Extract from Lecture Notebook

Georg Heinze

21 Feb 2020

Table of Contents

[4. Predictive modeling with the logistic regression model 1](#_Toc33189653)

[Predicting early deaths in patients with traumatic bleeding 1](#_Toc33189654)

[Download the data 1](#_Toc33189655)

[Intended statistical analysis 2](#_Toc33189656)

[Split into training and test data set 3](#_Toc33189657)

[Initial data analysis 4](#_Toc33189658)

[Exercise 4.1 24](#_Toc33189659)

[Modeling strategy 24](#_Toc33189660)

[Multivariable modeling 24](#_Toc33189661)

[Validation 29](#_Toc33189662)

[Discrimination 30](#_Toc33189663)

[Calibration 32](#_Toc33189664)

[Exercises 4.2 36](#_Toc33189665)

# 4. Predictive modeling with the logistic regression model

## Predicting early deaths in patients with traumatic bleeding

CRASH-2 was a large randomised placebo controlled trial among trauma patients with, or at risk of, significant haemorrhage, of the effects of antifibrinolytic treatment on death and transfusion requirement. The study is described at [the original trial website](http://crash2.lshtm.ac.uk/). A public version of the data set is found at a [repository of public data sets](http://biostat.mc.vanderbilt.edu/wiki/Main/DataSets) hosted by the Vanderbilt University’s Department of Biostatistics (Prof. Frank Harrell Jr.).

### Download the data

The data set can be downloaded directly as shown below:

crash2.url<-url("http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/crash2.rda")  
load(crash2.url)  
dim(crash2)

## [1] 20207 44

head(crash2)

## entryid source trandomised outcomeid sex age injurytime  
## 1 1 electronic CRF by email 2005-06-11 2 male 50 1  
## 2 2 electronic CRF by email 2005-05-29 190 female 27 1  
## 3 3 electronic CRF by email 2005-05-25 4 male 30 1  
## 4 4 electronic CRF by email 2005-05-26 5 male 40 2  
## 5 5 electronic CRF by email 2005-05-29 1154 female 23 4  
## 6 6 electronic CRF by email 2005-05-30 6 male 19 3  
## injurytype sbp rr cc hr gcseye gcsmotor gcsverbal gcs ddeath  
## 1 blunt 75 28 5 120 1 1 1 3 2005-06-14  
## 2 blunt 100 20 2 80 4 6 5 15 <NA>  
## 3 blunt 70 26 6 130 4 6 4 14 <NA>  
## 4 penetrating 60 20 5 120 2 5 3 10 2005-05-28  
## 5 penetrating 80 22 3 100 3 6 5 14 <NA>  
## 6 penetrating 90 30 5 90 4 6 5 15 2005-06-05  
## cause scauseother status ddischarge condition ndaysicu  
## 1 head injury <NA> <NA> <NA> 0  
## 2 <NA> discharged 2005-06-24 minor symptoms 0  
## 3 <NA> discharged 2005-06-08 no symptoms 6  
## 4 multi organ failure <NA> <NA> <NA> 2  
## 5 <NA> discharged 2005-06-24 no symptoms 0  
## 6 other PNEUMONIA <NA> <NA> <NA> 9  
## bheadinj bneuro bchest babdomen bpelvis bpe bdvt bstroke bbleed bmi bgi  
## 1 1 1 0 0 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0 0 0 0 0  
## 3 0 0 1 1 0 0 0 0 0 0 0  
## 4 0 0 1 1 0 0 0 0 0 0 0  
## 5 0 0 0 1 0 0 0 0 0 0 0  
## 6 0 0 1 0 0 0 0 0 0 0 0  
## bloading bmaint btransf ncell nplasma nplatelets ncryo bvii boxid packnum  
## 1 1 1 1 1 0 0 0 0 2001 25  
## 2 1 1 0 NA NA NA NA 0 2001 28  
## 3 1 1 1 2 0 0 0 0 2011 21  
## 4 1 1 1 4 0 0 0 0 2011 22  
## 5 1 1 0 NA NA NA NA 0 2011 23  
## 6 1 1 1 2 0 0 0 0 2011 24

### Intended statistical analysis

Suppose we are interested in fitting a logistic regression with the purpose of predicting early death in patients with traumatic bleeding. The following variables were deemed important to predict early death:

* Age (age, years)
* Sex (sex, male or female)
* Systolic blood pressure (sbp, mmHg)
* Heart rate (hr, 1/min)
* Respiratory rate (rr, 1/min)
* Glasgow coma score (gcs, points)
* Central capillary refill time (cc, seconds)
* Time since injury (injurytime, hours)
* Type of injury (injurytype, ‘blunt’, ‘penetrating’ or ‘blunt and penetrating’)

The outcome variable, early death (i.e., death within 28 days from injury) must be computed from the time span between date of death and date of randomisation using the following code:

# transform ddeath and trandomisation into an interpretable date format and then compute the difference  
# interpret NAs as 'not died within study period, at least not within 28 days'  
# if patients died after 28 days, treat as alive   
crash2$time2death <- as.numeric(as.Date(crash2$ddeath) - as.Date(crash2$trandomised))  
  
crash2$earlydeath[!is.na(crash2$time2death)] <- (crash2$time2death[!is.na(crash2$time2death)] <= 28) + 0   
# +0 to transform it from TRUE/FALSE to 1/0  
  
crash2$earlydeath[is.na(crash2$time2death)] <- 0 # NA in time2death means alive at day 28  
  
table(crash2$earlydeath)

##   
## 0 1   
## 17131 3076

mean(crash2$earlydeath)

## [1] 0.1522245

The binary outcome variable is now contained in variable earlydeath; 1 if died within 28 days, 0 otherwise

We learn from the table above that there were 3076 early deaths (15.2%).

Other variables that were used in the prediction model reported by [Perel et al, BMJ 2012](https://doi.org/10.1136/bmj.e5166) (economic region, treatment allocation) are not contained in the public use version of the data set and cannot be considered here.

### Split into training and test data set

As our task is prediction modeling, we would like to test our final model using a part of the data which we put aside. That part will be treated just like an independent test cohort and not used for modeling. We will use approximately 25% (approx. 5000 observations) of the data with the most recent date of randomization for testing. This allows for a ‘temporal external validation’.

n.training <- nrow(crash2) - 5000  
day\_test <- sort(as.Date(crash2$trandomised))[n.training]  
day\_test # all patients randomised after day\_test will be included in the test set, those randomised up to that day to the training set

## [1] "2009-03-01"

index.training <- as.Date(crash2$trandomised) < day\_test  
index.test <- as.Date(crash2$trandomised) >= day\_test  
  
sum(index.training)

## [1] 15188

sum(index.test)

## [1] 5019

Vectors index.training is TRUE for the first 15188 randomisations, and FALSE afterwards. Therefore, we can extract the training set by subsetting with index.training, and the test set by subsetting with index.test. We also reduce the size of the data sets by only including the relevant variables for modeling and for description.

rel.var <- c("entryid", "trandomised", "age", "sex", "sbp", "hr", "rr", "gcs", "cc", "injurytime","injurytype", "earlydeath")  
crash2.train <- crash2[index.training, rel.var]  
  
dim(crash2.train)

## [1] 15188 12

crash2.test <- crash2[index.test, rel.var]  
  
dim(crash2.test)

## [1] 5019 12

### Initial data analysis

library(GGally)

## Loading required package: ggplot2

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

crash2.idaplot <- ggpairs(crash2.train, columns=c("age", "sex", "sbp", "hr", "rr", "gcs", "cc", "injurytime", "injurytype"), progress=FALSE)  
crash2.idaplot

## Warning: Removed 3 rows containing non-finite values (stat\_density).

## Warning: Removed 3 rows containing non-finite values (stat\_boxplot).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 256 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 128 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 156 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 23 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 532 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 13 rows containing missing values

## Warning: Removed 3 rows containing non-finite values (stat\_boxplot).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 3 rows containing non-finite values (stat\_bin).

