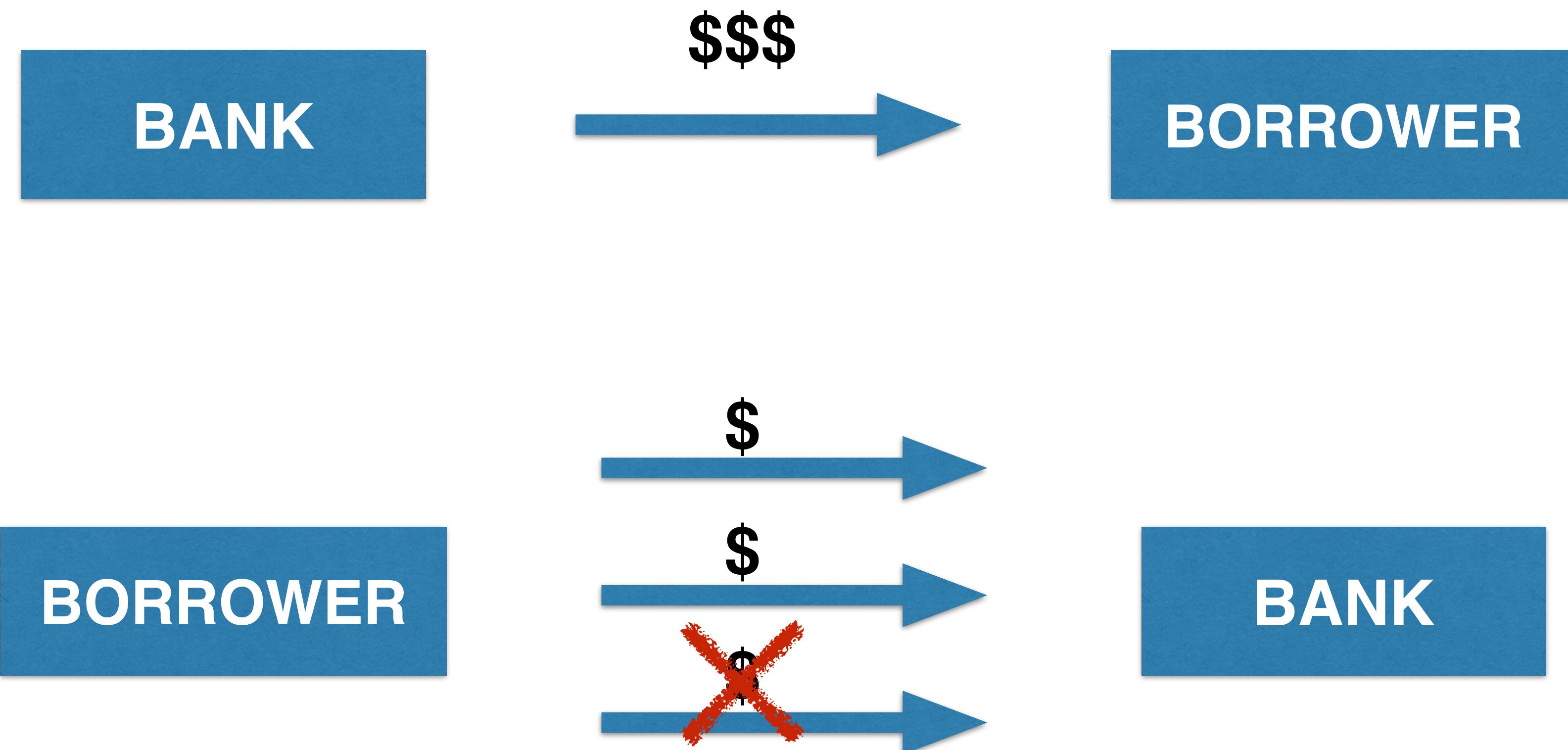




CREDIT RISK MODELING IN R

# Introduction and data structure

# What is loan default?



# Components of expected loss (EL)

- Probability of default (PD)
- Exposure at default (EAD)
- Loss given default (LGD)

$$\text{EL} = \text{PD} \times \text{EAD} \times \text{LGD}$$

# Information used by banks

- Application information:
  - income
  - marital status
  - ...
- Behavioral information
  - current account balance
  - payment arrears in account history
  - ...

# The data

```
> head(loan_data, 10)
  loan_status loan_amnt int_rate grade emp_length home_ownership annual_inc age
1          0      5000   10.65     B         10        RENT    24000  33
2          0      2400      NA     C         25        RENT    12252  31
3          0     10000   13.49     C         13        RENT    49200  24
4          0      5000      NA     A          3        RENT    36000  39
5          0      3000      NA     E          9        RENT    48000  24
6          0     12000   12.69     B         11        OWN    75000  28
7          1      9000   13.49     C          0        RENT    30000  22
8          0      3000   9.91     B          3        RENT    15000  22
9          1     10000   10.65     B          3        RENT  100000  28
10         0      1000   16.29     D          0        RENT    28000  22
```

# CrossTable

```
> library(gmodels)
> CrossTable(loan_data$home_ownership)
```

Cell Contents

		N
		N / Table Total

Total Observations in Table: 29092

MORTGAGE	OTHER	OWN	RENT
12002	97	2301	14692
0.413	0.003	0.079	0.505

# CrossTable

```
> CrossTable(loan_data$home_ownership, loan_data$loan_status, prop.r = TRUE,  
prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE)
```

loan_data\$home_ownership	loan_data\$loan_status		Row Total
	0	1	
MORTGAGE	10821	1181	12002
	0.902	0.098	0.413
OTHER	80	17	97
	0.825	0.175	0.003
OWN	2049	252	2301
	0.890	0.110	0.079
RENT	12915	1777	14692
	0.879	0.121	0.505
Column Total	25865	3227	29092



