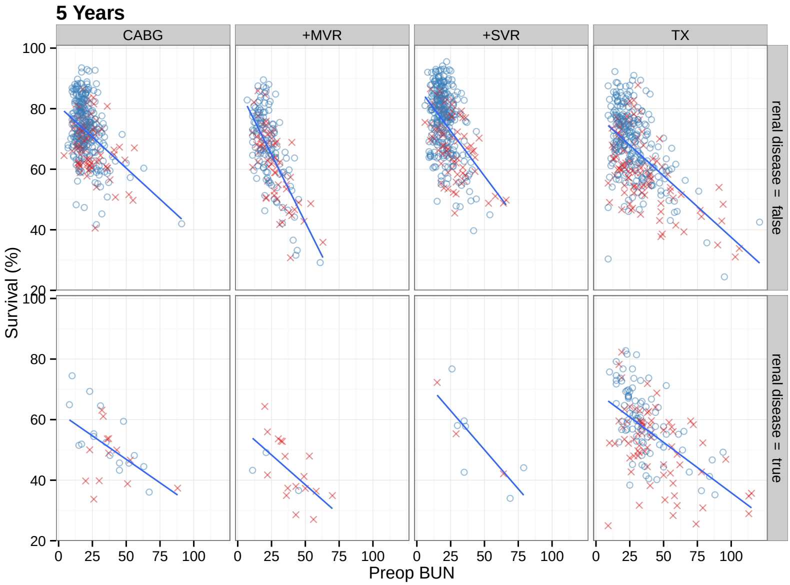
```
plot.ggPartial(rf.part ,...)
```



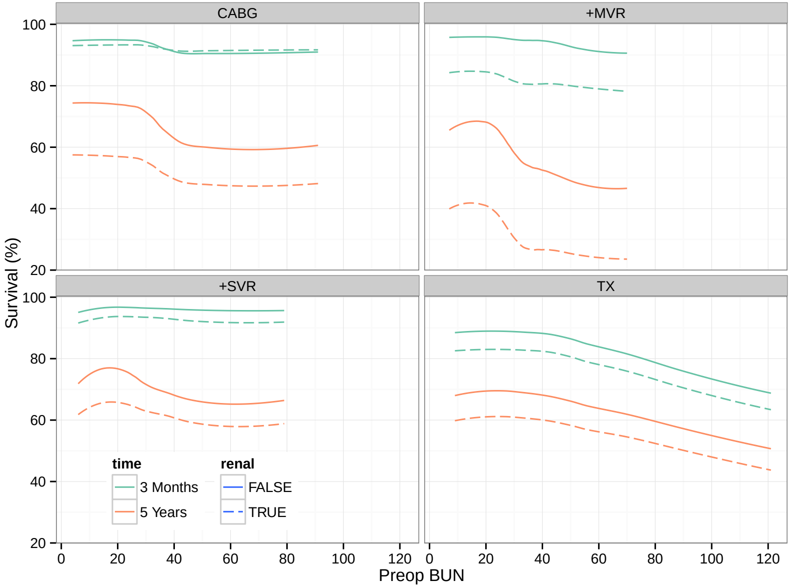
Conditional Plots



Conditional Plots



Partial Dependence Coplots



The ggRandomForest Package

For good prediction . . . and information too!

- Which Variables contribute?
  - Variable Importance (VIMP) - mispecification
  - Minimal Depth - segmentation and selection
- How do Variables contribute?
  - Variable Dependence plots - Covariate Trends
  - Partial Dependence plots - Risk Adjusted Trends

ggRandomForests

Unified graphics for Survival, Regression and Classification Forests

`https://github.com/ehrlinger/ggRandomForests`

john.ehrlinger@gmail.com

References I

- Breiman, L. (2001b). "Statistical Modeling: The Two Cultures". In: *Statistical Science* 16.3, pp. 199–231.
- Breiman, L. (2001a). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32.
- Liaw, A. and M. Wiener (2002). "Classification and Regression by randomForest". In: *R News* 2.3, pp. 18–22.
- Ishwaran, H. et al. (2008). "Random survival forests". In: *The Annals of Applied Statistics* 2.3, pp. 841–860.
- Ishwaran, H. and U. B. Kogalur (2014). *Random Forests for Survival, Regression and Classification (RF-SRC), R package version 1.5.2.*
- Wickham, H. (2009). *ggplot2: elegant graphics for data analysis*. Springer New York.

References II

Yoon, D. Y. et al. (2010). "Decision support in surgical management of ischemic cardiomyopathy". In: *The Journal of Thoracic and Cardiovascular Surgery* 139.2, pp. 283–293.