Using cross_validate()

Ludvig R. Olsen, Benjamin Zachariae

10/13/2016

Contents

Setting up

Start by setting the working directory.

Importing cross_validate()

Include cross_validate.R using source() so that we can use its functions
Change the path in source() to point to the file or put cross_validate.R in the working directory

```
source('cross_validate.R')
```

Loading data

```
require(graphics)
df = mtcars

df$am = as.factor(df$am)

head(df)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Using cross_validate()

Arguments

model: 'y~a+b+(1|c)'

data: Dataframe

id_column: Unique identifiers (e.g. subject, ID, or likewise)

cat_col: categorical column for balancing folds

nfolds: number of folds

family: gaussian or binomial

REML: Restricted Maximum Likelihood

cutoff: For deciding prediction class from prediction (binomial)

positive: Level from dependent variable to predict (1/2) (Levels are alphabetically ordered) *(binomial)*

do.plot: ROC curve plot *(binomial)*

which_plot: choice between available plots
- 'all' plots
- single plot (e.g. 'RMSE')
- list of plots (e.g. c('RMSE', 'r2'))

plot_theme: which theme to use with ggplot2

seed: A number for setting seed. Makes sure the folds are the same for model comparison.

model_verbose: Printed feedback on the used model (lm() / lmer() / glm() / glmer()) (BOOL)

Outputs

Gaussian outputs

RMSE: The mean Root Mean Square Error of all the folds

r2m: The mean marginal R-squared of all the folds

r2c: The mean conditional R-squared of all the folds

AIC: The mean Akaike Information Criterion of all the folds

AICc: The mean Corrected Akaike Information Criterion of all the folds

BIC: The mean Bayesian Information Criterion of all the folds

Folds: How many folds were used?

Convergence Warnings: A count of the folds where the model did not converge

Binomial outputs

ROC curve

AUC: Area Under the Curve

CI1: Confidence interval 1

CI2: Confidence interval 2 (same as AUC)

CI3: Confidence interval 3

Confusion Matrix

Kappa: Comparison of the Observed Accuracy and the Expected Accuracy (random chance)
((observed accuracy - expected accuracy)/(1 - expected accuracy))
Read more about Kappa

Sensitivity: true positive rate (TPR)
(n true positives / n true positives + n false negatives)

Specificity: true negative rate (TNR)
(n true negatives / n false positives + n true negatives)

Pos Pred Value: proportions of positive results (Positive Prediction Value)
(n true positives / n true and false positives)
Calculated from sensitivity, specificity, and prevalence

Neg Pred Value: proportions of negative results (Negative Prediction Value)
(n true negatives / n true and false negatives)
Calculated from sensitivity, specificity, and prevalence

Precision: same as Pos Pred Value

Recall: same as Sensitivity

F1: harmonic mean of precision and sensitivity
((1+beta^2) * precision * recall / ((beta^2 * precision) + recall) - where beta = 1

Prevalence: ??? proportion of positive found to be affecting a particular population ???
(n true positives + n false negatives / n total predictions)

Detection Rate:
(n true positives / n total predictions)

Detection Prevalence:
(n true and false positives / n total predictions)

Balanced Accuracy:
(sensitivity+specificity)/2

- Check also ?confusionMatrix from {caret} for formulas used

cross_validate()

Folds: How many folds were used?

Convergence Warnings: A count of the folds where the model did not converge

lm()

```
cross_validate(('mpg~am+cyl'), df, 1, 'am', nfolds=5, seed=1)
```

```
## [1] "Used lm()"
## [1] "Used lm()"
## [1] "Used lm()"
## [1] "Used lm()"
## [1] "Used lm()"
```

```
##                 RMSE                   r2m                   r2c
##            2.9609141             0.7466532             0.7466532
##                  AIC                  AICc                   BIC
##          134.4310978          136.3820358          139.3955944
##                Folds Convergence Warnings
##            5.0000000             0.0000000
```

lmer()

```
cross_validate(('mpg~am+cyl+(1|disp)'), df, 1, 'am',
               nfolds=5, REML=FALSE, seed=1)
```

```
## [1] "Used lme4::lmer()"
## [1] "Used lme4::lmer()"
## [1] "Used lme4::lmer()"
## [1] "Used lme4::lmer()"
## [1] "Used lme4::lmer()"
```

