# Boston

*Beate and Oliver*

*1/16/2020*

## Goal of this script

We want to implement linear transformation models in NN and compare the achieved NLL and estimated ceofficients with the MLT results.

We fit a transformation function $h : (y|x) \to (z|x)$ with the property $(z|x) = h(y|x) \sim N(0, 1)$

In a **linear** transformation model the transformation function has the special form: $h_Y(y) - \sum_i \beta_i x_i$

Then we know, that.

- $F_{Y|X=x}(y) = F_z(h_Y(y) - \sum_i \beta_i x_i)$

## Importing the required packages

```
library(MASS)
library(ggplot2)
```

```
## Warning: As of rlang 0.4.0, dplyr must be at least version 0.8.0.
## x dplyr 0.7.6 is too old for rlang 0.4.2.
## i Please update dplyr with `install.packages("dplyr")`.
```

```
library(mlt)
```

```
## Warning: package 'mlt' was built under R version 3.5.2
```

```
## Loading required package: basefun
```

```
## Warning: package 'basefun' was built under R version 3.5.2
```

```
## Loading required package: variables
```

```
## Warning: package 'variables' was built under R version 3.5.2
```

```
##
## Attaching package: 'variables'
```

```
## The following object is masked from 'package:ggplot2':
##
##     unit
```

```
library(basefun)
library(keras)
library(tensorflow)
```

```
## Warning: package 'tensorflow' was built under R version 3.5.2
```

```
library(tfprobability)
```

```
## Warning: package 'tfprobability' was built under R version 3.5.2
```

```
T_STEPS = 2000
```

Source functions h and h_dash in w and w/o batch magic

```
# source("mlt_utils.R")  # eg scaling fct
# # preparing eval_h an eval_h_dash, fct implemented in tfp
# source("mlt_utils_keras_v2.R")  # causes error when knittering
#source('https://raw.githubusercontent.com/tensorchiefs/dl_playr/master/mlt/bern_utils.R')
source('~/Documents/workspace/dl_playr/mlt/bern_utils.R')
source('data.R')
```

## Loading the data

We scale the y-varible to [0,1]

```
xy_dat = get_data_boston()
```

```
##  [1] "Names in X : crim"    "Names in X : zn"      "Names in X : indus"
##  [4] "Names in X : chas"    "Names in X : nox"     "Names in X : rm"
##  [7] "Names in X : age"     "Names in X : dis"     "Names in X : rad"
## [10] "Names in X : tax"     "Names in X : ptratio" "Names in X : b"
## [13] "Names in X : lstat"
```

```
dat = xy_dat$dat
sum(dat$y**2) # 299626.3 to compare with BH data in paper
```

```
## [1] 97.90634
```

```
dat$y_obs = dat$y
dat$y = NULL
y_range = xy_dat$scale
dat$y_scale = dat$y_obs
dat$y_obs = NULL
x = xy_dat$x
y = xy_dat$y
```

## Defining the model

We set up the formula for the model:

```
fm_large = (y_scale ~ crim + zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + b + lstat)
#fm_small = (y_scale ~  rm + lstat) #lm log lik 346
#fm_uni = (y_scale ~  rm)
(fm = fm_large)
```

```
## y_scale ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
##     tax + ptratio + b + lstat
```

```
is_univariate = TRUE
sum(dat$rm**2)  # 20234.6 to compare with BH data in paper
```

```
## [1] 20234.6
```

## Baseline Linear Model

```
fit_lm = lm(fm, data=dat)
fit_lm$coef
```

```
##   (Intercept)          crim            zn         indus          chas
##  6.990997e-01 -2.400252e-03  1.031566e-03  4.568584e-04  5.970520e-02
##           nox            rm           age           dis           rad
## -3.948136e-01  8.466367e-02  1.538277e-05 -3.279037e-02  6.801100e-03
##           tax       ptratio             b         lstat
## -2.741021e-04 -2.117216e-02  2.069263e-04 -1.166130e-02
```

```
(logLik_lm=logLik(fit_lm) )/nrow(dat) + log(y_range)# the larger the better
```

```
## 'log Lik.' 4.651261 (df=15)
```

## MLT fit and results

### Variable and Model definition and fit

```
nb = 8  # order defining the Number of Bernstein fct in polynom
len_theta = nb+1
# specify a numeric variable with data in [0,1] and principle bounds [0,Inf]
var_y <- numeric_var("y_scale", support = c(0, 1), bounds = c(-Inf, Inf), add = c(0,0))
# what is done with the bound information (default bounds c(-INF, INF)

# set up monoton increasing polynomial of order nb with Bernstein basis function
bb <- Bernstein_basis(var_y, order=nb, ui="increasing")

# set up grid in interval supp+add -> gives data.frame with col y_scale
y_grid <- as.data.frame(mkgrid(bb, n = 500))

# set up model for mlt
ctm = ctm(bb, shift=fm[-2L], data=dat, todistr="Normal")
#~-1 + crim
#ctm = ctm(bb, shift = ~ b + crim - 1, data=dat, todistr="Normal")
# fm[-2L] defnes the basis function for the shift term h_y(y) in h(y|x)=h_y(y)+h_x(x)
# the intercept is included in the baseline-trafo h_y(y) (not in linear predictor h_x(x))
```

Fit of the model:

```
# fit the mlt model
mlt_fit <- mlt(ctm, data = dat, verbose=TRUE)
```

## logLik with MLT

```
(logLik_mlt = logLik(mlt_fit))  #  df = nr-theta + nr-beta
```

```
## 'log Lik.' 567.4294 (df=22)
```

```
# compare to logLik of the baseline model - the larger the better
NLL_MLT = -logLik_mlt / nrow(dat) + log(y_range)
```

## Estimated coefficients with MLT

Get the coefficients of the trafo h from the mlt fit:

```
( mlt_fit$coef )
```

```
##  Bs1(y_scale)  Bs2(y_scale)  Bs3(y_scale)  Bs4(y_scale)  Bs5(y_scale)
## -12.949147588  -9.935968220  -9.926923993  -4.953969409  -4.099483539
##  Bs6(y_scale)  Bs7(y_scale)  Bs8(y_scale)  Bs9(y_scale)          crim
##  -3.552333101  -3.543250090  -3.539083121  -2.253323345   0.044629181
##            zn         indus          chas           nox            rm
##  -0.006412397  -0.010904403  -0.583180810   4.724994838  -0.467408980
##           age           dis           rad           tax       ptratio
##   0.002101098   0.293249835  -0.079827518   0.003507471   0.225942649
##             b         lstat
##  -0.002572815   0.161517331
```

```
( theta = mlt_fit$coef[1:(nb+1)] )
```

```
## Bs1(y_scale) Bs2(y_scale) Bs3(y_scale) Bs4(y_scale) Bs5(y_scale)
##   -12.949148    -9.935968    -9.926924    -4.953969    -4.099484
## Bs6(y_scale) Bs7(y_scale) Bs8(y_scale) Bs9(y_scale)
##    -3.552333    -3.543250    -3.539083    -2.253323
```