CREDIT RISK MODELING IN R

**Let's practice!**

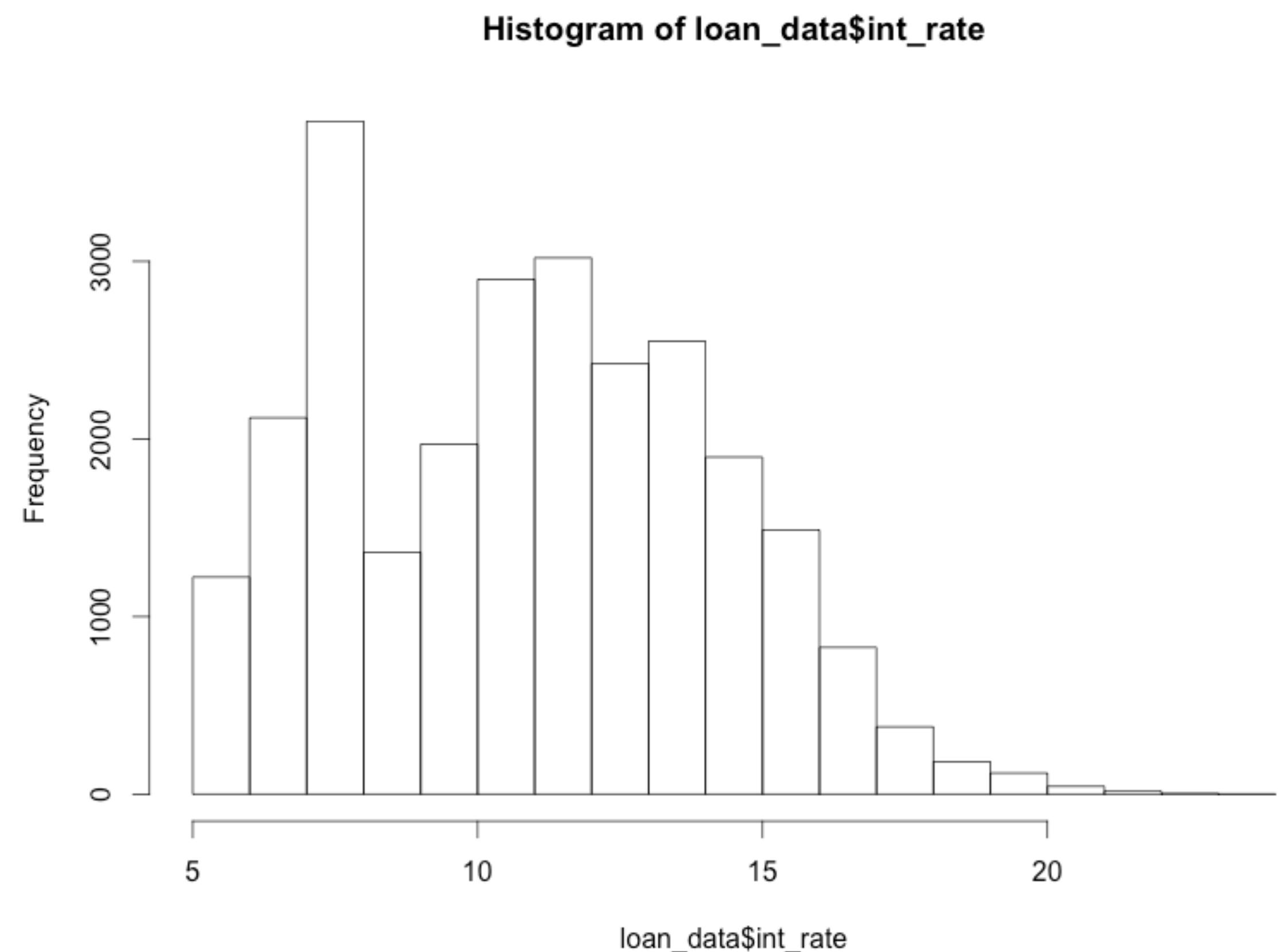


CREDIT RISK MODELING IN R

# Histograms and outliers

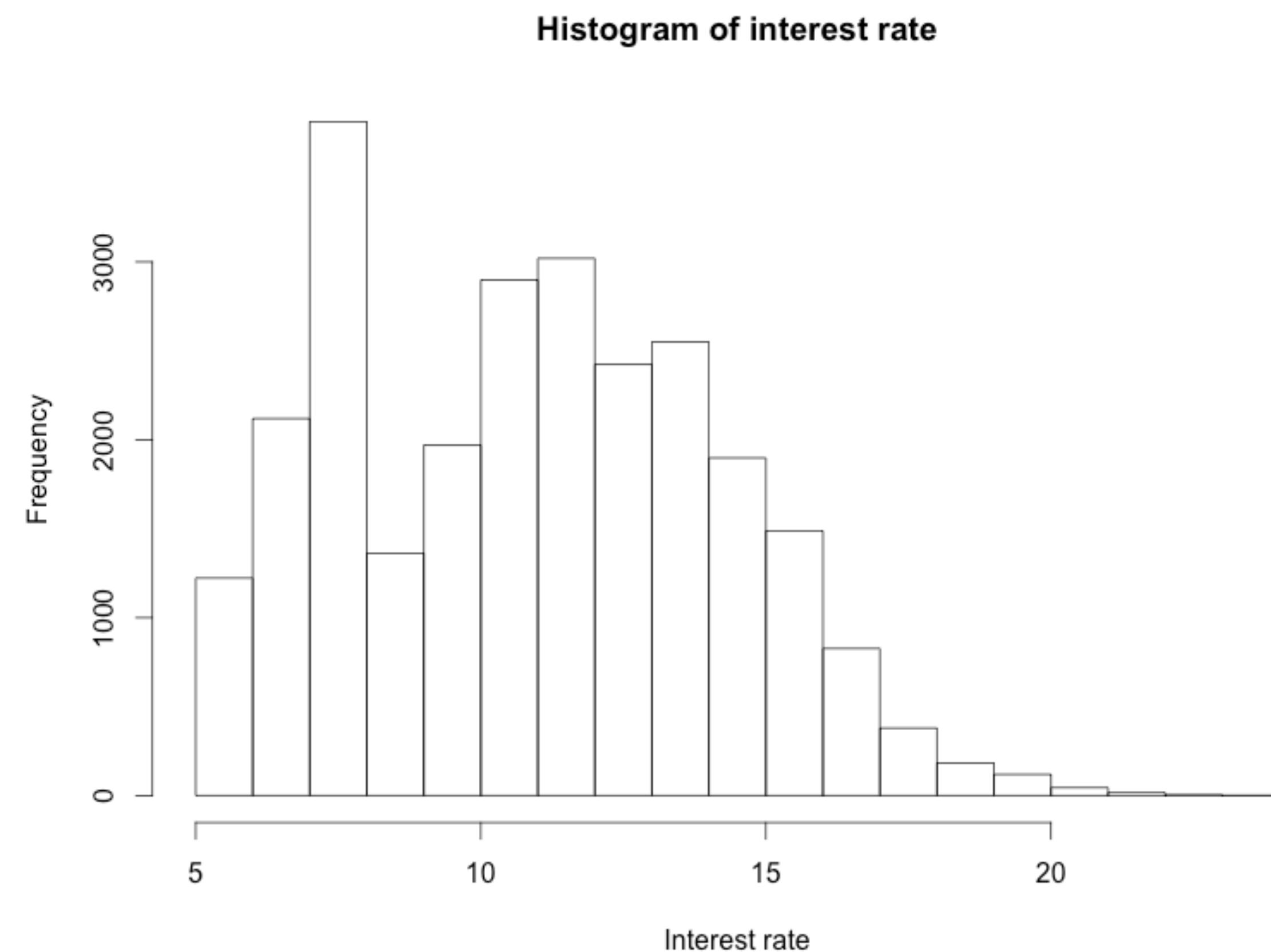
# Using function hist()

```
> hist(loan_data$int_rate)
```