```
##                 RMSE                   r2m                   r2c
##            2.9763431             0.7684761             0.9359944
##                  AIC                  AICc                   BIC
##          134.7742421          137.8515530          140.9798628
##                Folds Convergence Warnings
##            5.0000000             0.0000000
```

glm()

```
cross_validate(('am~mpg+cyl'), df, 1, 'am', nfolds=5,
               family='binomial', seed=1)
```

```
## [1] "Used glm()"
## [1] "Used glm()"
## [1] "Used glm()"
## [1] "Used glm()"
## [1] "Used glm()"
```

```
##                  AUC                  CI1                  CI2
##            0.7570850            0.5718954            0.7570850
##                  CI3                Kappa          Sensitivity
##            0.9422746            0.3360996            0.7894737
##          Specificity       Pos Pred Value       Neg Pred Value
##            0.5384615            0.7142857            0.6363636
##            Precision               Recall                   F1
##            0.7142857            0.7894737            0.7500000
##           Prevalence       Detection Rate  Detection Prevalence
##            0.5937500            0.4687500            0.6562500
##    Balanced Accuracy                Folds Convergence Warnings
##            0.6639676            5.0000000            0.0000000
```

glmer()

```
cross_validate(('am~mpg+cyl+(1|disp)'), df, 1, 'am',
               nfolds=5, family='binomial', seed=1)
```

```
## [1] "Used lme4::glmer()"
## [1] "Used lme4::glmer()"
## [1] "Used lme4::glmer()"
## [1] "Used lme4::glmer()"
## [1] "Used lme4::glmer()"
```

```
##                  AUC                  CI1                  CI2
##            0.7692308            0.5968977            0.7692308
##                  CI3                Kappa          Sensitivity
##            0.9415639            0.3793103            0.8947368
##          Specificity       Pos Pred Value       Neg Pred Value
##            0.4615385            0.7083333            0.7500000
##            Precision               Recall                   F1
##            0.7083333            0.8947368            0.7906977
##           Prevalence       Detection Rate  Detection Prevalence
##            0.5937500            0.5312500            0.7500000
##    Balanced Accuracy                Folds Convergence Warnings
##            0.6781377            5.0000000            0.0000000
```

model_verbose

In order to ensure the user that cross_validate() chooses the right model type (lm,lmer,glm or glmer), it automatically prints the type used for every fold. This doesn't necessarily look pretty, so it's possible to turn off this feature by setting model_verbose to FALSE.

```
cross_validate(('mpg~am+cyl'), df, 1, 'am', nfolds=5,
               seed=1, model_verbose = FALSE)
```