## Warning: Removed 254 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 125 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 153 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 20 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 530 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 11 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 256 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 254 rows containing non-finite values (stat\_bin).

## Warning: Removed 254 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 341 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 362 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 258 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 750 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 264 rows containing missing values

## Warning: Removed 254 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 128 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 125 rows containing non-finite values (stat\_bin).

## Warning: Removed 341 rows containing missing values (geom\_point).

## Warning: Removed 125 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 222 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 129 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 636 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 136 rows containing missing values

## Warning: Removed 125 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 156 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 153 rows containing non-finite values (stat\_bin).

## Warning: Removed 362 rows containing missing values (geom\_point).

## Warning: Removed 222 rows containing missing values (geom\_point).

## Warning: Removed 153 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 155 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 639 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 161 rows containing missing values

## Warning: Removed 153 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 23 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 20 rows containing non-finite values (stat\_bin).

## Warning: Removed 258 rows containing missing values (geom\_point).

## Warning: Removed 129 rows containing missing values (geom\_point).

## Warning: Removed 155 rows containing missing values (geom\_point).

## Warning: Removed 20 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 531 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 31 rows containing missing values

## Warning: Removed 20 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 532 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 530 rows containing non-finite values (stat\_bin).

## Warning: Removed 750 rows containing missing values (geom\_point).

## Warning: Removed 636 rows containing missing values (geom\_point).

## Warning: Removed 639 rows containing missing values (geom\_point).

## Warning: Removed 531 rows containing missing values (geom\_point).

## Warning: Removed 530 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 538 rows containing missing values

## Warning: Removed 530 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 13 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 11 rows containing non-finite values (stat\_bin).

## Warning: Removed 264 rows containing missing values (geom\_point).

## Warning: Removed 136 rows containing missing values (geom\_point).

## Warning: Removed 161 rows containing missing values (geom\_point).

## Warning: Removed 31 rows containing missing values (geom\_point).

## Warning: Removed 538 rows containing missing values (geom\_point).

## Warning: Removed 11 rows containing non-finite values (stat\_density).

## Warning: Removed 11 rows containing non-finite values (stat\_boxplot).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 3 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 254 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 125 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 153 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

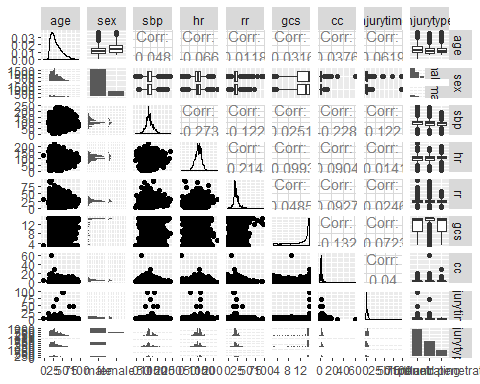
## Warning: Removed 20 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 530 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 11 rows containing non-finite values (stat\_bin).



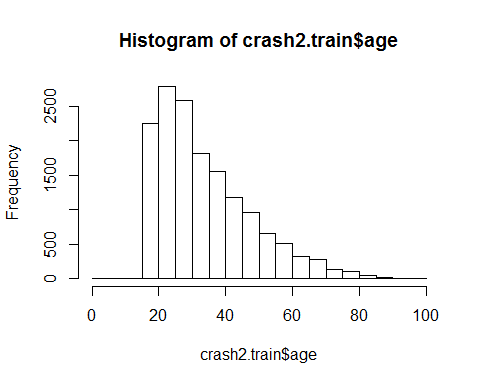
summary(crash2)