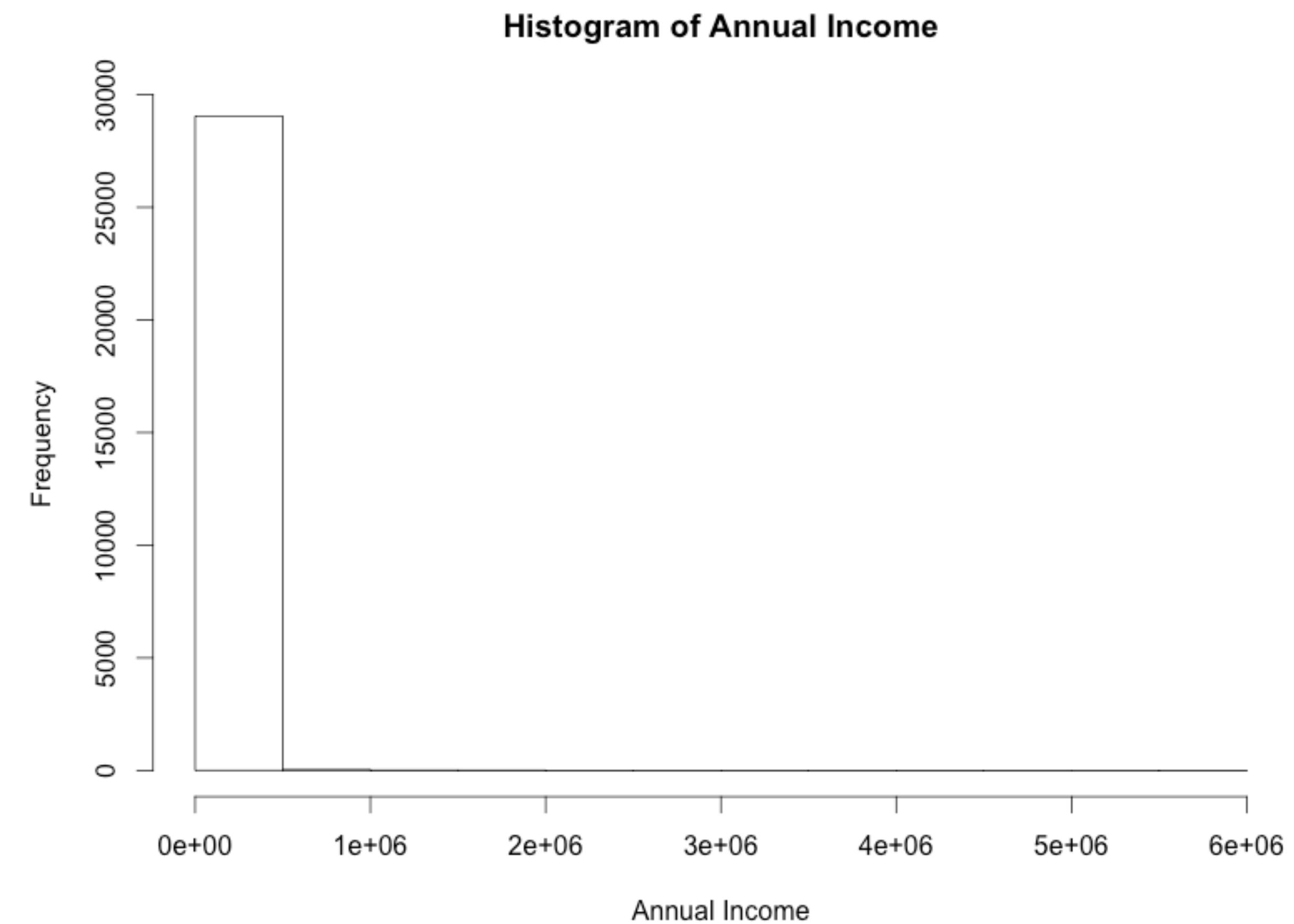
# Using function `hist()`

```
> hist(loan_data$int_rate, main = "Histogram of interest rate", xlab = "Interest rate")
```



# Using function hist() on annual\_inc

```
hist(loan_data$annual_inc, xlab= "Annual Income", main= "Histogram of Annual Income")
```