```
##                RMSE                 r2m                 r2c
##           2.9609141           0.7466532           0.7466532
##                 AIC                AICc                 BIC
##         134.4310978         136.3820358         139.3955944
##               Folds Convergence Warnings
##           5.0000000           0.0000000
```

Plotting

The built-in plotting options allow the user to visualise the process. Depending on the family (gaussian or binomial) a variety of plots are available.

Binomial currently only plots the *ROC curve*

Gaussian allows for the following plot options:

'RMSE' - boxplot of the Root Mean Square Errors from each folds

'r2' - boxplot of the R-squared values (both marginal and conditional) from each fold

'IC' - boxplot of the Information Criterion (AIC and BIC) from each fold

'coefficients' - boxplot of the model estimates of the fixed effects for each fold

'all' prints all the available plots

Choose multiple plots with a list: c('RMSE', 'coefficients')

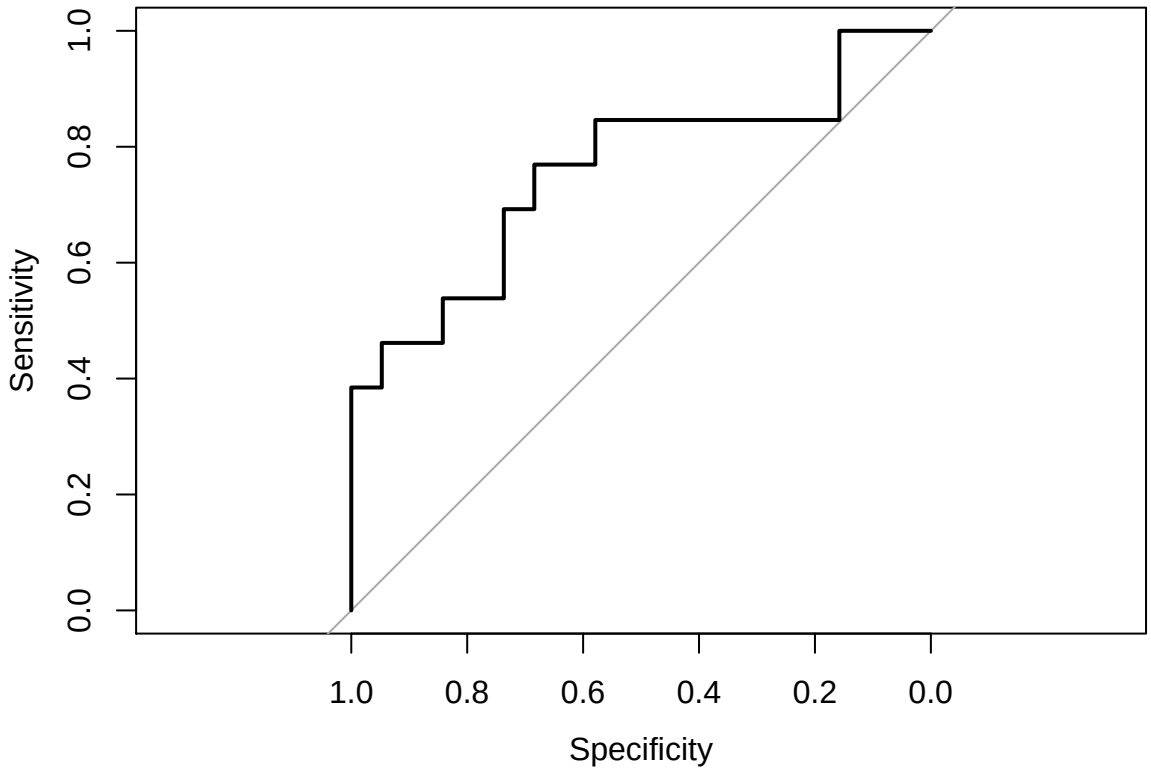
Arguments
Set the do.plot argument to print the plots:
do.plot = TRUE

Choose the plots you want with which_plot:
which_plot = 'all' *(only used for gaussian models)*

Set your favourite ggplot2 theme:
plot_theme = theme_bw() *(only used for gaussian models)*

Binomial plots

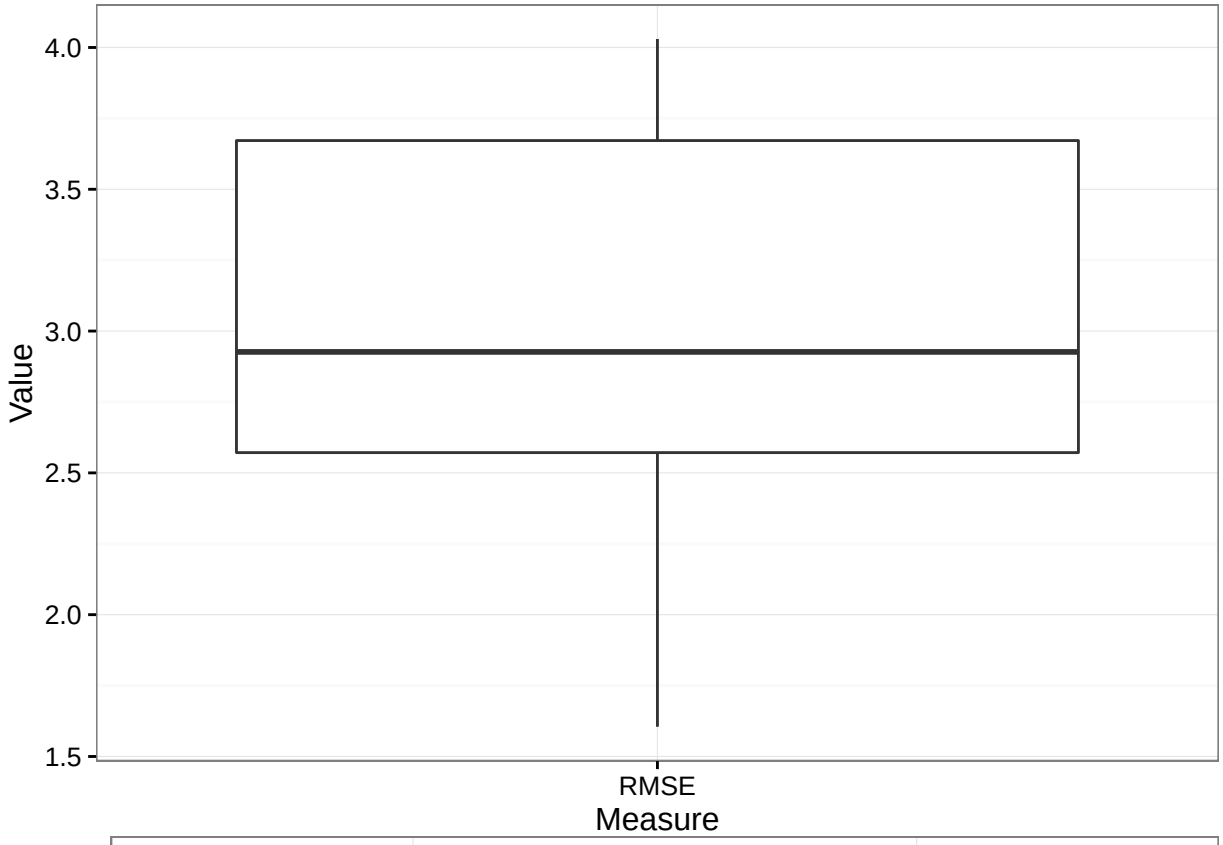
```
cross_validate(('am~mpg+cyl'), df, 1, 'am', nfolds=5,
               family='binomial', seed=1,
               model_verbose = FALSE, do.plot = TRUE,
               plot_theme = theme_bw())
```

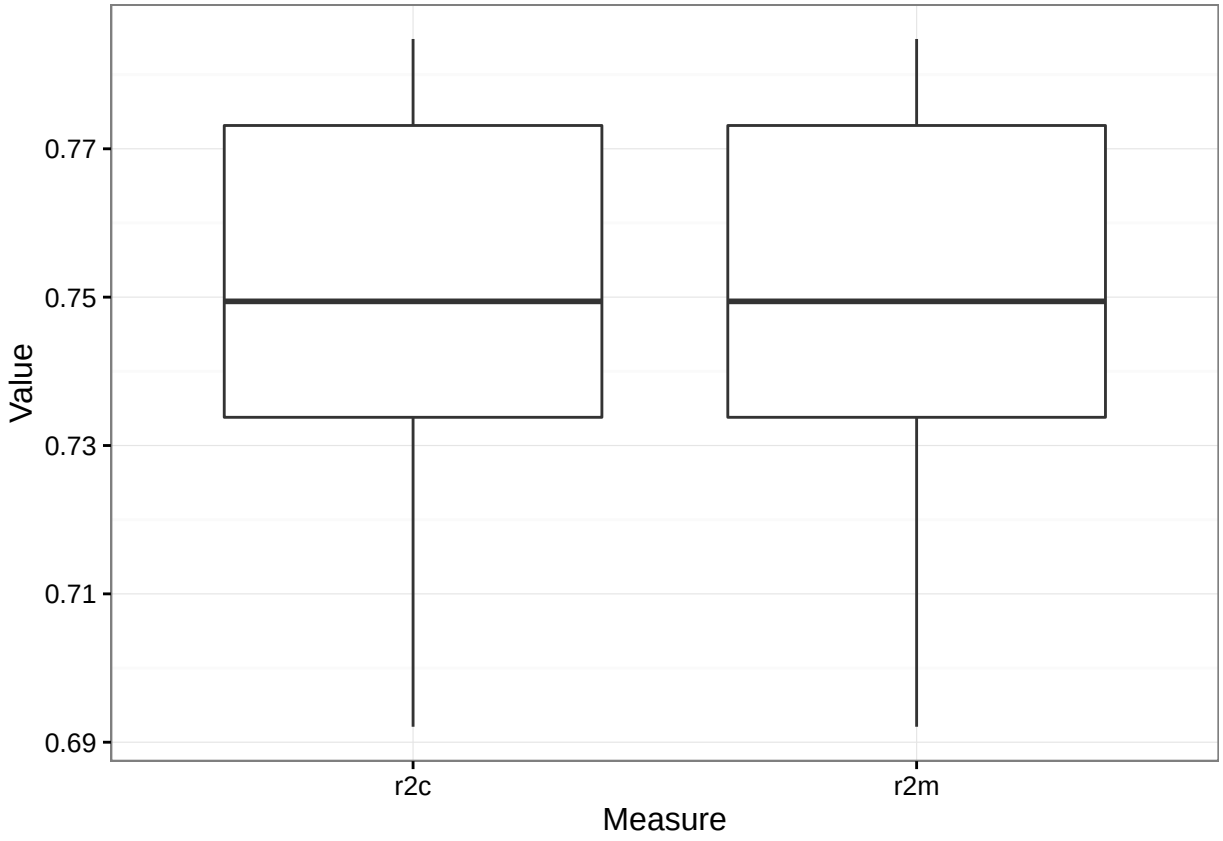


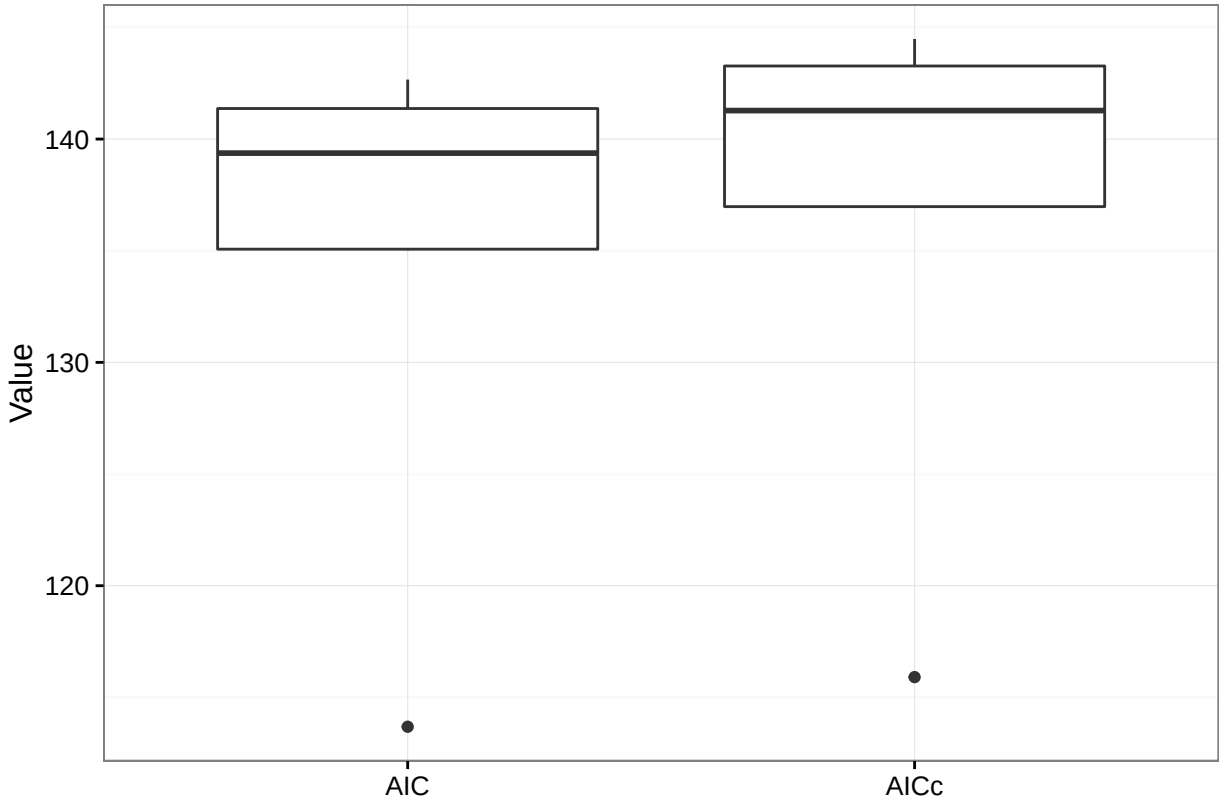