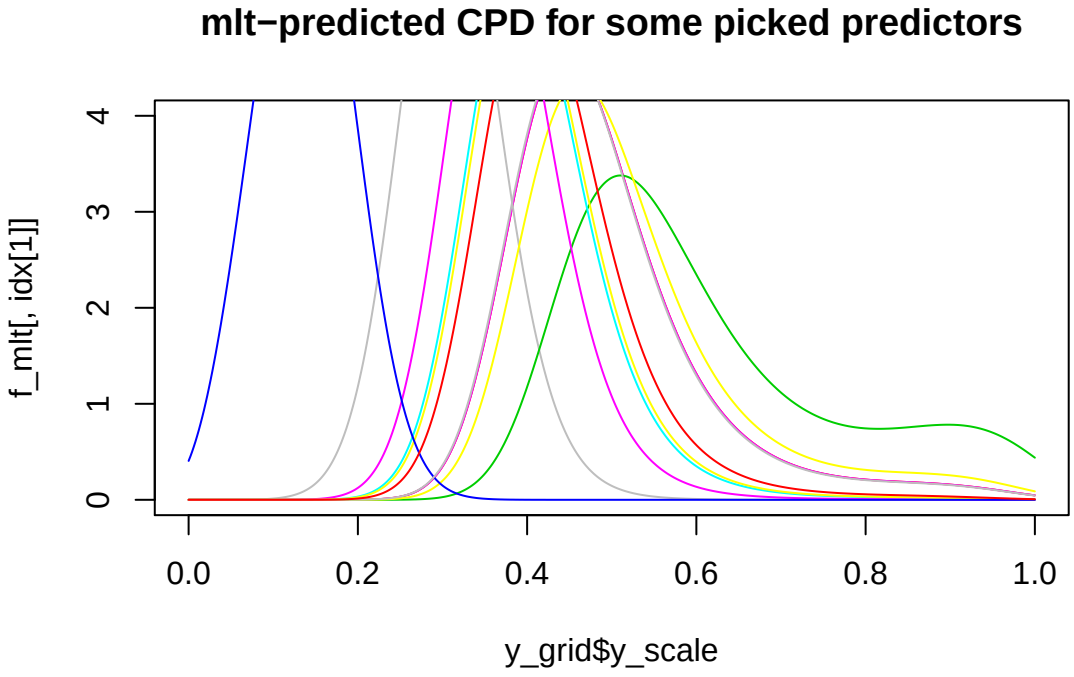
```
( beta = mlt_fit$coef[(nb+2):length(mlt_fit$coef)] )
```

```
##         crim           zn        indus         chas          nox
##  0.044629181 -0.006412397 -0.010904403 -0.583180810  4.724994838
##           rm          age          dis          rad          tax
## -0.467408980  0.002101098  0.293249835 -0.079827518  0.003507471
##      ptratio            b        lstat
##  0.225942649 -0.002572815  0.161517331
```

The conditional PDF for some observations

```
  f_mlt = predict(mlt_fit, newdata=dat,  q=y_grid$y_scale, type='density')

  q_mlt = predict(mlt_fit, newdata=dat,
                  prob=c(0.025,0.25,0.5, 0.75,0.975), type='quantile')
  q_mlt = t(q_mlt)
  #q_mlt = matrix(q_mlt$exact, ncol = 5, byrow = TRUE)
  set.seed(3)
  idx = sample(1:ncol(f_mlt))[1:10]
  m = max(f_mlt[,idx])
  plot(y_grid$y_scale, f_mlt[,idx[1]], type='l',col='red', ylim=c(0,4),
       main="mlt-predicted CPD for some picked predictors")
  for (i in idx){
    lines(y_grid$y_scale, f_mlt[,i], col=i)
  }
```


mlt−predicted CPD for some picked predictors
f_mlt[, idx[1]]
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0
y_grid$y_scale

# NN

## NN approach for a linear shift model, modeled with NN

Fitting means to find the *nb* coefficients *theta* for the Bernsteinpolynom which approximaties the transformation function with *nb* being set to:

```
nb
```

```
## [1] 8
```