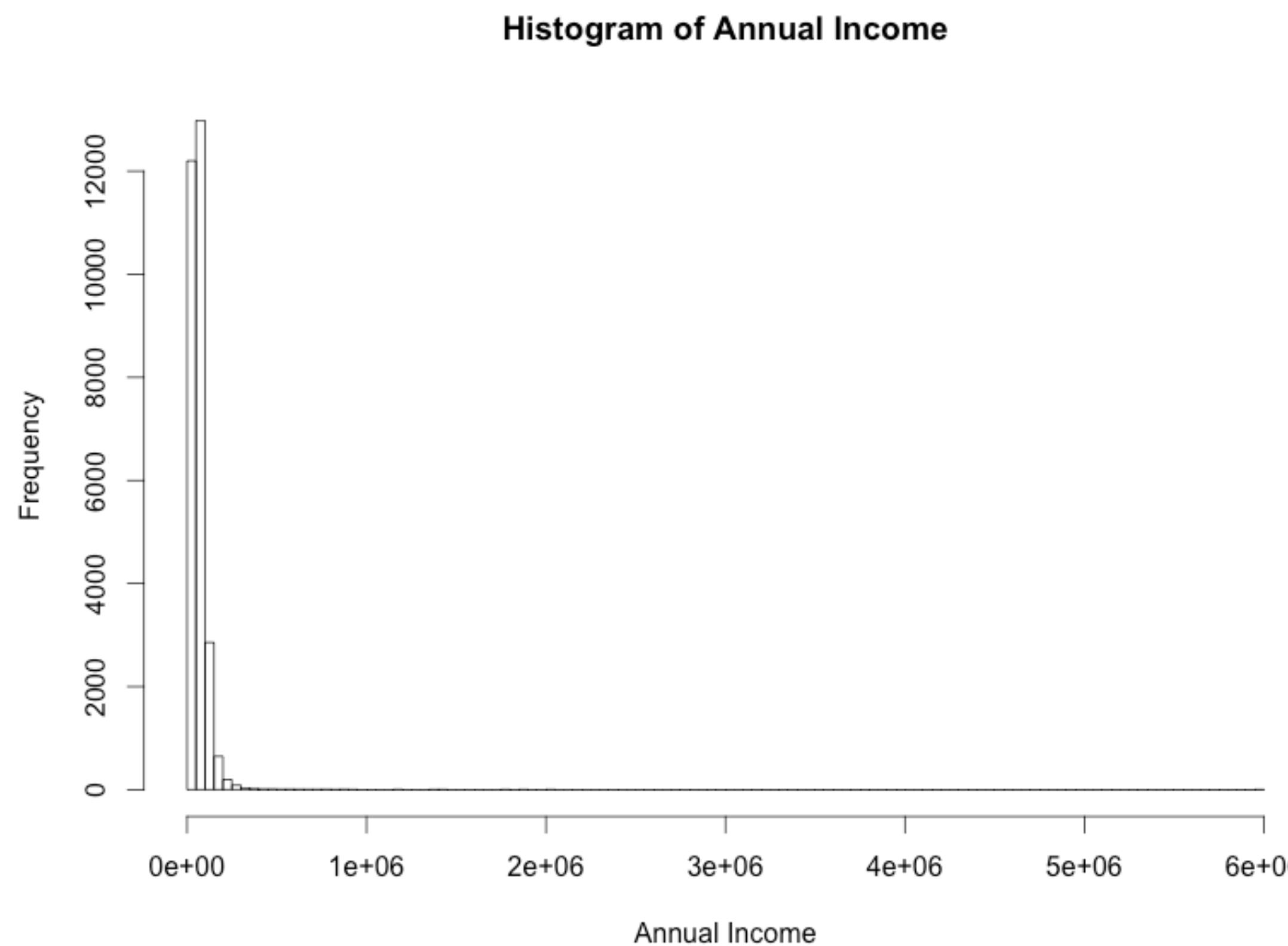


# Using function hist() on annual\_inc

```
> hist_income <- hist(loan_data$annual_inc, xlab = "Annual Income", main =  
"Histogram of Annual Income")  
  
> hist_income$breaks  
[1] 0 500000 1000000 1500000 2000000 2500000 3000000 3500000 4000000  
4500000 5000000 5500000 6000000
```