```
##                 AUC                 CI1                 CI2
##           0.7570850           0.5718954           0.7570850
##                 CI3               Kappa         Sensitivity
##           0.9422746           0.3360996           0.7894737
##         Specificity      Pos Pred Value      Neg Pred Value
##           0.5384615           0.7142857           0.6363636
##           Precision              Recall                  F1
##           0.7142857           0.7894737           0.7500000
##          Prevalence      Detection Rate Detection Prevalence
##           0.5937500           0.4687500           0.6562500
##    Balanced Accuracy               Folds Convergence Warnings
##           0.6639676           5.0000000           0.0000000
```

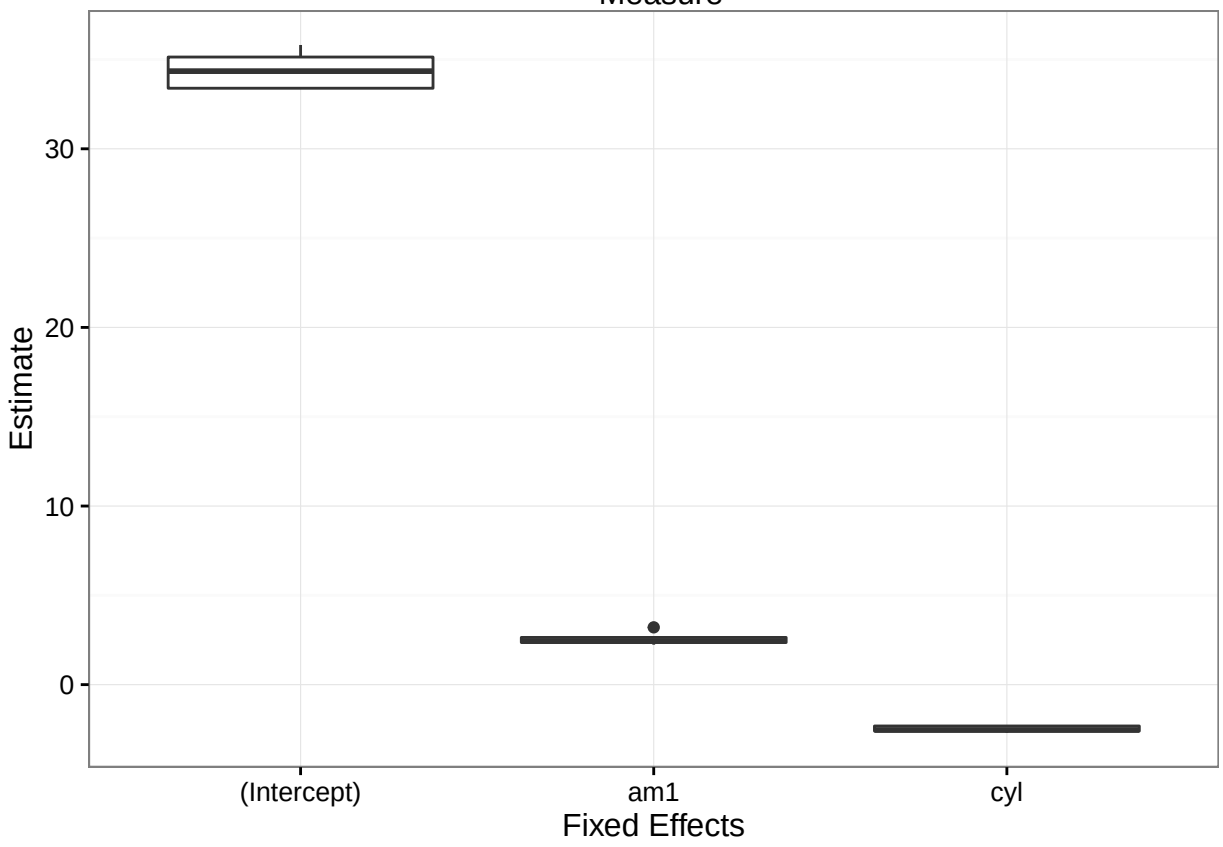
Gaussian plots