## entryid source   
## Min. : 1 telephone : 86   
## 1st Qu.: 5060 telephone entered manually : 6   
## Median :10130 electronic CRF by email :13767   
## Mean :10132 paper CRF enteredd in electronic CRF: 1711   
## 3rd Qu.:15204 electronic CRF : 4637   
## Max. :20270   
##   
## trandomised outcomeid sex age   
## Min. :2005-05-19 Min. : 1 male :16935 Min. : 1.00   
## 1st Qu.:2007-06-10 1st Qu.: 5062 female: 3271 1st Qu.:24.00   
## Median :2008-06-06 Median :10116 NA's : 1 Median :30.00   
## Mean :2008-04-01 Mean :10112 Mean :34.56   
## 3rd Qu.:2009-02-26 3rd Qu.:15164 3rd Qu.:43.00   
## Max. :2010-01-23 Max. :20199 Max. :99.00   
## NA's :80 NA's :4   
## injurytime injurytype sbp   
## Min. : 0.100 blunt :11189 Min. : 4.00   
## 1st Qu.: 1.000 penetrating : 6552 1st Qu.: 80.00   
## Median : 2.000 blunt and penetrating: 2466 Median : 95.00   
## Mean : 2.844 Mean : 98.45   
## 3rd Qu.: 4.000 3rd Qu.:110.00   
## Max. :96.000 Max. :250.00   
## NA's :11 NA's :320   
## rr cc hr gcseye   
## Min. : 1.00 Min. : 1.000 Min. : 3.0 Min. :0.000   
## 1st Qu.:20.00 1st Qu.: 2.000 1st Qu.: 90.0 1st Qu.:2.000   
## Median :22.00 Median : 3.000 Median :105.0 Median :4.000   
## Mean :23.06 Mean : 3.267 Mean :104.5 Mean :2.948   
## 3rd Qu.:26.00 3rd Qu.: 4.000 3rd Qu.:120.0 3rd Qu.:4.000   
## Max. :96.00 Max. :60.000 Max. :220.0 Max. :4.000   
## NA's :191 NA's :611 NA's :137 NA's :732   
## gcsmotor gcsverbal gcs ddeath   
## Min. :0.000 Min. :0.000 Min. : 3.00 Min. :2005-05-28   
## 1st Qu.:5.000 1st Qu.:2.000 1st Qu.:11.00 1st Qu.:2007-05-01   
## Median :6.000 Median :5.000 Median :15.00 Median :2008-05-03   
## Mean :4.795 Mean :3.611 Mean :12.47 Mean :2008-03-08   
## 3rd Qu.:6.000 3rd Qu.:5.000 3rd Qu.:15.00 3rd Qu.:2009-02-18   
## Max. :6.000 Max. :5.000 Max. :15.00 Max. :2010-01-24   
## NA's :732 NA's :735 NA's :23 NA's :17121   
## cause scauseother   
## head injury : 1225 :19924   
## bleeding : 1064 Missing data : 16   
## multi organ failure: 447 ARDS : 7   
## other : 271 SEPTICAEMIA : 4   
## pulmonary embolism : 40 bleeding & head injury : 3   
## (Other) : 42 Bronconeumonia Nosocomial: 3   
## NA's :17118 (Other) : 250   
## status ddischarge   
## discharged :13681 Min. :2005-05-26   
## still in hospital : 1969 1st Qu.:2007-06-30   
## transferred to other hospital: 1388 Median :2008-06-25   
## NA's : 3169 Mean :2008-04-18   
## 3rd Qu.:2009-03-12   
## Max. :2010-02-10   
## NA's :3185   
## condition ndaysicu   
## no symptoms :2817 Min. : 0.000   
## minor symptoms :6115 1st Qu.: 0.000   
## some restriction in lifestyle but independent :4085 Median : 0.000   
## dependent, but not requiring constant attention :2567 Mean : 2.662   
## fully dependent, requiring attention day and night:1372 3rd Qu.: 3.000   
## NA's :3251 Max. :58.000   
## NA's :182   
## bheadinj bneuro bchest babdomen   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.0000 Median :0.0000 Median :0.0000 Median :0.0000   
## Mean :0.3191 Mean :0.1043 Mean :0.1512 Mean :0.2505   
## 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:0.0000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## NA's :80 NA's :80 NA's :80 NA's :80   
## bpelvis bpe bdvt bstroke   
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :0.00000 Median :0.00000 Median :0.00000 Median :0.00000   
## Mean :0.06613 Mean :0.00691 Mean :0.00402 Mean :0.00601   
## 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.00000   
## NA's :80 NA's :80 NA's :80 NA's :80   
## bbleed bmi bgi bloading   
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.0000   
## 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.:1.0000   
## Median :0.00000 Median :0.00000 Median :0.00000 Median :1.0000   
## Mean :0.07716 Mean :0.00432 Mean :0.01312 Mean :0.9909   
## 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:0.00000 3rd Qu.:1.0000   
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.0000   
## NA's :80 NA's :80 NA's :80 NA's :80   
## bmaint btransf ncell nplasma   
## Min. :0.0000 Min. :0.0000 Min. : 0.000 Min. : 0.00   
## 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.: 2.000 1st Qu.: 0.00   
## Median :1.0000 Median :1.0000 Median : 3.000 Median : 0.00   
## Mean :0.9423 Mean :0.5084 Mean : 3.919 Mean : 1.44   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.: 5.000 3rd Qu.: 1.00   
## Max. :1.0000 Max. :1.0000 Max. :60.000 Max. :60.00   
## NA's :80 NA's :80 NA's :9963 NA's :9964   
## nplatelets ncryo bvii boxid   
## Min. : 0.000 Min. : 0.000 Min. :0.0000 Min. :2001   
## 1st Qu.: 0.000 1st Qu.: 0.000 1st Qu.:0.0000 1st Qu.:2734   
## Median : 0.000 Median : 0.000 Median :0.0000 Median :4464   
## Mean : 0.553 Mean : 0.258 Mean :0.0023 Mean :5131   
## 3rd Qu.: 0.000 3rd Qu.: 0.000 3rd Qu.:0.0000 3rd Qu.:8226   
## Max. :87.000 Max. :61.000 Max. :1.0000 Max. :9065   
## NA's :9964 NA's :9964 NA's :374   
## packnum time2death earlydeath   
## Min. :21.00 Min. : 0.000 Min. :0.0000   
## 1st Qu.:27.00 1st Qu.: 0.000 1st Qu.:0.0000   
## Median :45.00 Median : 1.000 Median :0.0000   
## Mean :51.35 Mean : 3.623 Mean :0.1522   
## 3rd Qu.:82.00 3rd Qu.: 4.000 3rd Qu.:0.0000   
## Max. :98.00 Max. :397.000 Max. :1.0000   
## NA's :17121

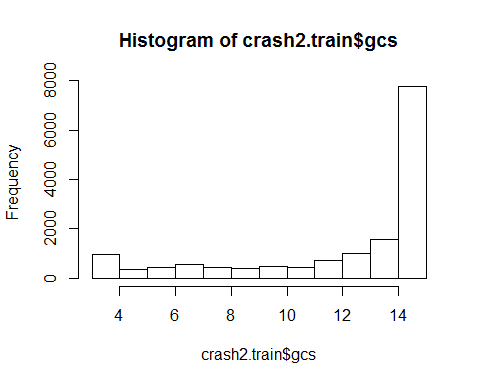
summary(crash2.train)

## entryid trandomised age sex   
## Min. : 1 Min. :2005-05-19 Min. : 1.00 male :12793   
## 1st Qu.: 3802 1st Qu.:2007-02-09 1st Qu.:24.00 female: 2395   
## Median : 7614 Median :2008-01-10 Median :30.00   
## Mean : 7693 Mean :2007-10-25 Mean :34.45   
## 3rd Qu.:11417 3rd Qu.:2008-08-08 3rd Qu.:42.00   
## Max. :20270 Max. :2009-02-28 Max. :96.00   
## NA's :3   
## sbp hr rr gcs   
## Min. : 4.00 Min. : 3 Min. : 1.00 Min. : 3.00   
## 1st Qu.: 80.00 1st Qu.: 90 1st Qu.:20.00 1st Qu.:11.00   
## Median : 90.00 Median :107 Median :22.00 Median :15.00   
## Mean : 97.91 Mean :105 Mean :23.21 Mean :12.44   
## 3rd Qu.:110.00 3rd Qu.:120 3rd Qu.:26.00 3rd Qu.:15.00   
## Max. :240.00 Max. :220 Max. :95.00 Max. :15.00   
## NA's :254 NA's :125 NA's :153 NA's :20   
## cc injurytime injurytype   
## Min. : 1.000 Min. : 0.100 blunt :8412   
## 1st Qu.: 2.000 1st Qu.: 1.000 penetrating :4792   
## Median : 3.000 Median : 2.000 blunt and penetrating:1984   
## Mean : 3.285 Mean : 2.865   
## 3rd Qu.: 4.000 3rd Qu.: 4.000   
## Max. :60.000 Max. :96.000   
## NA's :530 NA's :11   
## earlydeath   
## Min. :0.0000   
## 1st Qu.:0.0000   
## Median :0.0000   
## Mean :0.1542   
## 3rd Qu.:0.0000   
## Max. :1.0000   
##

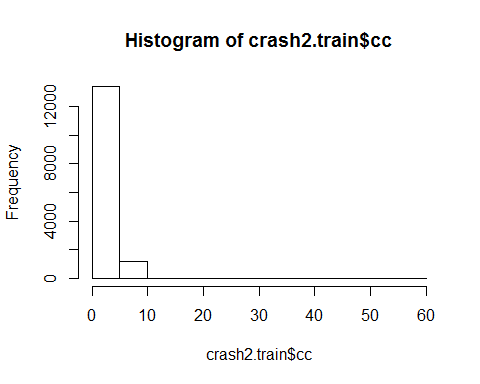
# we go deeper into histograms of some skew-looking variables  
hist(crash2.train$age)



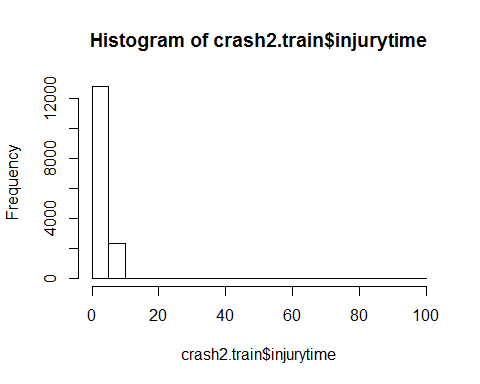
hist(crash2.train$gcs)



hist(crash2.train$cc)



hist(crash2.train$injurytime)



Note that graphs can also be printed into a pdf file, e.g. like in the following code. The pdf output is ‘opened’ with the pdf(file="...pdf") command and closed with dev.off(). Don’t forget to close it! If you do, all plots produced later will be added to the pdf but that pdf will never be finalized.

pdf(file="CRASH2 IDA.pdf")  
 crash2.idaplot

## Warning: Removed 3 rows containing non-finite values (stat\_density).

## Warning: Removed 3 rows containing non-finite values (stat\_boxplot).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 256 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 128 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 156 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 23 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 532 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 13 rows containing missing values

## Warning: Removed 3 rows containing non-finite values (stat\_boxplot).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 3 rows containing non-finite values (stat\_bin).