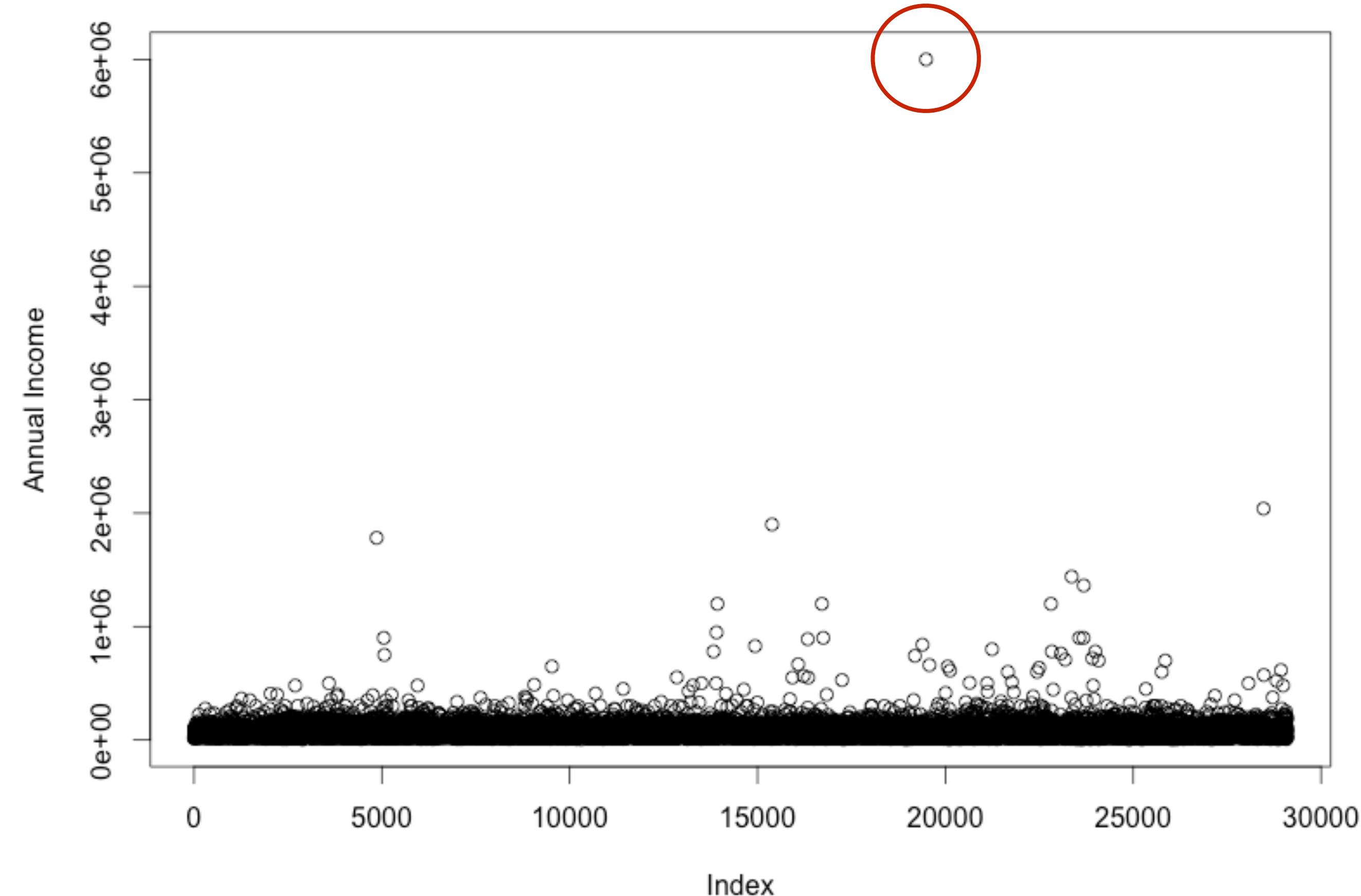
# The breaks-argument

```
> n_breaks <- sqrt(nrow(loan_data)) # = 170.5638  
  
> hist_income_n <- hist(loan_data$annual_inc, breaks= n_breaks, xlab = "Annual  
Income", main = "Histogram of Annual Income")
```



# annual\_inc

```
plot(loan_data$annual_inc, ylab = "Annual Income")
```



# Outliers

- When is a value an outlier?
  - expert judgement
  - rule of thumb:  $Q_1 - 1.5 * IQR$   
 $Q_3 + 1.5 * IQR$
  - mostly: combination of both

# Expert judgement - rule of thumb

**“Annual salaries > \$ 3 million are outliers”**

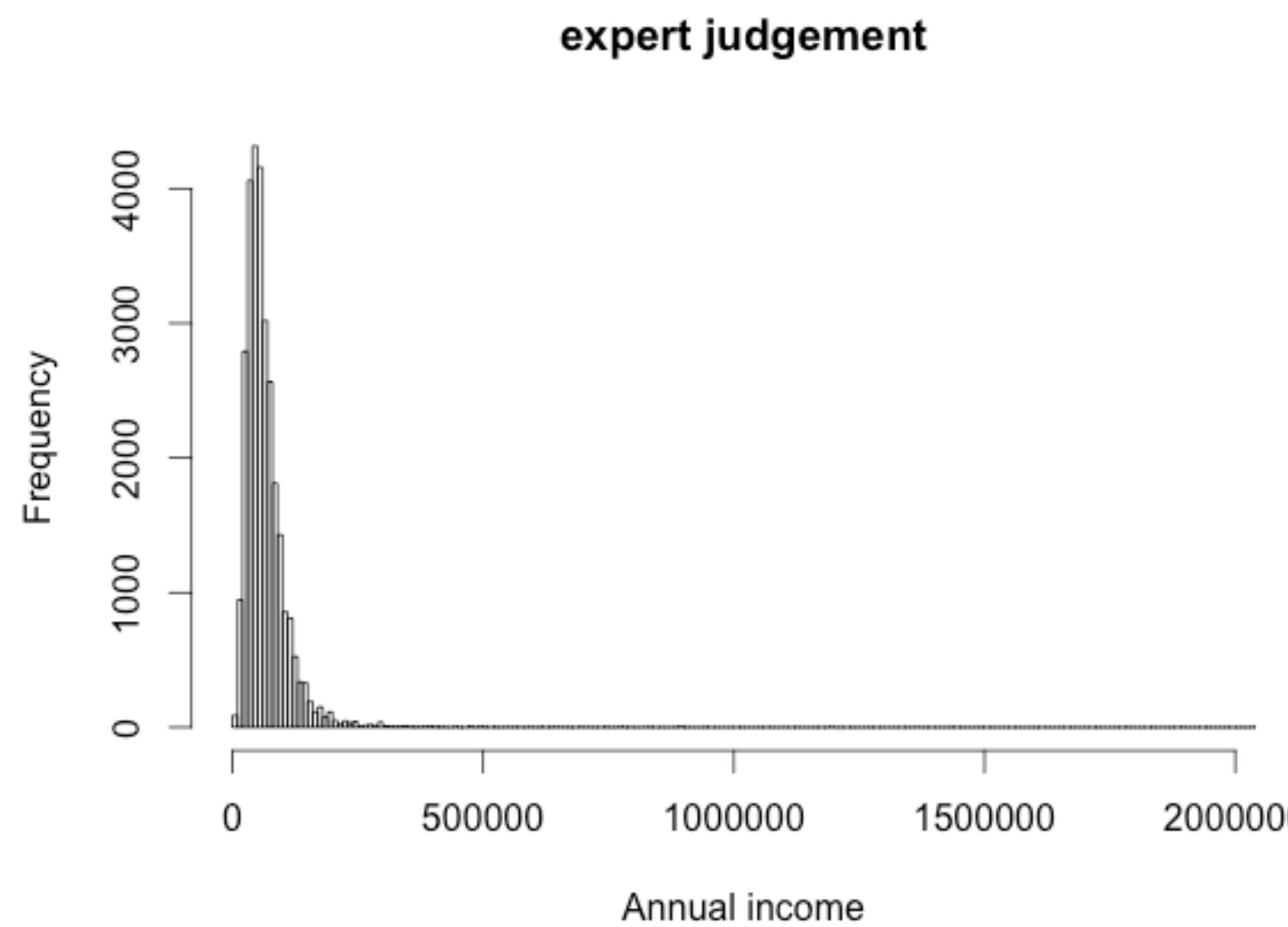
```
> index_outlier_expert <- which(loan_data$annual_inc > 3000000)  
> loan_data_expert <- loan_data[-index_outlier_expert, ]
```