```
cross_validate(('mpg~am+cyl'), df, 1, 'am', nfolds=5,
               seed=1, model_verbose = FALSE,
               do.plot = TRUE, which_plot = 'all',
               plot_theme = theme_bw())
```


4.0
3.5
3.0
2.5
2.0
1.5
Value
RMSE
Measure


0.77
0.75
0.73
0.71
0.69
Value
r2c
r2m
Measure


140
130
120
Value
AIC
AICc
Measure


30
20
10
0
Estimate
(Intercept)
am1
cyl
Fixed Effects

```
##                RMSE                 r2m                 r2c
##           2.9609141           0.7466532           0.7466532
##                 AIC                AICc                 BIC
##         134.4310978         136.3820358         139.3955944
##               Folds Convergence Warnings
##           5.0000000           0.0000000
```

Using cross_validate_list()

If you have a list of models that you wish to compare, cross_validate_list() makes it really easy. You pass it the list and it returns a dataframe with the results of the models.
N.B. remember to set the seed, so all models use the same folds.

lm()

```
models_lm = c("mpg~am", "mpg~am+cyl", "mpg~am+cyl+wt")

models_lm_df = cross_validate_list(models_lm, df, 1, 'am',
                                   seed=1, model_verbose=FALSE)

models_lm_df
```

```
##       RMSE       r2m       r2c      AIC     AICc      BIC Folds
## 1 4.967704 0.3495878 0.3495878 157.7222 158.8380 161.4455     5
## 2 2.960914 0.7466532 0.7466532 134.4311 136.3820 139.3956     5
## 3 2.662724 0.8148838 0.8148838 127.0694 130.1467 133.2750     5
##   Convergence Warnings dependent     fixed
## 1                    0       mpg        am
## 2                    0       mpg    am+cyl
## 3                    0       mpg am+cyl+wt
```

lmer()

```
models_lmer = c("mpg~am+(1|disp)", "mpg~am+cyl+(1|disp)", "mpg~am+cyl+wt+(1|disp)")

models_lmer_df = cross_validate_list(models_lmer, df, 1, 'am',
                                     seed=1, model_verbose=FALSE)

models_lmer_df
```

```
##       RMSE       r2m       r2c      AIC     AICc      BIC Folds
## 1 5.096974 0.3560127 0.9414678 155.7834 157.7344 160.7479     5
## 2 2.976343 0.7684761 0.9359944 134.7742 137.8516 140.9799     5
## 3 2.756166 0.8352889 0.9033470 128.4559 132.9986 135.9027     5
##   Convergence Warnings dependent     fixed random
## 1                    0       mpg        am 1|disp
## 2                    0       mpg    am+cyl 1|disp
## 3                    0       mpg am+cyl+wt 1|disp
```

lm() and lmer() at once

Notice that we get NA in the random column with the lm() model, as it doesn't contain random effects.

```
models_lm_mixed = c("mpg~am", "mpg~am+(1|disp)")

models_lm_mixed_df = cross_validate_list(models_lm_mixed, df, 1, 'am',
                                         seed=1, model_verbose=FALSE)

models_lm_mixed_df
```

```
##       RMSE       r2m       r2c      AIC     AICc      BIC Folds
## 1 4.967704 0.3495878 0.3495878 157.7222 158.8380 161.4455     5
## 2 5.096974 0.3560127 0.9414678 155.7834 157.7344 160.7479     5
##   Convergence Warnings dependent fixed random
## 1                    0       mpg    am   <NA>
## 2                    0       mpg    am 1|disp
```

glm()

Notice that we get warning messages.
If these are convergence warnings, they will be counted, so you can discard the model.
Else you will have to read the warning messages that specifies the model and the fold where the warning was issued.

```
models_glm = c("am~mpg", "am~mpg+cyl", "am~mpg+cyl+wt")

models_glm_df = cross_validate_list(models_glm, df, 1, 'am',
                                    family='binomial', seed=1,
                                    model_verbose=FALSE)
```

```
## ------------------------

## cross_validate(): Warning:

## In model:

## am~mpg+cyl+wt

## In fold:

## 2

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
models_glm_df
```

```
##         AUC       CI1       CI2       CI3     Kappa Sensitivity
## 1 0.8056680 0.6313771 0.8056680 0.9799589 0.3949580   0.8421053
## 2 0.7570850 0.5718954 0.7570850 0.9422746 0.3360996   0.7894737
## 3 0.8582996 0.7166857 0.8582996 0.9999135 0.6800000   0.8421053
##   Specificity Pos Pred Value Neg Pred Value Precision    Recall        F1
## 1   0.5384615      0.7272727      0.7000000 0.7272727 0.8421053 0.7804878
## 2   0.5384615      0.7142857      0.6363636 0.7142857 0.7894737 0.7500000
## 3   0.8461538      0.8888889      0.7857143 0.8888889 0.8421053 0.8648649
##   Prevalence Detection Rate Detection Prevalence Balanced Accuracy Folds
## 1    0.59375        0.50000              0.68750         0.6902834     5
## 2    0.59375        0.46875              0.65625         0.6639676     5
## 3    0.59375        0.50000              0.56250         0.8441296     5
##   Convergence Warnings  dependent      fixed
## 1                    0         am        mpg
## 2                    0         am    mpg+cyl
## 3                    0         am mpg+cyl+wt
```

glmer()