## Warning: Removed 254 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 125 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 153 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 20 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 530 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 11 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 256 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 254 rows containing non-finite values (stat\_bin).

## Warning: Removed 254 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 341 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 362 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 258 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 750 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 264 rows containing missing values

## Warning: Removed 254 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 128 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 125 rows containing non-finite values (stat\_bin).

## Warning: Removed 341 rows containing missing values (geom\_point).

## Warning: Removed 125 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 222 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 129 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 636 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 136 rows containing missing values

## Warning: Removed 125 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 156 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 153 rows containing non-finite values (stat\_bin).

## Warning: Removed 362 rows containing missing values (geom\_point).

## Warning: Removed 222 rows containing missing values (geom\_point).

## Warning: Removed 153 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 155 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 639 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 161 rows containing missing values

## Warning: Removed 153 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 23 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 20 rows containing non-finite values (stat\_bin).

## Warning: Removed 258 rows containing missing values (geom\_point).

## Warning: Removed 129 rows containing missing values (geom\_point).

## Warning: Removed 155 rows containing missing values (geom\_point).

## Warning: Removed 20 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 531 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 31 rows containing missing values

## Warning: Removed 20 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 532 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 530 rows containing non-finite values (stat\_bin).

## Warning: Removed 750 rows containing missing values (geom\_point).

## Warning: Removed 636 rows containing missing values (geom\_point).

## Warning: Removed 639 rows containing missing values (geom\_point).

## Warning: Removed 531 rows containing missing values (geom\_point).

## Warning: Removed 530 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 538 rows containing missing values

## Warning: Removed 530 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 13 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 11 rows containing non-finite values (stat\_bin).

## Warning: Removed 264 rows containing missing values (geom\_point).

## Warning: Removed 136 rows containing missing values (geom\_point).

## Warning: Removed 161 rows containing missing values (geom\_point).

## Warning: Removed 31 rows containing missing values (geom\_point).

## Warning: Removed 538 rows containing missing values (geom\_point).

## Warning: Removed 11 rows containing non-finite values (stat\_density).

## Warning: Removed 11 rows containing non-finite values (stat\_boxplot).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 3 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 254 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 125 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 153 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 20 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 530 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 11 rows containing non-finite values (stat\_bin).

hist(crash2.train$age)  
 hist(crash2.train$gcs)  
 hist(crash2.train$cc)  
 hist(crash2.train$injurytime)  
dev.off()

## png   
## 2

Conclusions from initial data analysis:

* there is a single patient with age = 1 year -> omit that person (by setting his/her age to NA)
* cc and injurytime have quite skew distribution. These variables could be modeled as they to obtain effects per time unit (seconds or hours, respectively). However, with very skew distributions of predictors high values can obtain disproportionally high impact on the result. I usually prefer to transform such variables first. The log-base-2 transformation (log2()) is particularly attractive as it still allows to interpret the regression coefficients ‘per log2 unit’ which essentially means ‘per doubling of the original variable’.
* There is relevant correlation between rr and hr, and between cc and sbp. (These correlations should be compared with expectations based on clinical expertise.)

As a next step, we perform the exclusion and transformations outlined above and repeat the initial data analysis with the updated data set:

# set age of patient with age==1 to NA  
crash2.train$age[crash2.train$age == 1] <- NA  
  
# create log2 of cc and injurytime  
  
crash2.train$log2\_cc <- log2(crash2.train$cc)  
crash2.train$log2\_injurytime <- log2(crash2.train$injurytime)  
  
summary(crash2.train$log2\_cc)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.000 1.000 1.585 1.544 2.000 5.907 530

summary(crash2.train$log2\_injurytime)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## -3.322 0.000 1.000 1.055 2.000 6.585 11

ggpairs(crash2.train[,c("age", "sex", "sbp", "hr", "rr", "gcs", "log2\_cc", "log2\_injurytime", "injurytype")], progress=FALSE)

## Warning: Removed 4 rows containing non-finite values (stat\_density).

## Warning: Removed 4 rows containing non-finite values (stat\_boxplot).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 256 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 129 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 157 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 24 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 533 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 14 rows containing missing values

## Warning: Removed 4 rows containing non-finite values (stat\_boxplot).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 4 rows containing non-finite values (stat\_bin).

## Warning: Removed 254 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 125 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 153 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 20 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 530 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 11 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 256 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 254 rows containing non-finite values (stat\_bin).

## Warning: Removed 254 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 341 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 362 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 258 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 750 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 264 rows containing missing values

## Warning: Removed 254 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 129 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 125 rows containing non-finite values (stat\_bin).

## Warning: Removed 341 rows containing missing values (geom\_point).

## Warning: Removed 125 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 222 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 129 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 636 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 136 rows containing missing values

## Warning: Removed 125 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 157 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 153 rows containing non-finite values (stat\_bin).

## Warning: Removed 362 rows containing missing values (geom\_point).

## Warning: Removed 222 rows containing missing values (geom\_point).

## Warning: Removed 153 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 155 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 639 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 161 rows containing missing values

## Warning: Removed 153 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 24 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 20 rows containing non-finite values (stat\_bin).

## Warning: Removed 258 rows containing missing values (geom\_point).

## Warning: Removed 129 rows containing missing values (geom\_point).

## Warning: Removed 155 rows containing missing values (geom\_point).

## Warning: Removed 20 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 531 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 31 rows containing missing values

## Warning: Removed 20 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 533 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 530 rows containing non-finite values (stat\_bin).

## Warning: Removed 750 rows containing missing values (geom\_point).

## Warning: Removed 636 rows containing missing values (geom\_point).

## Warning: Removed 639 rows containing missing values (geom\_point).

## Warning: Removed 531 rows containing missing values (geom\_point).

## Warning: Removed 530 rows containing non-finite values (stat\_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method = "pearson", :  
## Removed 538 rows containing missing values

## Warning: Removed 530 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 14 rows containing missing values (geom\_point).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 11 rows containing non-finite values (stat\_bin).