**Use of a rule of thumb: outlier if bigger than  $Q3 + 1.5 * IQR$**

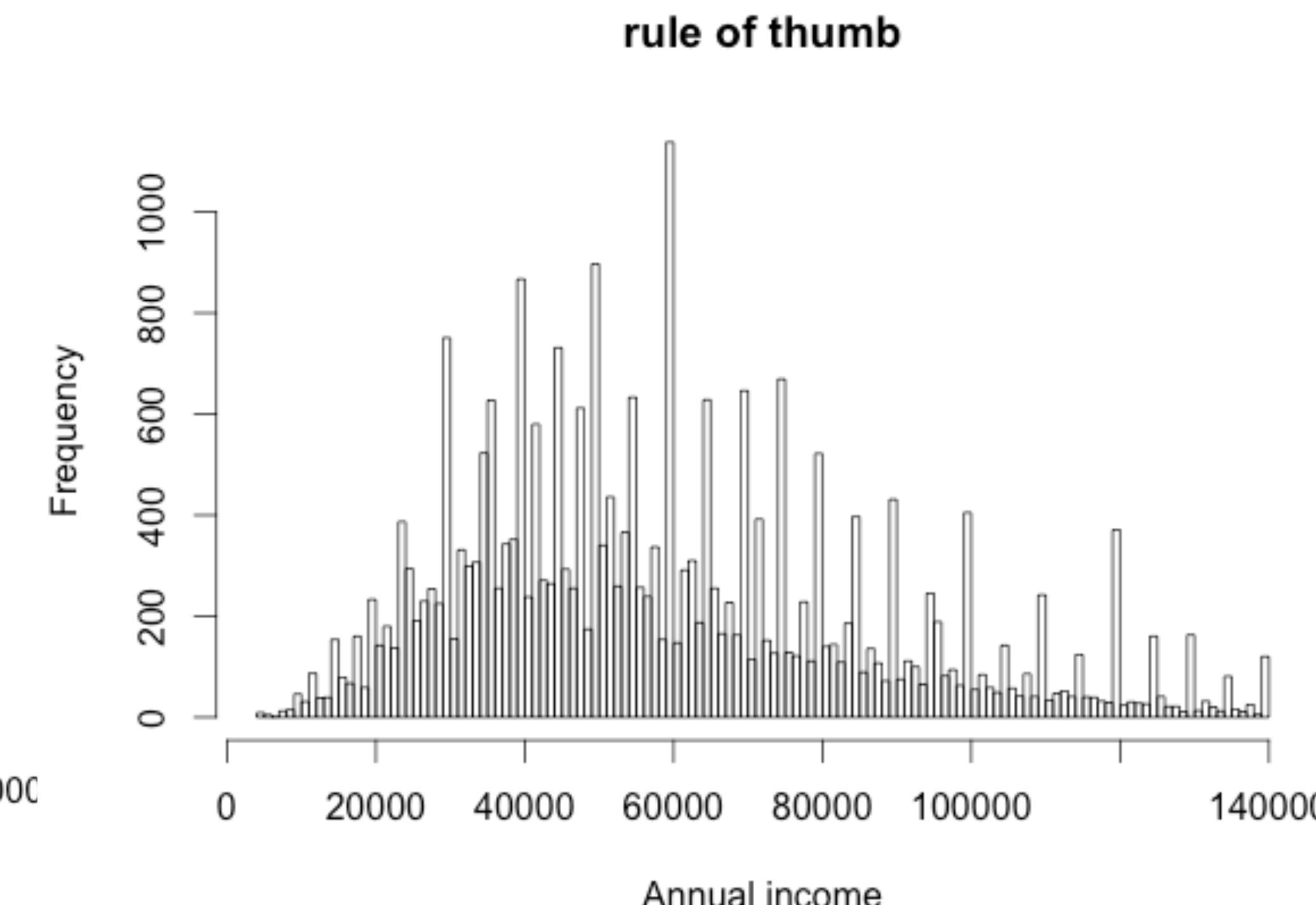
```
outlier_cutoff <- quantile(loan_data$annual_inc, 0.75) + 1.5 * IQR(loan_data$annual_inc)  
index_outlier_ROT <- which(loan_data$annual_inc > outlier_cutoff)  
loan_data_ROT <- loan_data[-index_outlier_ROT, ]
```

# histograms

```
hist(loan_data_expert$annual_inc,  
sqrt(nrow(loan_data_expert)), xlab =  
"Annual income expert judgement")
```