## Preparing input and output

```
y = tf$Variable(as.matrix(dat$y_scale)[,drop=FALSE], dtype='float32')
y$shape # has to be (#y,1)
```

```
## (506, 1)
```

```
# conditional - we give the rm-variables as input to the NN
#x = tf$Variable(as.matrix(dat$rm)[,drop=FALSE], dtype='float32')

#x = tf$Variable(as.matrix(dat[,c('rm','lstat'),drop=FALSE]), dtype='float32')

#dat$chas = as.numeric(as.character(dat$chas))
x = tf$Variable(x, dtype='float32') #all
x$shape  # has to be (#y,1) for a univariate model
```

```
## (506, 13)
```

```
source('model_3.R')
source('bern_utils.R')
source("model_utils.R")
x_dim = as.integer(dim(x)[2])
model_3 = new_model_3(len_theta = as.integer(len_theta), x_dim = x_dim, y_range=y_range)
T_OUT = 100
run = 1
history = model_train(model_3, make_hist(),x_train = x, y_train = y, x_test = x, y_test = y, T_STEPS=20
```

```
## [1] "100 model_3: likelihood (in optimize)  6.11696147918701 likelihood (in test)  6.11471080780029"
## [1] "200 model_3: likelihood (in optimize)  5.9031457901001 likelihood (in test)  5.9011058807373"
## [1] "300 model_3: likelihood (in optimize)  5.707435131073 likelihood (in test)  5.70555830001831"
## [1] "400 model_3: likelihood (in optimize)  5.52712059020996 likelihood (in test)  5.52538967132568"
## [1] "500 model_3: likelihood (in optimize)  5.36096286773682 likelihood (in test)  5.35936975479126"
## [1] "600 model_3: likelihood (in optimize)  5.20807552337646 likelihood (in test)  5.20661067962646"
## [1] "700 model_3: likelihood (in optimize)  5.06770896911621 likelihood (in test)  5.06636619567871"
## [1] "800 model_3: likelihood (in optimize)  4.93916988372803 likelihood (in test)  4.93794202804565"
## [1] "900 model_3: likelihood (in optimize)  4.82178831100464 likelihood (in test)  4.82066869735718"
## [1] "1000 model_3: likelihood (in optimize)  4.71489810943604 likelihood (in test)  4.71388006210327
## [1] "1100 model_3: likelihood (in optimize)  4.61783313751221 likelihood (in test)  4.61690950393677
## [1] "1200 model_3: likelihood (in optimize)  4.52992010116577 likelihood (in test)  4.5290846824646"
## [1] "1300 model_3: likelihood (in optimize)  4.4504828453064 likelihood (in test)  4.44972896575928"
## [1] "1400 model_3: likelihood (in optimize)  4.37883949279785 likelihood (in test)  4.37816047668457
## [1] "1500 model_3: likelihood (in optimize)  4.3143048286438 likelihood (in test)  4.31369304656982"
## [1] "1600 model_3: likelihood (in optimize)  4.25619316101074 likelihood (in test)  4.25564241409302
## [1] "1700 model_3: likelihood (in optimize)  4.20382499694824 likelihood (in test)  4.20332765579224
```

```
## [1] "1800 model_3: likelihood (in optimize)  4.15652656555176 likelihood (in test)  4.15607690811157
## [1] "1900 model_3: likelihood (in optimize)  4.1136417388916 likelihood (in test)  4.11323308944702"
## [1] "2000 model_3: likelihood (in optimize)  4.07453870773315 likelihood (in test)  4.07416486740112
## [1] "2100 model_3: likelihood (in optimize)  4.03862190246582 likelihood (in test)  4.03827667236328
## [1] "2200 model_3: likelihood (in optimize)  4.00534105300903 likelihood (in test)  4.00502014160156
## [1] "2300 model_3: likelihood (in optimize)  3.97420525550842 likelihood (in test)  3.9739031791687"
## [1] "2400 model_3: likelihood (in optimize)  3.94478631019592 likelihood (in test)  3.94449949264526
## [1] "2500 model_3: likelihood (in optimize)  3.91672420501709 likelihood (in test)  3.91644930839539
## [1] "2600 model_3: likelihood (in optimize)  3.88972210884094 likelihood (in test)  3.88945651054382
## [1] "2700 model_3: likelihood (in optimize)  3.86354112625122 likelihood (in test)  3.86328268051147
## [1] "2800 model_3: likelihood (in optimize)  3.83799171447754 likelihood (in test)  3.83773899078369
## [1] "2900 model_3: likelihood (in optimize)  3.81292414665222 likelihood (in test)  3.81267547607422
## [1] "3000 model_3: likelihood (in optimize)  3.78822040557861 likelihood (in test)  3.78797483444214
## [1] "3100 model_3: likelihood (in optimize)  3.76378703117371 likelihood (in test)  3.76354384422302
## [1] "3200 model_3: likelihood (in optimize)  3.73955059051514 likelihood (in test)  3.73930907249451
## [1] "3300 model_3: likelihood (in optimize)  3.71545243263245 likelihood (in test)  3.71521210670471
## [1] "3400 model_3: likelihood (in optimize)  3.69144773483276 likelihood (in test)  3.69120788574219
## [1] "3500 model_3: likelihood (in optimize)  3.6675021648407 likelihood (in test)  3.66726279258728"
## [1] "3600 model_3: likelihood (in optimize)  3.6435923576355 likelihood (in test)  3.64335346221924"
## [1] "3700 model_3: likelihood (in optimize)  3.61970472335815 likelihood (in test)  3.61946582794189
## [1] "3800 model_3: likelihood (in optimize)  3.59583377838135 likelihood (in test)  3.59559512138367
## [1] "3900 model_3: likelihood (in optimize)  3.57198309898376 likelihood (in test)  3.57174468040466
## [1] "4000 model_3: likelihood (in optimize)  3.54816293716431 likelihood (in test)  3.54792475700378
## [1] "4100 model_3: likelihood (in optimize)  3.52438926696777 likelihood (in test)  3.52415204048157
## [1] "4200 model_3: likelihood (in optimize)  3.50068378448486 likelihood (in test)  3.50044703483582
## [1] "4300 model_3: likelihood (in optimize)  3.47707056999207 likelihood (in test)  3.47683501243591
## [1] "4400 model_3: likelihood (in optimize)  3.45357656478882 likelihood (in test)  3.45334219932556
## [1] "4500 model_3: likelihood (in optimize)  3.43022966384888 likelihood (in test)  3.42999696731567
## [1] "4600 model_3: likelihood (in optimize)  3.40705823898315 likelihood (in test)  3.40682721138"
## [1] "4700 model_3: likelihood (in optimize)  3.38408994674683 likelihood (in test)  3.38386130332947
## [1] "4800 model_3: likelihood (in optimize)  3.36135220527649 likelihood (in test)  3.36112594604492
## [1] "4900 model_3: likelihood (in optimize)  3.33887100219727 likelihood (in test)  3.33864736557007
## [1] "5000 model_3: likelihood (in optimize)  3.31667041778564 likelihood (in test)  3.31644988059998
## [1] "5100 model_3: likelihood (in optimize)  3.29477381706238 likelihood (in test)  3.29455637931824
## [1] "5200 model_3: likelihood (in optimize)  3.27320337295532 likelihood (in test)  3.27298927307129
## [1] "5300 model_3: likelihood (in optimize)  3.25198006629944 likelihood (in test)  3.25176978111267
## [1] "5400 model_3: likelihood (in optimize)  3.23112440109253 likelihood (in test)  3.23091769218445
## [1] "5500 model_3: likelihood (in optimize)  3.21065545082092 likelihood (in test)  3.21045303344727
## [1] "5600 model_3: likelihood (in optimize)  3.19059252738953 likelihood (in test)  3.19039392471313
## [1] "5700 model_3: likelihood (in optimize)  3.17095232009888 likelihood (in test)  3.17075800895691
## [1] "5800 model_3: likelihood (in optimize)  3.15175080299377 likelihood (in test)  3.15156102180481
## [1] "5900 model_3: likelihood (in optimize)  3.13300132751465 likelihood (in test)  3.13281607627869
## [1] "6000 model_3: likelihood (in optimize)  3.11471390724182 likelihood (in test)  3.11453342437744
## [1] "6100 model_3: likelihood (in optimize)  3.09689593315125 likelihood (in test)  3.09672021865845
## [1] "6200 model_3: likelihood (in optimize)  3.07955026626587 likelihood (in test)  3.07937908172607
## [1] "6300 model_3: likelihood (in optimize)  3.06267642974854 likelihood (in test)  3.06251001358032
## [1] "6400 model_3: likelihood (in optimize)  3.04627013206482 likelihood (in test)  3.04610824584961
## [1] "6500 model_3: likelihood (in optimize)  3.03032302856445 likelihood (in test)  3.0301661491394"
## [1] "6600 model_3: likelihood (in optimize)  3.01482439041138 likelihood (in test)  3.01467180252075
## [1] "6700 model_3: likelihood (in optimize)  2.99976110458374 likelihood (in test)  2.99961256980896
## [1] "6800 model_3: likelihood (in optimize)  2.98511910438538 likelihood (in test)  2.98497486114502
## [1] "6900 model_3: likelihood (in optimize)  2.97088432312012 likelihood (in test)  2.97074389457703
## [1] "7000 model_3: likelihood (in optimize)  2.95704412460327 likelihood (in test)  2.95690751075745
## [1] "7100 model_3: likelihood (in optimize)  2.94358777999878 likelihood (in test)  2.9434552192688"
```

```
## [1] "7200 model_3: likelihood (in optimize)  2.93050765991211 likelihood (in test)  2.93037867546082
## [1] "7300 model_3: likelihood (in optimize)  2.91779851913452 likelihood (in test)  2.91767311096191
## [1] "7400 model_3: likelihood (in optimize)  2.9054582118988 likelihood (in test)  2.90533685684204"
## [1] "7500 model_3: likelihood (in optimize)  2.89348745346069 likelihood (in test)  2.89336967468262
## [1] "7600 model_3: likelihood (in optimize)  2.88188815116882 likelihood (in test)  2.88177394866943
## [1] "7700 model_3: likelihood (in optimize)  2.87066388130188 likelihood (in test)  2.87055349349976
## [1] "7800 model_3: likelihood (in optimize)  2.85981869697571 likelihood (in test)  2.85971212387085
## [1] "7900 model_3: likelihood (in optimize)  2.84935665130615 likelihood (in test)  2.84925389289856
## [1] "8000 model_3: likelihood (in optimize)  2.83928203582764 likelihood (in test)  2.83918333053589
## [1] "8100 model_3: likelihood (in optimize)  2.82959794998169 likelihood (in test)  2.82950305938721
## [1] "8200 model_3: likelihood (in optimize)  2.82030701637268 likelihood (in test)  2.82021617889404
## [1] "8300 model_3: likelihood (in optimize)  2.81141018867493 likelihood (in test)  2.81132316589355
## [1] "8400 model_3: likelihood (in optimize)  2.80290794372559 likelihood (in test)  2.80282497406006
## [1] "8500 model_3: likelihood (in optimize)  2.7947986125946 likelihood (in test)  2.79471921920776"
## [1] "8600 model_3: likelihood (in optimize)  2.78707933425903 likelihood (in test)  2.78700423240662
## [1] "8700 model_3: likelihood (in optimize)  2.77974605560303 likelihood (in test)  2.77967476844788
## [1] "8800 model_3: likelihood (in optimize)  2.77279376983643 likelihood (in test)  2.77272629737854
## [1] "8900 model_3: likelihood (in optimize)  2.76621580123901 likelihood (in test)  2.76615190505981
## [1] "9000 model_3: likelihood (in optimize)  2.7600040435791 likelihood (in test)  2.75994348526001"
## [1] "9100 model_3: likelihood (in optimize)  2.754150390625 likelihood (in test)  2.75409317016602"
## [1] "9200 model_3: likelihood (in optimize)  2.74864506721497 likelihood (in test)  2.74859142303467
## [1] "9300 model_3: likelihood (in optimize)  2.74347829818726 likelihood (in test)  2.74342823028564
## [1] "9400 model_3: likelihood (in optimize)  2.73863983154297 likelihood (in test)  2.73859310150146
## [1] "9500 model_3: likelihood (in optimize)  2.73411893844604 likelihood (in test)  2.73407506942749
## [1] "9600 model_3: likelihood (in optimize)  2.72990417480469 likelihood (in test)  2.72986364364624
## [1] "9700 model_3: likelihood (in optimize)  2.72598433494568 likelihood (in test)  2.7259464263916"
## [1] "9800 model_3: likelihood (in optimize)  2.7223482131958 likelihood (in test)  2.72231340408325"
## [1] "9900 model_3: likelihood (in optimize)  2.71898412704468 likelihood (in test)  2.7189519405365"
## [1] "10000 model_3: likelihood (in optimize)  2.71588063240051 likelihood (in test)  2.71585083007812
## [1] "10100 model_3: likelihood (in optimize)  2.71302604675293 likelihood (in test)  2.71299886703491
## [1] "10200 model_3: likelihood (in optimize)  2.71040892601013 likelihood (in test)  2.71038365364075
## [1] "10300 model_3: likelihood (in optimize)  2.70801687240601 likelihood (in test)  2.70799398422241
## [1] "10400 model_3: likelihood (in optimize)  2.70583868026733 likelihood (in test)  2.70581793785095
## [1] "10500 model_3: likelihood (in optimize)  2.70386171340942 likelihood (in test)  2.70384311676025
## [1] "10600 model_3: likelihood (in optimize)  2.70207500457764 likelihood (in test)  2.70205783843994
## [1] "10700 model_3: likelihood (in optimize)  2.70046544075012 likelihood (in test)  2.70045018196106
## [1] "10800 model_3: likelihood (in optimize)  2.69902181625366 likelihood (in test)  2.69900798797607
## [1] "10900 model_3: likelihood (in optimize)  2.69773149490356 likelihood (in test)  2.69771909713745
## [1] "11000 model_3: likelihood (in optimize)  2.69658279418945 likelihood (in test)  2.69657182693481
## [1] "11100 model_3: likelihood (in optimize)  2.69556355476379 likelihood (in test)  2.69555425643921
## [1] "11200 model_3: likelihood (in optimize)  2.69466280937195 likelihood (in test)  2.69465446472168
## [1] "11300 model_3: likelihood (in optimize)  2.69386887550354 likelihood (in test)  2.69386148452759
## [1] "11400 model_3: likelihood (in optimize)  2.69317102432251 likelihood (in test)  2.69316434860229
## [1] "11500 model_3: likelihood (in optimize)  2.69255828857422 likelihood (in test)  2.69255280494690
## [1] "11600 model_3: likelihood (in optimize)  2.69202136993408 likelihood (in test)  2.69201660156250
## [1] "11700 model_3: likelihood (in optimize)  2.69155073165894 likelihood (in test)  2.69154644012451
## [1] "11800 model_3: likelihood (in optimize)  2.69113779067993 likelihood (in test)  2.69113397598267
## [1] "11900 model_3: likelihood (in optimize)  2.69077444076538 likelihood (in test)  2.69077110290527
## [1] "12000 model_3: likelihood (in optimize)  2.69045376777649 likelihood (in test)  2.69045114517212
## [1] "12100 model_3: likelihood (in optimize)  2.69016981124878 likelihood (in test)  2.69016695022583
## [1] "12200 model_3: likelihood (in optimize)  2.68991613388062 likelihood (in test)  2.68991374969482
## [1] "12300 model_3: likelihood (in optimize)  2.68968820571899 likelihood (in test)  2.68968629837036
## [1] "12400 model_3: likelihood (in optimize)  2.68948245048523 likelihood (in test)  2.68948030471802
## [1] "12500 model_3: likelihood (in optimize)  2.68929481506348 likelihood (in test)  2.68929266929626
```

```
## [1] "12600 model_3: likelihood (in optimize)  2.68912220001221 likelihood (in test)  2.6891202926635
## [1] "12700 model_3: likelihood (in optimize)  2.68896198272705 likelihood (in test)  2.6889605522155
## [1] "12800 model_3: likelihood (in optimize)  2.68881320953369 likelihood (in test)  2.6888117790222
## [1] "12900 model_3: likelihood (in optimize)  2.68867325782776 likelihood (in test)  2.6886720657348
## [1] "13000 model_3: likelihood (in optimize)  2.68854141235352 likelihood (in test)  2.6885402202606
## [1] "13100 model_3: likelihood (in optimize)  2.68841624259949 likelihood (in test)  2.6884150505065
## [1] "13200 model_3: likelihood (in optimize)  2.68829727172852 likelihood (in test)  2.6882958412170
## [1] "13300 model_3: likelihood (in optimize)  2.68818306922913 likelihood (in test)  2.6881818771362
## [1] "13400 model_3: likelihood (in optimize)  2.6880738735199 likelihood (in test)  2.68807291984558
## [1] "13500 model_3: likelihood (in optimize)  2.68796873092651 likelihood (in test)  2.6879677772522
## [1] "13600 model_3: likelihood (in optimize)  2.68786764144897 likelihood (in test)  2.6878662109375
## [1] "13700 model_3: likelihood (in optimize)  2.68776917457581 likelihood (in test)  2.6877682209014
## [1] "13800 model_3: likelihood (in optimize)  2.6876745223999 likelihood (in test)  2.68767356872559
## [1] "13900 model_3: likelihood (in optimize)  2.68758249282837 likelihood (in test)  2.6875815391540
## [1] "14000 model_3: likelihood (in optimize)  2.68749284744263 likelihood (in test)  2.6874921321868
## [1] "14100 model_3: likelihood (in optimize)  2.68740582466125 likelihood (in test)  2.6874051094055
## [1] "14200 model_3: likelihood (in optimize)  2.68732118606567 likelihood (in test)  2.6873202323913
## [1] "14300 model_3: likelihood (in optimize)  2.6872386932373 likelihood (in test)  2.68723773956299
## [1] "14400 model_3: likelihood (in optimize)  2.68715786933899 likelihood (in test)  2.6871571540832
## [1] "14500 model_3: likelihood (in optimize)  2.68707942962646 likelihood (in test)  2.6870787143707
## [1] "14600 model_3: likelihood (in optimize)  2.68700265884399 likelihood (in test)  2.6870017051696
## [1] "14700 model_3: likelihood (in optimize)  2.68692779541016 likelihood (in test)  2.6869268417358
## [1] "14800 model_3: likelihood (in optimize)  2.68685436248779 likelihood (in test)  2.6868538856506
## [1] "14900 model_3: likelihood (in optimize)  2.68678307533264 likelihood (in test)  2.6867823600769
## [1] "15000 model_3: likelihood (in optimize)  2.68671369552612 likelihood (in test)  2.6867129802703
## [1] "15100 model_3: likelihood (in optimize)  2.68664574623108 likelihood (in test)  2.6866450309753
## [1] "15200 model_3: likelihood (in optimize)  2.68657970428467 likelihood (in test)  2.6865789890289
## [1] "15300 model_3: likelihood (in optimize)  2.68651533126831 likelihood (in test)  2.6865143775939
## [1] "15400 model_3: likelihood (in optimize)  2.68645286560059 likelihood (in test)  2.6864521503448
## [1] "15500 model_3: likelihood (in optimize)  2.68639183044434 likelihood (in test)  2.6863913536071
## [1] "15600 model_3: likelihood (in optimize)  2.6863329410553 likelihood (in test)  2.68633246421814
## [1] "15700 model_3: likelihood (in optimize)  2.68627595901489 likelihood (in test)  2.6862750053405
## [1] "15800 model_3: likelihood (in optimize)  2.68622040748596 likelihood (in test)  2.6862201690673
## [1] "15900 model_3: likelihood (in optimize)  2.68616724014282 likelihood (in test)  2.6861665248870
## [1] "16000 model_3: likelihood (in optimize)  2.68611574172974 likelihood (in test)  2.6861152648925
## [1] "16100 model_3: likelihood (in optimize)  2.68606567382812 likelihood (in test)  2.6860651969909
## [1] "16200 model_3: likelihood (in optimize)  2.6860179901123 likelihood (in test)  2.68601751327515
## [1] "16300 model_3: likelihood (in optimize)  2.68597173690796 likelihood (in test)  2.6859712600708
## [1] "16400 model_3: likelihood (in optimize)  2.68592739105225 likelihood (in test)  2.6859269142150
## [1] "16500 model_3: likelihood (in optimize)  2.68588495254517 likelihood (in test)  2.6858844757080
## [1] "16600 model_3: likelihood (in optimize)  2.68584418296814 likelihood (in test)  2.6858439445495
## [1] "16700 model_3: likelihood (in optimize)  2.68580508232117 likelihood (in test)  2.6858048439025
## [1] "16800 model_3: likelihood (in optimize)  2.68576765060425 likelihood (in test)  2.6857674121856
## [1] "16900 model_3: likelihood (in optimize)  2.68573188781738 likelihood (in test)  2.6857316493988
## [1] "17000 model_3: likelihood (in optimize)  2.68569803237915 likelihood (in test)  2.6856975555419
## [1] "17100 model_3: likelihood (in optimize)  2.68566513061523 likelihood (in test)  2.6856651306152
## [1] "17200 model_3: likelihood (in optimize)  2.68563389778137 likelihood (in test)  2.6856336593627
## [1] "17300 model_3: likelihood (in optimize)  2.68560409545898 likelihood (in test)  2.6856038570404
## [1] "17400 model_3: likelihood (in optimize)  2.68557596206665 likelihood (in test)  2.6855754852294
## [1] "17500 model_3: likelihood (in optimize)  2.68554902076721 likelihood (in test)  2.6855487823486
## [1] "17600 model_3: likelihood (in optimize)  2.68552303314209 likelihood (in test)  2.6855225563049
## [1] "17700 model_3: likelihood (in optimize)  2.68549847602844 likelihood (in test)  2.6854982376098
## [1] "17800 model_3: likelihood (in optimize)  2.68547511100769 likelihood (in test)  2.6854748725891
## [1] "17900 model_3: likelihood (in optimize)  2.68545269966125 likelihood (in test)  2.6854524612426
```