```
models_glmer = c("am~mpg+(1|disp)", "am~mpg+cyl+(1|disp)", "am~mpg+cyl+wt+(1|disp)")

models_glmer_df = cross_validate_list(models_glmer, df, 1, 'am',
                                      family='binomial', seed=1,
                                      model_verbose=FALSE)

models_glmer_df
```

```
##         AUC       CI1       CI2       CI3     Kappa Sensitivity
## 1 0.8259109 0.6801943 0.8259109 0.9716276 0.4410480   0.9473684
## 2 0.7692308 0.5968977 0.7692308 0.9415639 0.3793103   0.8947368
## 3 0.8765182 0.7398379 0.8765182 1.0000000 0.7408907   0.8947368
##   Specificity Pos Pred Value Neg Pred Value Precision    Recall        F1
## 1   0.4615385      0.7200000      0.8571429 0.7200000 0.9473684 0.8181818
## 2   0.4615385      0.7083333      0.7500000 0.7083333 0.8947368 0.7906977
## 3   0.8461538      0.8947368      0.8461538 0.8947368 0.8947368 0.8947368
##   Prevalence Detection Rate Detection Prevalence Balanced Accuracy Folds
## 1    0.59375        0.56250              0.78125         0.7044534     5
## 2    0.59375        0.53125              0.75000         0.6781377     5
## 3    0.59375        0.53125              0.59375         0.8704453     5
##   Convergence Warnings  dependent      fixed random
## 1                    0         am        mpg 1|disp
## 2                    0         am    mpg+cyl 1|disp
## 3                    0         am mpg+cyl+wt 1|disp
```

Convergence Warnings

If we get convergence warnings while fitting our model on a fold, the values of that fold will return NA and we will count the warning (see convergence_warnings in the output).
The model and fold that didn't converge are messaged.
Whenever you are comparing models, consider discarding the ones with convergence warnings.

```
models_conv = c("am~cyl+wt+qsec+vs+carb+(1|disp)", "am~mpg+cyl+wt+(1|disp)")

models_conv_df = cross_validate_list(models_conv, df, 1, 'am',
                                     family = 'binomial', seed=1,
                                     model_verbose=FALSE)
```

```
## -------------------------------------

## cross_validate(): Convergence Warning:

## In model:

## am~cyl+wt+qsec+vs+carb+(1|disp)

## In fold:

## 1

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : unable to evaluate scaled gradient

## -------------------------------------

## cross_validate(): Convergence Warning:

## In model:

## am~cyl+wt+qsec+vs+carb+(1|disp)

## In fold:

## 2

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : unable to evaluate scaled gradient

## -------------------------------------

## cross_validate(): Convergence Warning:

## In model:

## am~cyl+wt+qsec+vs+carb+(1|disp)

## In fold:

## 3
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : unable to evaluate scaled gradient

## ------------------------------------

## cross_validate(): Convergence Warning:

## In model:

## am~cyl+wt+qsec+vs+carb+(1|disp)

## In fold:

## 4

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : unable to evaluate scaled gradient

## ------------------------------------

## cross_validate(): Convergence Warning:

## In model:

## am~cyl+wt+qsec+vs+carb+(1|disp)

## In fold:

## 5

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : unable to evaluate scaled gradient

## Confusion Matrix error as the model didn't converge.

## Receiver Operator Characteristic (ROC) Curve error as the model didn't converge.
```

```
models_conv_df
```

```
##         AUC       CI1       CI2 CI3     Kappa Sensitivity Specificity
## 1        NA        NA        NA  NA        NA          NA          NA
## 2 0.8765182 0.7398379 0.8765182   1 0.7408907   0.8947368   0.8461538
##   Pos Pred Value Neg Pred Value Precision    Recall        F1 Prevalence
## 1             NA             NA        NA        NA        NA         NA
## 2      0.8947368      0.8461538 0.8947368 0.8947368 0.8947368    0.59375
##   Detection Rate Detection Prevalence Balanced Accuracy Folds
## 1             NA                   NA                NA     5
## 2        0.53125              0.59375         0.8704453     5
##   Convergence Warnings  dependent               fixed random
## 1                    5        am cyl+wt+qsec+vs+carb 1|disp
## 2                    0        am          mpg+cyl+wt 1|disp
```