## Warning: Removed 264 rows containing missing values (geom\_point).

## Warning: Removed 136 rows containing missing values (geom\_point).

## Warning: Removed 161 rows containing missing values (geom\_point).

## Warning: Removed 31 rows containing missing values (geom\_point).

## Warning: Removed 538 rows containing missing values (geom\_point).

## Warning: Removed 11 rows containing non-finite values (stat\_density).

## Warning: Removed 11 rows containing non-finite values (stat\_boxplot).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 4 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 254 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 125 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 153 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

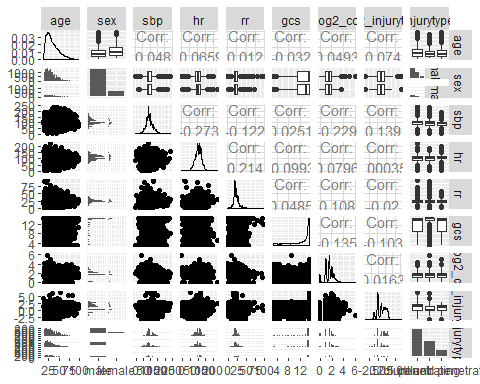
## Warning: Removed 20 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

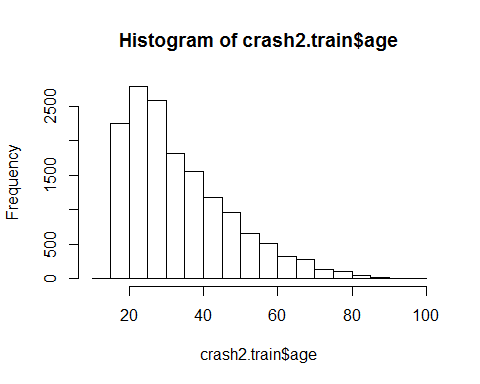
## Warning: Removed 530 rows containing non-finite values (stat\_bin).

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

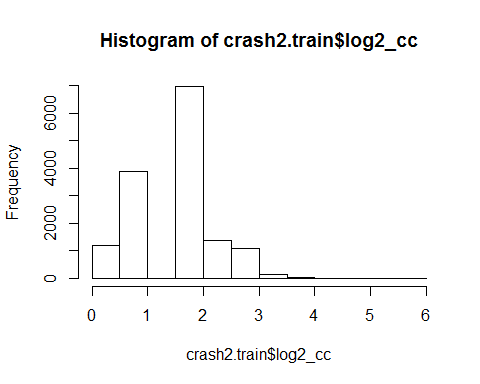
## Warning: Removed 11 rows containing non-finite values (stat\_bin).



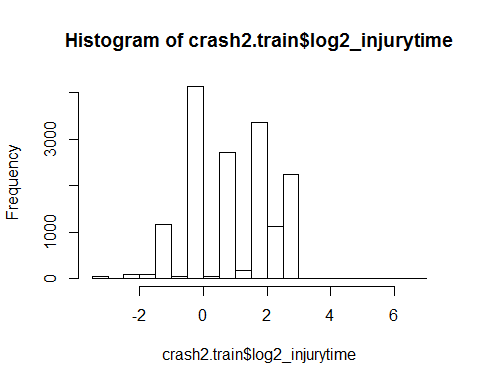
hist(crash2.train$age)



hist(crash2.train$log2\_cc)



hist(crash2.train$log2\_injurytime)

 ### Data description

Using the R package TableOne, we can create a customary data description table. We will generate columns by injury type.

library(tableone)  
  
tabone<-CreateTableOne(vars=c("age", "sex", "sbp", "hr", "rr", "gcs", "cc", "log2\_cc", "injurytime", "log2\_injurytime"), strata="injurytype", data=crash2.train, test=FALSE)  
  
print(tabone, nonnormal=c("age","sbp","hr","rr","gcs","cc","log2\_cc", "injurytime", "log2\_injurytime"))

## Stratified by injurytype  
## blunt penetrating   
## n 8412 4792   
## age (median [IQR]) 32.00 [24.00, 45.00] 29.00 [23.00, 38.00]   
## sex = female (%) 1625 (19.3) 452 (9.4)   
## sbp (median [IQR]) 100.00 [85.00, 115.00] 90.00 [80.00, 110.00]  
## hr (median [IQR]) 106.00 [90.00, 120.00] 100.00 [90.00, 120.00]  
## rr (median [IQR]) 22.00 [20.00, 26.00] 22.00 [20.00, 27.00]   
## gcs (median [IQR]) 14.00 [9.00, 15.00] 15.00 [14.00, 15.00]   
## cc (median [IQR]) 3.00 [2.00, 4.00] 3.00 [2.00, 4.00]   
## log2\_cc (median [IQR]) 1.58 [1.00, 2.00] 1.58 [1.00, 2.00]   
## injurytime (median [IQR]) 3.00 [1.00, 5.00] 1.00 [1.00, 3.00]   
## log2\_injurytime (median [IQR]) 1.58 [0.00, 2.32] 0.00 [0.00, 1.58]   
## Stratified by injurytype  
## blunt and penetrating   
## n 1984   
## age (median [IQR]) 30.00 [24.00, 40.00]   
## sex = female (%) 318 (16.0)   
## sbp (median [IQR]) 90.00 [80.00, 100.00]   
## hr (median [IQR]) 110.00 [100.00, 120.00]  
## rr (median [IQR]) 24.00 [20.00, 28.00]   
## gcs (median [IQR]) 14.00 [9.00, 15.00]   
## cc (median [IQR]) 3.00 [2.00, 4.00]   
## log2\_cc (median [IQR]) 1.58 [1.00, 2.00]   
## injurytime (median [IQR]) 2.00 [1.00, 4.00]   
## log2\_injurytime (median [IQR]) 1.00 [0.00, 2.00]

We should also evaluate the number and proportions of missing values, per variable and as total number with any missing values.

# the apply() function computes per column of a data set (second argument set to 2) or per row of a data set (1)  
# the thing we want to compute is two values combined using the c() function: the sum (count) of missing values and the proportion  
# is.na(X) is TRUE if a value of X is NA, and FALSE otherwise.  
# if we sum over TRUEs and FALSEs, we get the number of TRUE's (they are like 1's)  
# if we take the mean over TRUEs and FALSEs, we get the proportion of TRUEs (as they are like 1s)  
# the t() function transposes the result (rows<->columns)  
  
out<-t(apply(crash2.train, 2, function(X) c(sum(is.na(X)), mean(is.na(X)))))  
# here we assign column names  
colnames(out)<- c("Number NAs", "Proportion NAs")  
  
# here we apply per row the any() function which returns TRUE if any of its elements is TRUE  
total\_na\_perobs<-apply(crash2.train, 1, function(X) any(is.na(X)))  
  
# now we put the TOTAL result underneath the variable-wise missing value summary  
out<-rbind(out, TOTAL=c(sum(total\_na\_perobs), mean(total\_na\_perobs)))  
  
# output rounded to three decimal numbers  
round(out,3)

## Number NAs Proportion NAs  
## entryid 0 0.000  
## trandomised 0 0.000  
## age 4 0.000  
## sex 0 0.000  
## sbp 254 0.017  
## hr 125 0.008  
## rr 153 0.010  
## gcs 20 0.001  
## cc 530 0.035  
## injurytime 11 0.001  
## injurytype 0 0.000  
## earlydeath 0 0.000  
## log2\_cc 530 0.035  
## log2\_injurytime 11 0.001  
## TOTAL 907 0.060

By omitting all observations with any missing values in the variables, we loose only about 6% of the data. Given the large initial sample size, this is a feasible way of dealing with the missing data.

### Exercise 4.1

1. (easy) Why is number of NAs in the last line (TOTAL) greater than those of any individual variable?
2. (easy) Why is the number of NAs for injurytime the same as for log2\_injurytime?