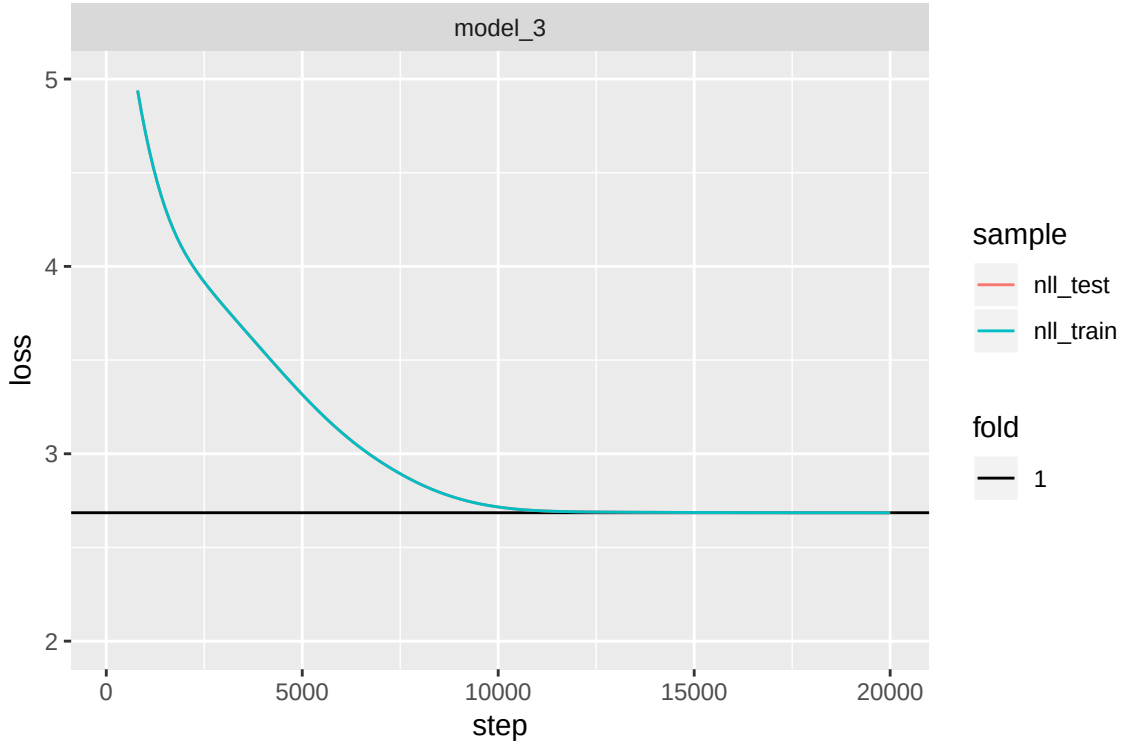
```
## [1] "18000 model_3: likelihood (in optimize)  2.68543148040771 likelihood (in test)  2.6854310035705
## [1] "18100 model_3: likelihood (in optimize)  2.68541121482849 likelihood (in test)  2.6854109764099
## [1] "18200 model_3: likelihood (in optimize)  2.68539190292358 likelihood (in test)  2.6853914260864
## [1] "18300 model_3: likelihood (in optimize)  2.68537330627441 likelihood (in test)  2.6853733062744
## [1] "18400 model_3: likelihood (in optimize)  2.68535566329956 likelihood (in test)  2.6853556632995
## [1] "18500 model_3: likelihood (in optimize)  2.68533897399902 likelihood (in test)  2.6853389739990
## [1] "18600 model_3: likelihood (in optimize)  2.68532299995422 likelihood (in test)  2.6853227615356
## [1] "18700 model_3: likelihood (in optimize)  2.68530774116516 likelihood (in test)  2.6853075027465
## [1] "18800 model_3: likelihood (in optimize)  2.68529319763184 likelihood (in test)  2.6852931976318
## [1] "18900 model_3: likelihood (in optimize)  2.68527936935425 likelihood (in test)  2.6852793693542
## [1] "19000 model_3: likelihood (in optimize)  2.68526601791382 likelihood (in test)  2.6852660179138
## [1] "19100 model_3: likelihood (in optimize)  2.68525362014771 likelihood (in test)  2.6852533817291
## [1] "19200 model_3: likelihood (in optimize)  2.68524122238159 likelihood (in test)  2.6852412223815
## [1] "19300 model_3: likelihood (in optimize)  2.68522977828979 likelihood (in test)  2.6852297782897
## [1] "19400 model_3: likelihood (in optimize)  2.68521881103516 likelihood (in test)  2.6852185726165
## [1] "19500 model_3: likelihood (in optimize)  2.6852080821991 likelihood (in test)  2.6852080821991"
## [1] "19600 model_3: likelihood (in optimize)  2.68519806861877 likelihood (in test)  2.6851980686187
## [1] "19700 model_3: likelihood (in optimize)  2.68518853187561 likelihood (in test)  2.6851882934570
## [1] "19800 model_3: likelihood (in optimize)  2.68517923355103 likelihood (in test)  2.6851789951324
## [1] "19900 model_3: likelihood (in optimize)  2.68517017364502 likelihood (in test)  2.6851701736450
## [1] "20000 model_3: likelihood (in optimize)  2.68516159057617 likelihood (in test)  2.6851618289947
```

```
history$step = as.integer(history$step)
history$fold = as.integer(history$fold)
history$nll_train = as.numeric(history$nll_train)
history$nll_test = as.numeric(history$nll_test)
history$OK = NULL# = as.numeric(history$OK)

library(tidyr)
h = gather(history, 'sample', 'loss', nll_train:nll_test)
h$loss = as.numeric(h$loss)
h$sample = as.factor(h$sample)
h$fold = as.factor(h$fold)
hh =h[!is.na(h$loss),]

ggplot(hh, aes(x=step,y=loss, color=sample, linetype=fold)) +
ylim(2,5) + geom_hline(yintercept=NLL_MLT)+  geom_line() + facet_grid(. ~ method)
```

```
## Warning: Removed 14 rows containing missing values (geom_path).
```




```
beta_nn = model_3$model_beta$get_weights()
beta_nn
```

```
## [[1]]
##               [,1]
##  [1,] -0.38338089
##  [2,]  0.15528989
##  [3,]  0.07052130
##  [4,]  0.14730200
##  [5,] -0.54899490
##  [6,]  0.33143666
##  [7,] -0.06111212
##  [8,] -0.62621558
##  [9,]  0.70007944
## [10,] -0.59370339
## [11,] -0.49053204
## [12,]  0.23254244
## [13,] -1.14878154
```

```
mlt_fit$coef[10:22]
```

```
##         crim           zn        indus         chas          nox
##  0.044629181 -0.006412397 -0.010904403 -0.583180810  4.724994838
##           rm          age          dis          rad          tax
## -0.467408980  0.002101098  0.293249835 -0.079827518  0.003507471
##      ptratio            b        lstat
##  0.225942649 -0.002572815  0.161517331
```

```
  one = tf$ones(shape = c(1,1))
  to_theta(model_3$model_hy(one))
```