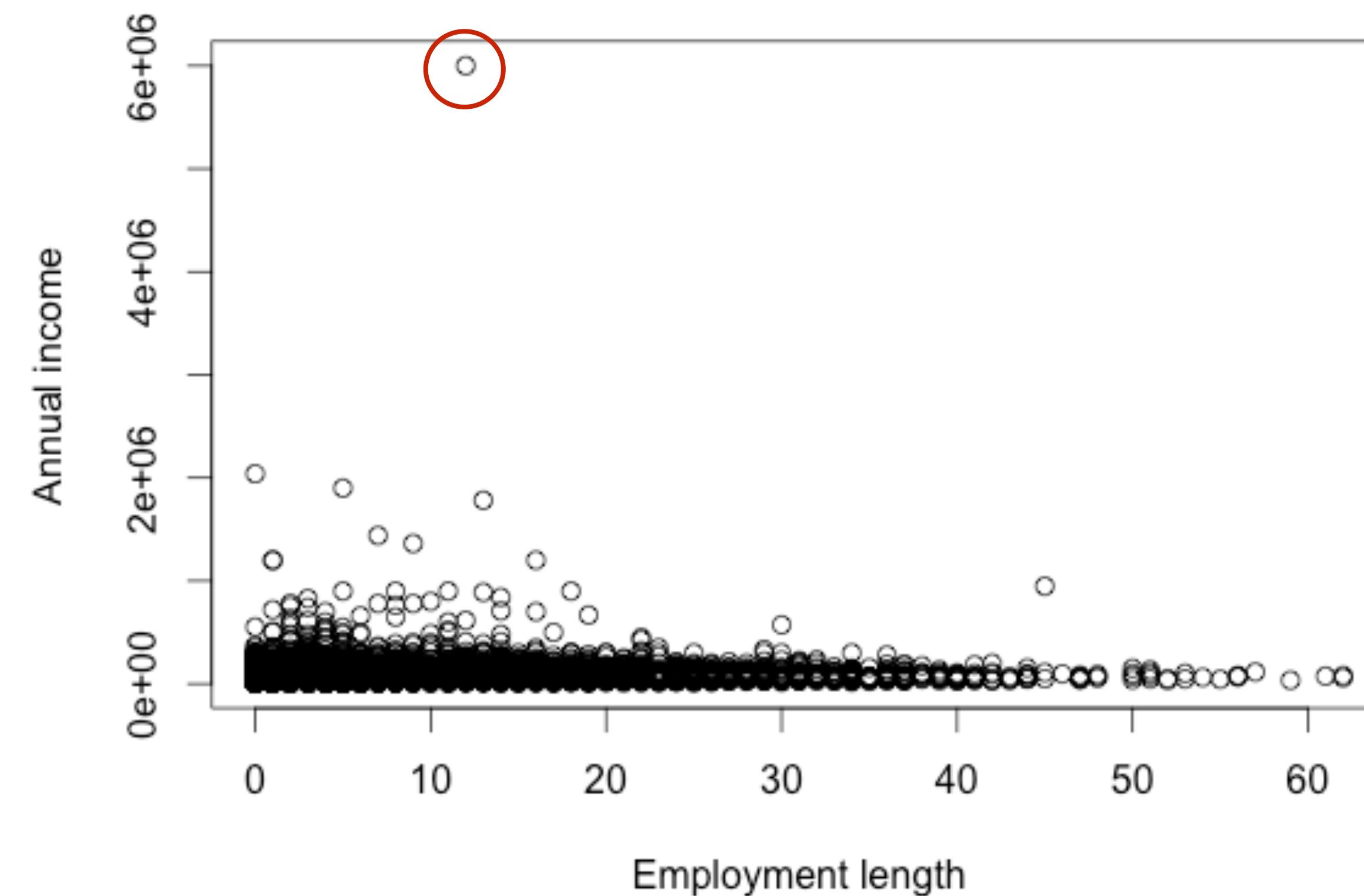


```
hist(loan_data_ROT$annual_inc,  
sqrt(nrow(loan_data_ROT)), xlab =  
"Annual income rule of thumb")
```



# bivariate plot

```
plot(loan_data$emp_length, loan_data$annual_inc, xlab= "Employment length",
ylab= "Annual income")
```





CREDIT RISK MODELING IN R

**Let's practice!**



CREDIT RISK MODELING IN R

# Missing data and coarse classification

# Outlier deleted

loan_status	loan_amnt	int_rate	grade	emp_length	home_ownership	annual_inc	age
0	5000	12.73	C	12	MORTGAGE	8000000	144



# Missing inputs

# Missing inputs

```
> summary(loan_data$emp_length)
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
emp_length	0.000	2.000	4.000	6.145	8.000	62.000	809

# Missing inputs: strategies

- Delete row/column
- Replace
- Keep



# Delete rows

```
index_NA <- which(is.na(loan_data$emp_length))
loan_data_no_NA <- loan_data[-c(index_NA), ]
```



# Delete column

```
loan_data_delete_employ <- loan_data  
loan_data_delete_employ$emp_length <- NULL
```

loan_status	loan_amnt	int_rate	grade	emp_length	home_ownership	annual_inc	age
...	...	...	...	...	...	...	...
125	0	6000	14.27	C	14	94800	23
126	1	2500	7.51	A	NA	12000	21
127	0	13500	9.91	B	2	36000	30
128	0	25000	12.42	B	2	225000	30
129	0	10000	NA	C	2	45900	65
130	0	2500	13.49	C	4	27200	26
...	...	...	...	...	...	...	...
2112	0	7600	6.03	A	41	70920	28
2113	0	10000	11.71	B	5	48132	22
2114	0	8000	6.62	A	17	42000	24
2115	0	4475	NA	B	NA	15000	23
2116	0	5750	8.90	A	3	17000	21



# Replace: median imputation

```
index_NA <- which(is.na(loan_data$emp_length))
loan_data_replace <- loan_data
loan_data_replace$emp_length[index_NA] <- median(loan_data$emp_length, na.rm = TRUE)
```



# Replace: median imputation