### Modeling strategy

A multivariable logistic regression model will be estimated to predict early death from the independent variables age, sex, systolic blood pressure, heart rate, respiratory rate, Glasgow coma scale, capillary refill time, injury time and injury type.

For continuous variables, we will use natural splines with three degrees of freedom. For gcs with its semi-continuous scale, we will just use a polynomial of degree 2. It is unlikely that we observe a U-shaped or even more complex association of gcs with early death and therefore a less flexible model with fewer degrees of freedom can be used.

### Multivariable modeling

library(splines)  
fit.crash2 <- glm(data=crash2.train, formula=earlydeath ~ ns(age, df=3) + sex + ns(sbp, df=3)+ ns(hr, df=3) + ns(rr, df=3)+ poly(gcs, 2, raw=TRUE) + ns(log2\_cc, df=3)+ns(log2\_injurytime, df=3) + injurytype , family=binomial)  
  
summary(fit.crash2)

##   
## Call:  
## glm(formula = earlydeath ~ ns(age, df = 3) + sex + ns(sbp, df = 3) +   
## ns(hr, df = 3) + ns(rr, df = 3) + poly(gcs, 2, raw = TRUE) +   
## ns(log2\_cc, df = 3) + ns(log2\_injurytime, df = 3) + injurytype,   
## family = binomial, data = crash2.train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4735 -0.4693 -0.3209 -0.2315 2.9048   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.012135 0.681927 4.417 1.00e-05 \*\*\*  
## ns(age, df = 3)1 0.203072 0.144088 1.409 0.158727   
## ns(age, df = 3)2 2.044403 0.308501 6.627 3.43e-11 \*\*\*  
## ns(age, df = 3)3 2.543022 0.261913 9.709 < 2e-16 \*\*\*  
## sexfemale 0.055613 0.075160 0.740 0.459348   
## ns(sbp, df = 3)1 -2.634095 0.205545 -12.815 < 2e-16 \*\*\*  
## ns(sbp, df = 3)2 -1.735102 0.813820 -2.132 0.033003 \*   
## ns(sbp, df = 3)3 1.295564 0.492716 2.629 0.008553 \*\*   
## ns(hr, df = 3)1 -0.265951 0.201532 -1.320 0.186952   
## ns(hr, df = 3)2 -0.955861 0.915503 -1.044 0.296447   
## ns(hr, df = 3)3 1.016030 0.691303 1.470 0.141634   
## ns(rr, df = 3)1 1.757361 0.223895 7.849 4.19e-15 \*\*\*  
## ns(rr, df = 3)2 -0.478706 0.819594 -0.584 0.559169   
## ns(rr, df = 3)3 0.333245 1.143336 0.291 0.770694   
## poly(gcs, 2, raw = TRUE)1 -0.424637 0.042386 -10.018 < 2e-16 \*\*\*  
## poly(gcs, 2, raw = TRUE)2 0.008248 0.002163 3.812 0.000138 \*\*\*  
## ns(log2\_cc, df = 3)1 0.593541 0.208959 2.840 0.004505 \*\*   
## ns(log2\_cc, df = 3)2 0.529335 0.568878 0.930 0.352118   
## ns(log2\_cc, df = 3)3 -0.170713 0.918389 -0.186 0.852537   
## ns(log2\_injurytime, df = 3)1 0.374635 0.230501 1.625 0.104098   
## ns(log2\_injurytime, df = 3)2 0.927017 1.071429 0.865 0.386921   
## ns(log2\_injurytime, df = 3)3 -1.053139 0.936190 -1.125 0.260623   
## injurytypepenetrating -0.232569 0.073716 -3.155 0.001605 \*\*   
## injurytypeblunt and penetrating -0.199487 0.082394 -2.421 0.015473 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 11936.8 on 14280 degrees of freedom  
## Residual deviance: 8948.2 on 14257 degrees of freedom  
## (907 observations deleted due to missingness)  
## AIC: 8996.2  
##   
## Number of Fisher Scoring iterations: 5

To visualize the splines fitting, we can create effect plots in which the impact of a variable on the predictions of the model are plotted. The idea is here to compute the predictions from the fitted model given one variable is varied across its full range, and all other variables are held constant at their means (by default, any other values can be chosen). In order to compute effects plots, the package effects is needed. If not already available, it can be installed using install.packages("effects").

The following plots depict the role of the continuous predictors in the multivariable prediction model:

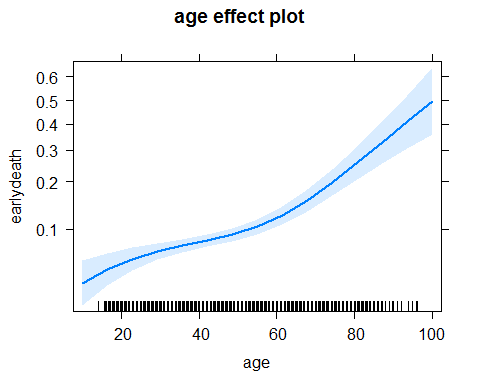
library(effects)

## Warning: package 'effects' was built under R version 3.6.2

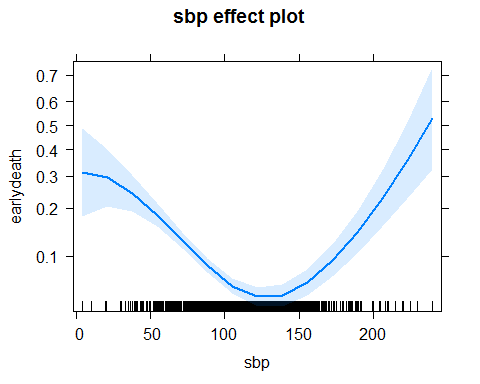
## Loading required package: carData

## lattice theme set by effectsTheme()  
## See ?effectsTheme for details.

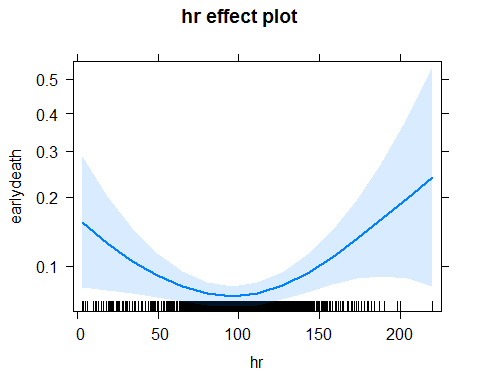
age.eff <- Effect("age", fit.crash2)  
plot(age.eff)



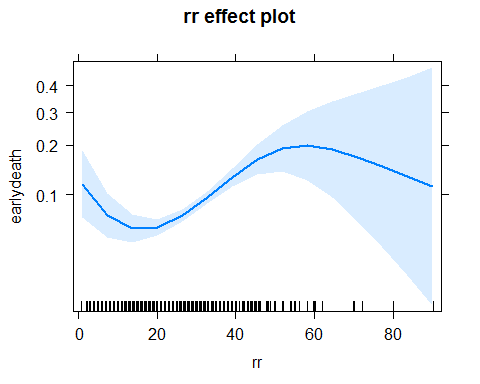
sbp.eff <- Effect("sbp", fit.crash2)  
plot(sbp.eff)