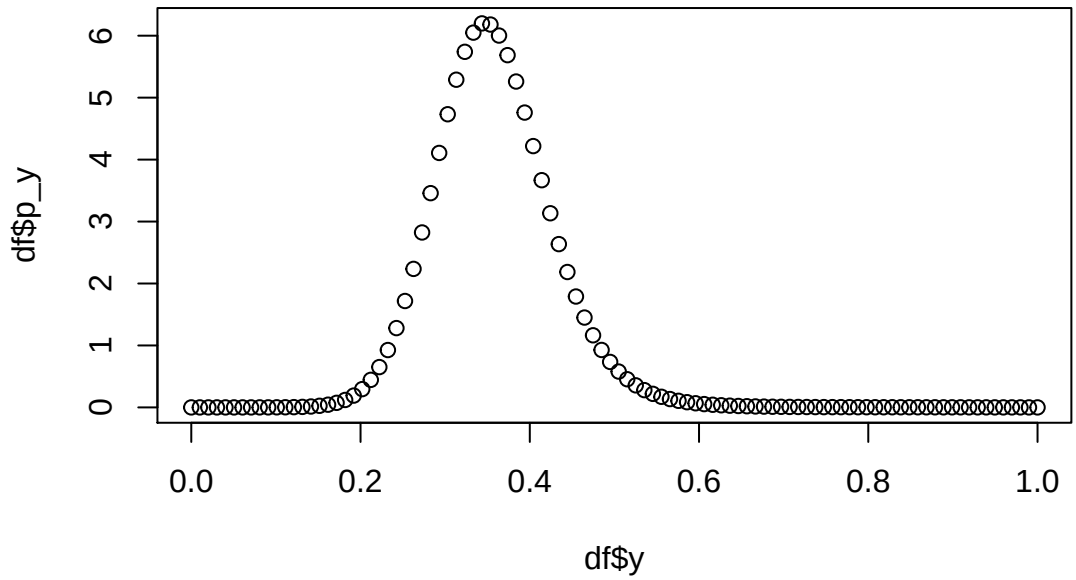
```
## tf.Tensor(
```

```
## [[-6.1408763 -3.038928  -2.9112551  1.3451097  3.1466594  3.268198
##    3.270366   3.2710438  4.5800066]], shape=(1, 9), dtype=float32)
```

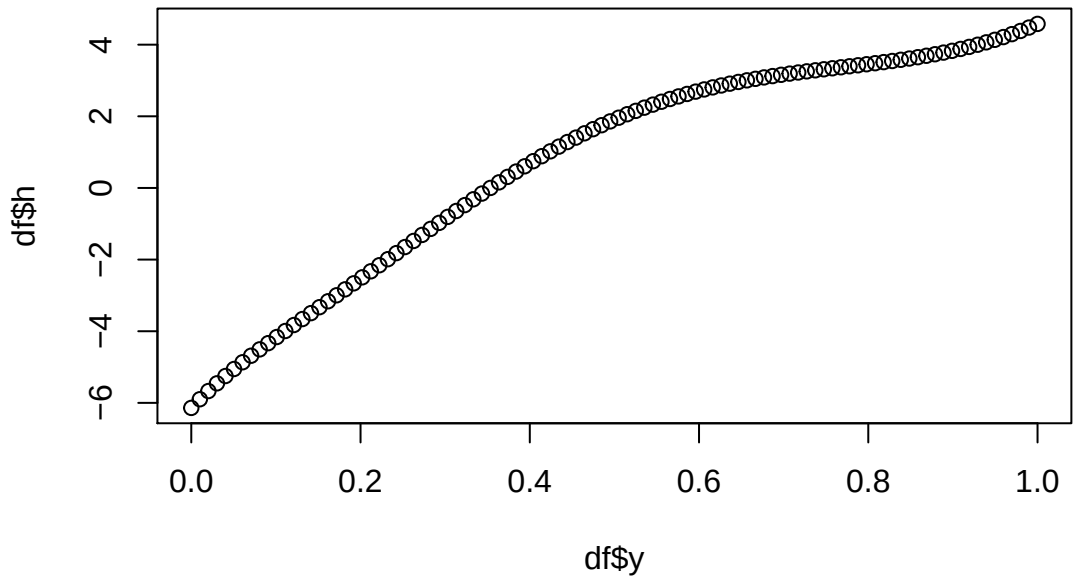
```
  mlt_fit$coef[1:9]
```

```
## Bs1(y_scale) Bs2(y_scale) Bs3(y_scale) Bs4(y_scale) Bs5(y_scale)
##   -12.949148    -9.935968    -9.926924    -4.953969    -4.099484
## Bs6(y_scale) Bs7(y_scale) Bs8(y_scale) Bs9(y_scale)
##    -3.552333    -3.543250    -3.539083    -2.253323
```