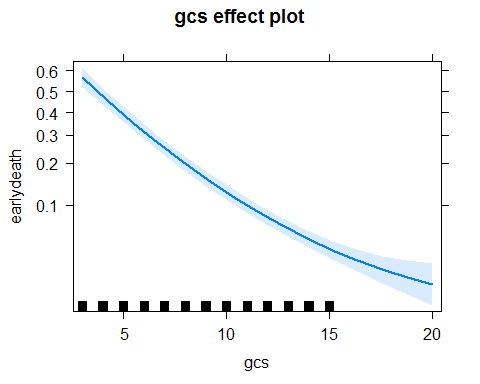
hr.eff <- Effect("hr", fit.crash2)  
plot(hr.eff)



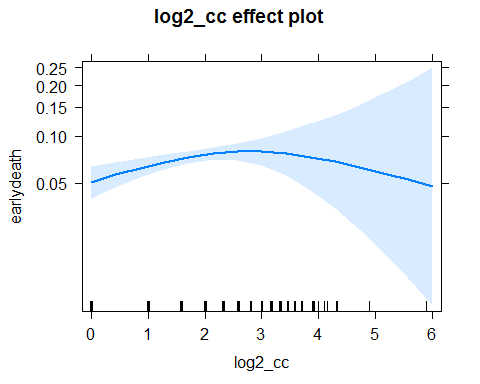
rr.eff <- Effect("rr", fit.crash2)  
plot(rr.eff)



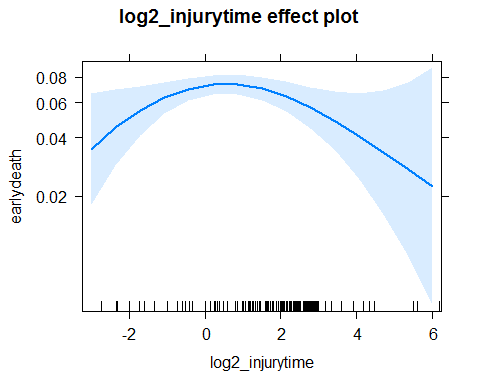
gcs.eff <- Effect("gcs", fit.crash2)  
plot(gcs.eff)



log2\_cc.eff <- Effect("log2\_cc", fit.crash2)  
plot(log2\_cc.eff)



log2\_injurytime.eff <- Effect("log2\_injurytime", fit.crash2)  
plot(log2\_injurytime.eff)



### Validation

Validation of a prediction model means: to assess its performance. The result of a validation is not ‘yes it’s valid’ or ‘no it’s not valid’, but rather a couple of performance measures that allow potential users to estimate if the model is useful enough for their purpose or not.

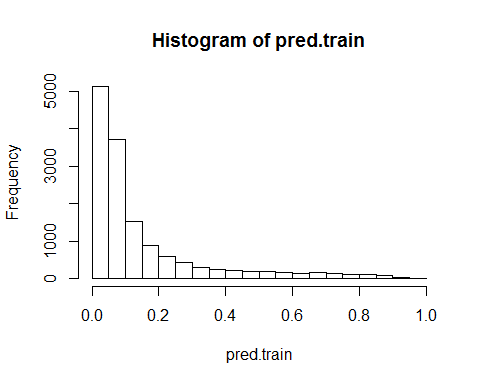
Validation can be performed internally, to see if the model was correctly specified, e.g. using residual plots in the linear model or by computing R-squared values. An external validation evaluates how the model performs if applied in a different population. External validation can be geographical (a population that is in a different geographical area), temporal (if an ‘old’ model is still performing well enough), or may show if a model can be transferred to a different setting (e.g. from inpatient to outpatient clinic).

Aspects that are usually evaluated when validating a model are its calibration, its discriminative ability, and the explained variation. These aspects are partly related, but may also complement each other; e.g. a model may be highly discriminative but poorly calibrated.

### Discrimination

We make use of the predict() function to copmute the predicted probabilites based on the model fit. That function takes care of transforming the continuous variables such that the spline fitting is directly applied to them.

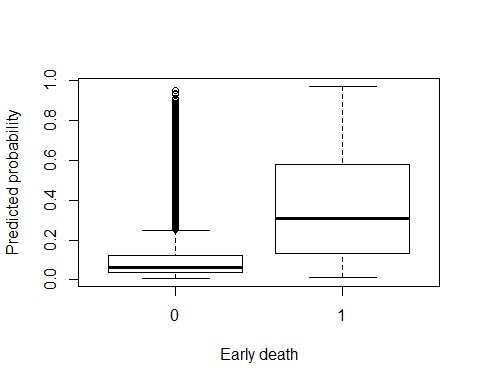
pred.train <- predict(fit.crash2, newdata=crash2.train, type="response")  
  
hist(pred.train)



The histogram clearly shows that most patients are assigned a low predicted probability, and only few receive a higher predicted probability.

By comparing the predicted probabilities between patients with event and patients without, we can see how well the model discriminates high-risk from low-risk individuals.

boxplot(pred.train~crash2.train$earlydeath, ylab="Predicted probability", xlab="Early death")



The impression of that graph that predicted probabilities are higher in those subjects who died within 28 days from injury can be summarized in two numbers:

1. We can compare the mean predicted probabilities between individuals who died and those who survived and compute their mean difference. This measure is known as ‘coefficient of discrimination’ and can be used as an analogue of R-squared in logistic regression as it ranges between 0 and 1.
2. The predicted probabilities in these two groups of subjects can also be compared nonparametrically using a Mann-Whitney-U-statistic. This number can be interpreted as , i.e., the probability that a patient who survived was assigned a lower predicted probability than a patient who survived. It is also known as concordance index, c-index or area under the ROC. The c-index has a null value of 0.5 (where the distributions of the probabilities are stochastically equal), and a maximum value of 1.

Here we ‘misuse’ a linear model to compute the coefficient of discrimination. The interpretation of the slope, i.e., the coefficient of earlydeath, is (as usual) that of an expected difference in the predicted probabilites comparing patients who died to those who survived (= the definition of the coefficient of discrimination).

# Computing the coefficient of discrimination  
lm(pred.train ~ crash2.train$earlydeath)

##   
## Call:  
## lm(formula = pred.train ~ crash2.train$earlydeath)  
##   
## Coefficients:  
## (Intercept) crash2.train$earlydeath   
## 0.1095 0.2563

The coefficient of discrimination in this example is 0.2562611, which is a good value for a logistic regression model. Ideally, this number should be 1. In the worst case, it will be 0 (meaning that the model is meaningless for the outcome).

The concordance index can be computed using the simple code below:

# Simple function to compute the c-index via the Wilcoxon statistic  
cindex<-function(x,y) {  
 if(is.factor(y)) {  
 lev<-levels(y)  
 y<-(y==lev[1])\*1  
 }  
 sy0<-sum(y==0)  
 sy1<-sum(y==1)  
 c1<-wilcox.test(x~y)$statistic/(sy0\*sy1)  
 if(sy1<sy0) c1<-1-c1  
 return(c1)  
}  
  
cindex(x=pred.train, y=crash2.train$earlydeath)

## W   
## 0.8619571

In this code the Wilcoxon test statistic, which is the sum of ranks of the values (of predicted probabilities) in one group, is divided by the product of the group sizes. This gives the Mann-Whitney statistic which is equal to the concordance index. In the statement if(sy1<sy0) c1<-1-c1 its value is ‘flipped’ if it was computed for the smaller group.