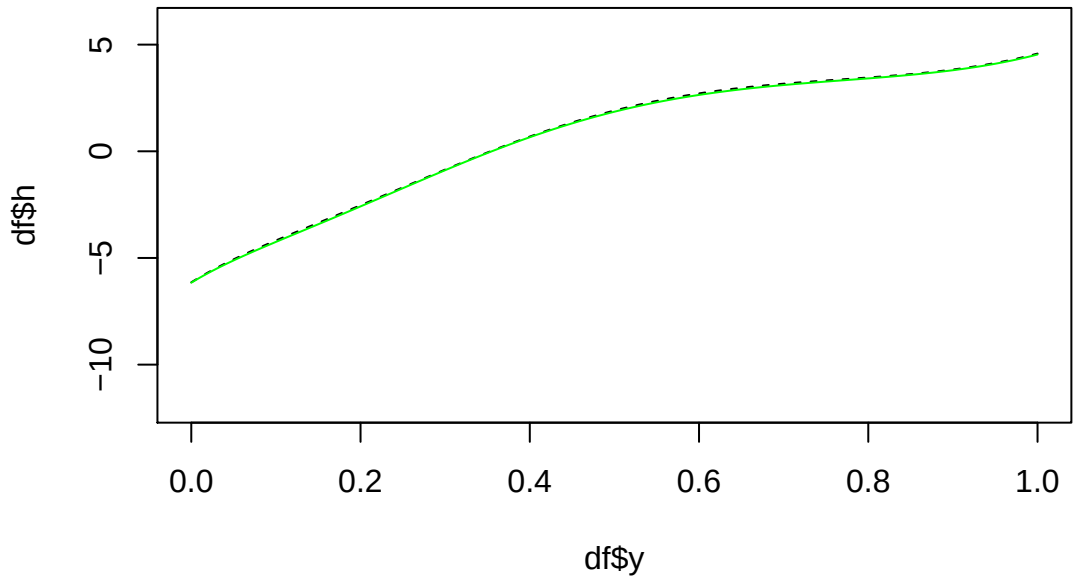
```
out_row = model_3$model_hy(one) #Pick row and compute CPD
df = bernp.p_y_h(model_3$bernp, out_row, from = 0, to = 1, length.out = 100)
plot(df$y, df$p_y)
```



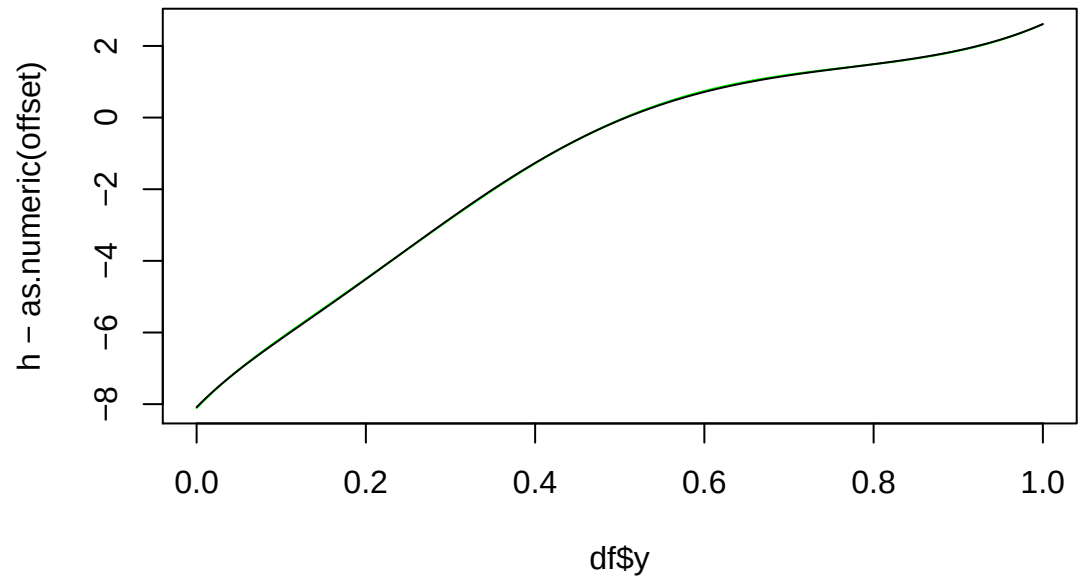

```
plot(df$y, df$h)
```




```
ddf = predict(bb, newdata = y_grid, coef = mlt_fit$coef[1:9], type='trafo')
plot(df$y, df$h, ylim=c(-12,6), type='l', lty=2)
lines(y_grid$y_scale, ddf+6.79, type='l',col='green')
```



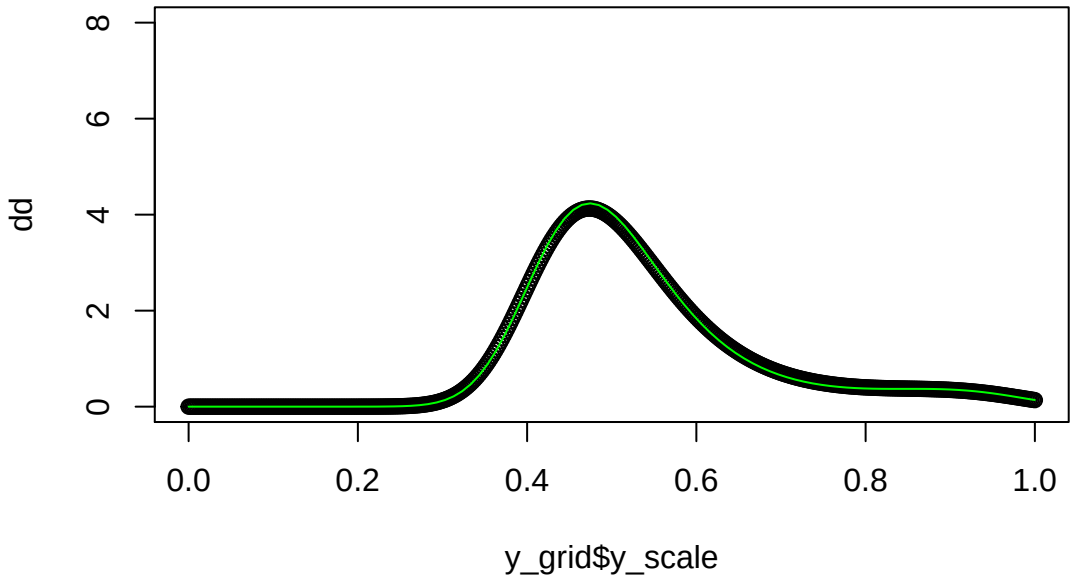

```
out_row = model_3$model_hy(one) #Pick row and compute CPD
df = bernp.p_y_h(model_3$bernp, out_row, from = 0, to = 1, length.out = 100)
h = df$h
offset = as.numeric(beta_nn[[1]]) %*% x[1,1:13]$numpy()
plot(df$y, h - as.numeric(offset), type='l', col='green')
dd = predict(mlt_fit, newdata = dat[1,], q=y_grid$y_scale, type='trafo')
lines(y_grid$y_scale, dd)
```




```
out_row = model_3$model_hy(one) #Pick row and compute CPD
offset = as.numeric(beta_nn[[1]]) %*% x[1,1:13]$numpy()
df = bernp.p_y_h(model_3$bernp, out_row, from = 0, to = 1, length.out = 100, out_eta = offset)
h = df$p_y

dd = predict(mlt_fit, newdata = dat[1,], q=y_grid$y_scale, type='density')
plot(y_grid$y_scale, dd, ylim=c(0,8))
```

```
lines(df$y, h , type='l', col='green')
```




```
sum(h)/length(h)
```

```
## [1] 0.9861924
```

```
sum(dd)/length(dd)
```

```
## [1] 0.9936181
```