The concordance index is 0.862 and also quite good.

### Calibration

A calibration plot compares the observed outcome rates against the predicted probabilities. To construct such plots, the data are often grouped by the predicted probabilities and then the observed rate in each group is plotted against the mean predicted probability in that group.

We will carry out two calibration evaluations; one in the training set itself (to assess internal calibration), and one in the test set. In the training set we expect good calibration as the model calibrates itself. Still, calibration (e.g. agreement of observed rates and predicted probabilities) can be poor at the edges, i.e. for high-risk individuals.

More interesting is calibration in the test set. Here we compute predicted probabilities from the model, but now using the data on the independent variables of the test set, and compare them to the observed outcomes.

Using the function calibration() in the package caret we can automate the binning of observations into groups by predicted probabilities. By default, 10 bins are produced.

library(caret)

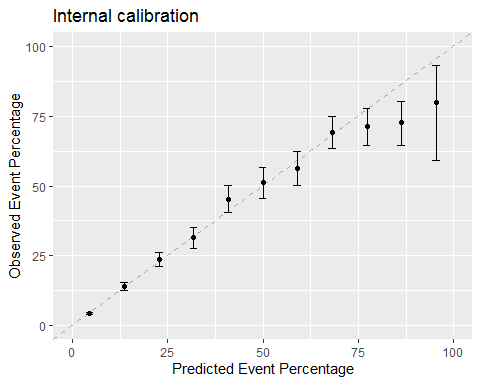
## Warning: package 'caret' was built under R version 3.6.2

## Loading required package: lattice

calib.internal<-calibration(factor(crash2.train$earlydeath, levels=c(1,0))~pred.train, cuts=11)  
ggplot(calib.internal)+ylim(c(0,100))+xlim(c(0,100))+ggtitle("Internal calibration")+xlab("Predicted Event Percentage")

## Warning: Removed 1 rows containing missing values (geom\_point).

## Warning: Removed 1 rows containing missing values (geom\_errorbar).



In the **external (temporal) validation** we apply the model to the data that was not used to fit the model.

First we have to compute log2\_cc and log2\_injurytime also in the test set for compatibility of the models. Then we compute the predicted probabilities in the test set applying the model estimated in the training step. This is actually the most important step!

crash2.test$log2\_cc <- log2(crash2.test$cc)  
crash2.test$log2\_injurytime <- log2(crash2.test$injurytime)  
  
  
  
pred.test <- predict(fit.crash2, newdata=crash2.test, type="response")

Now we can proceed by computing coefficient of discrimination and concordance index:

# coefficient of discrimination (second coefficient)  
lm(pred.test~crash2.test$earlydeath)

##   
## Call:  
## lm(formula = pred.test ~ crash2.test$earlydeath)  
##   
## Coefficients:  
## (Intercept) crash2.test$earlydeath   
## 0.1093 0.2441

# AUROC or c-index  
cindex(x=pred.test, y=crash2.test$earlydeath)

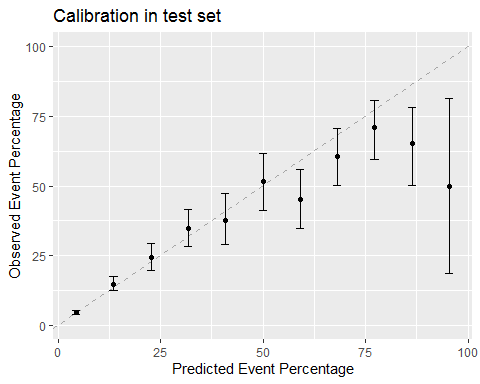
## W   
## 0.8378945

Finally we also show the calibration curve for the test set, the ultimate validation of the model.

# calibration curve  
calib<-calibration(factor(crash2.test$earlydeath, levels=c(1,0))~pred.test, cuts=11)  
ggplot(calib)+ylim(c(0,100))+ggtitle("Calibration in test set")+xlab("Predicted Event Percentage")

## Warning: Removed 1 rows containing missing values (geom\_point).

## Warning: Removed 1 rows containing missing values (geom\_errorbar).



A summary of the calibration curve is obtained by computing the so-called calibration-in-the-large and the calibration slope.

These values are obtained by computing the linear predictors in the test set and then conducting a logistic regression of the actual event status on the linear predictors. The slope (regression coefficient) of that regression is the calibration slope and ideally equal to 1.

If this is done in the training set, then the intercept will be exactly 0 and the slope exactly 1.

Usually in external validations we observe values for the slope which are slightly smaller than 1. This is called the ‘shrinkage’ effect: in an independent application of the model, it seems that the effect is smaller than it was anticipated. The reason for this can be overfitting of the effects in the training set (the model adopts too much to the data is no longer fully generalizable), or just because the effects are different in the test population.

In our study we get:

# compute linear predictors in the test set  
linpred.test <- predict(fit.crash2, newdata=crash2.test, type="link")  
  
# logistic regression on the linear predictors  
calib.mod <- glm(data=crash2.test, earlydeath ~ linpred.test, family=binomial)  
summary(calib.mod)

##   
## Call:  
## glm(formula = earlydeath ~ linpred.test, family = binomial, data = crash2.test)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.1415 -0.4772 -0.3394 -0.2579 2.7573   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.19782 0.06706 -2.95 0.00318 \*\*   
## linpred.test 0.87203 0.03307 26.37 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3973.8 on 4845 degrees of freedom  
## Residual deviance: 3112.6 on 4844 degrees of freedom  
## (173 observations deleted due to missingness)  
## AIC: 3116.6  
##   
## Number of Fisher Scoring iterations: 5

The calibration slope is indeed slightly smaller than 1, it is 0.8720338.

To evaluate the calibration-in-the-large, the easiest is to compare the observed event rate with the mean predicted probability in the test set:

mean(crash2.test$earlydeath)

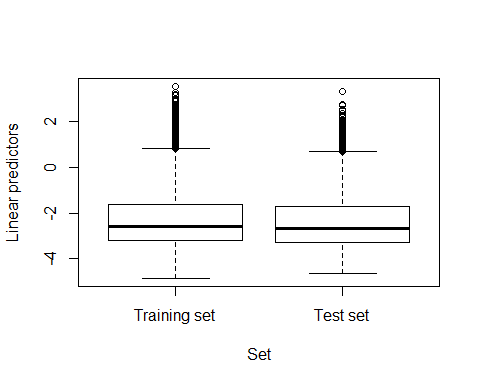
## [1] 0.1462443

# na.rm=TRUE makes sure that NAs are removed before the mean is computed; otherwise the mean would be NA itself  
mean(pred.test, na.rm=TRUE)

## [1] 0.1441697

One reason for miscalibration can be a different casemix in the test set. To investigate the comparability of the training and the test set (the ‘casemix’), one can compare the linear predictors between the two cohorts:

linpred.train <- predict(fit.crash2, crash2.train)  
  
boxplot(linpred.train, linpred.test, ylab="Linear predictors", xlab="Set", axes=F)  
axis(1, at=c(1,2), lab=c("Training set", "Test set"))  
axis(2)  
box()



### Exercises 4.2

1. Try to compute a logistic regression of early death on the linear predictors in the training set! What are the intercept and the slope?
2. How do you interpret the coefficient of discrimination in the test set?
3. How do you interpret the c-index in the test set?