```
index_NA <- which(is.na(loan_data$emp_length))
loan_data_replace <- loan_data
loan_data_replace$emp_length[index_NA] <- median(loan_data$emp_length, na.rm = TRUE)
```

# Keep

- Keep NA
- Problem: will cause row deletions for many models
- Solution: coarse classification, put variable in “bins”
  - new variable emp\_cat
  - range: 0-62 years —> make bins of +/- 15 years
  - categories: “0-15”, “15-30”, “30-45”, “45+”, “missing”



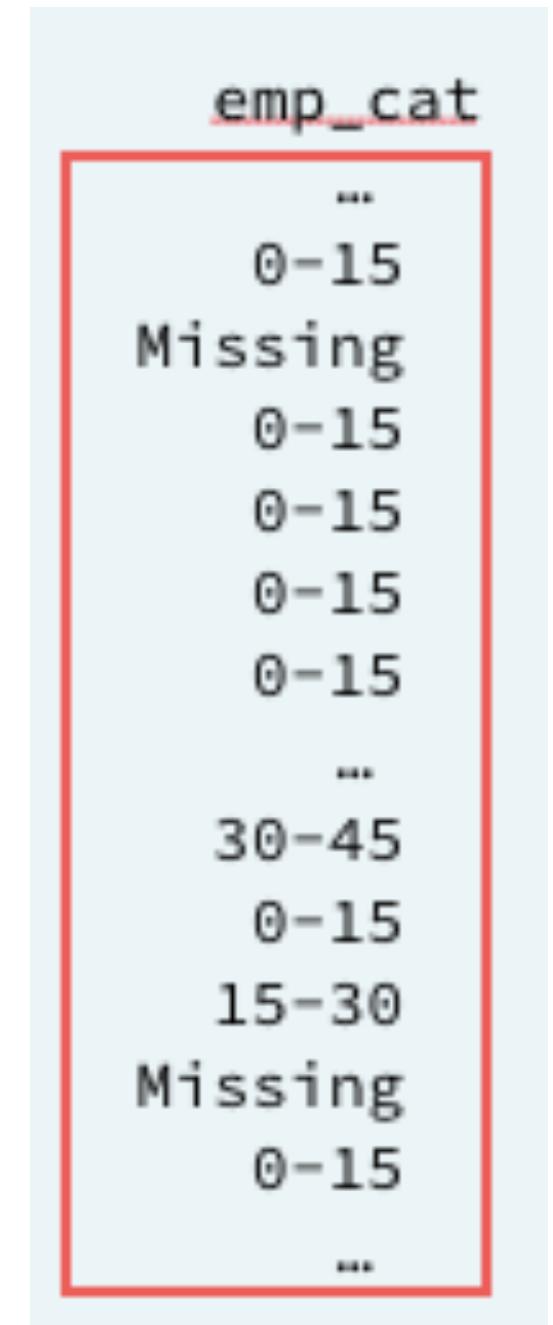
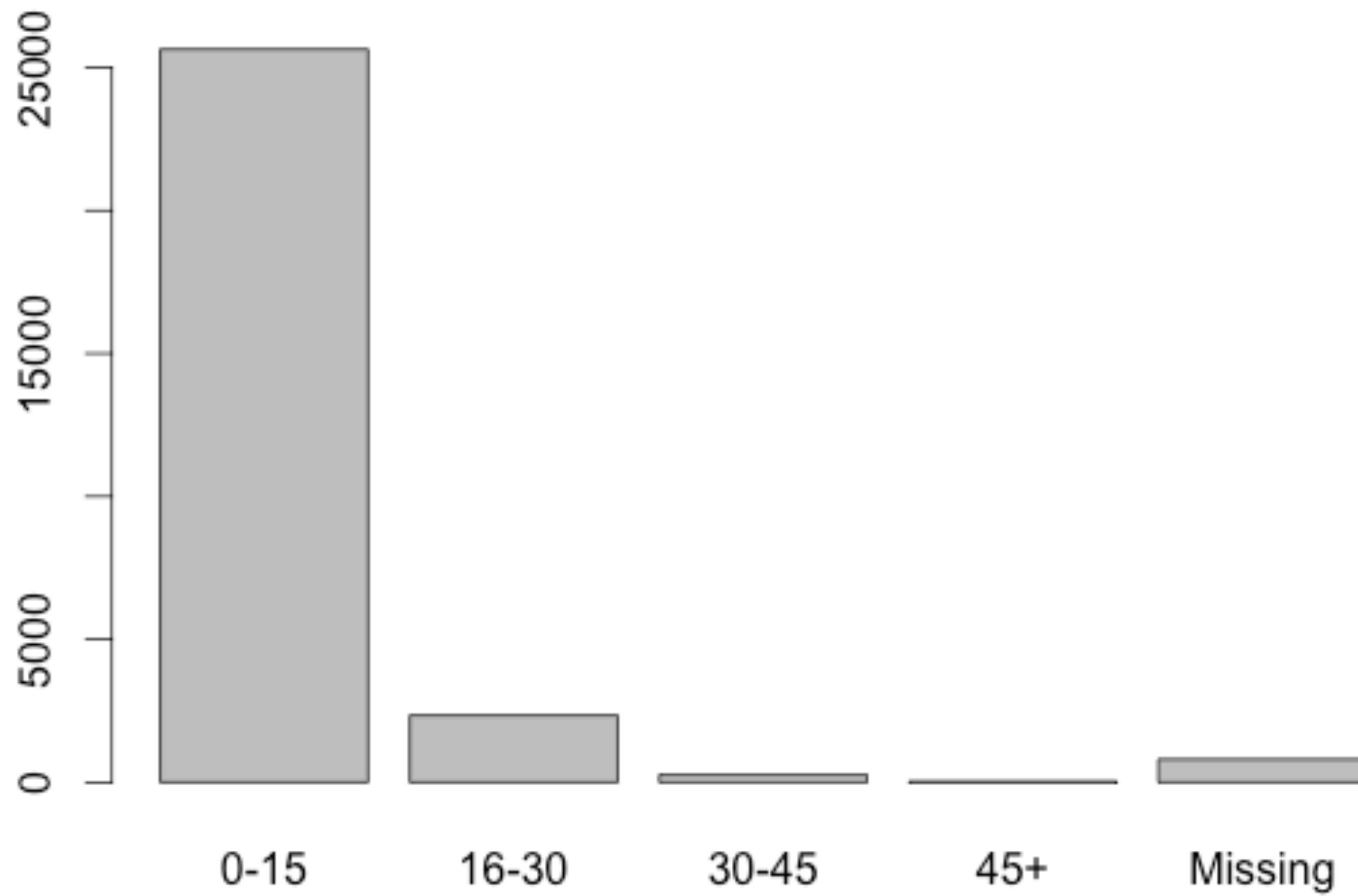
# Keep: coarse classification



# Keep: coarse classification

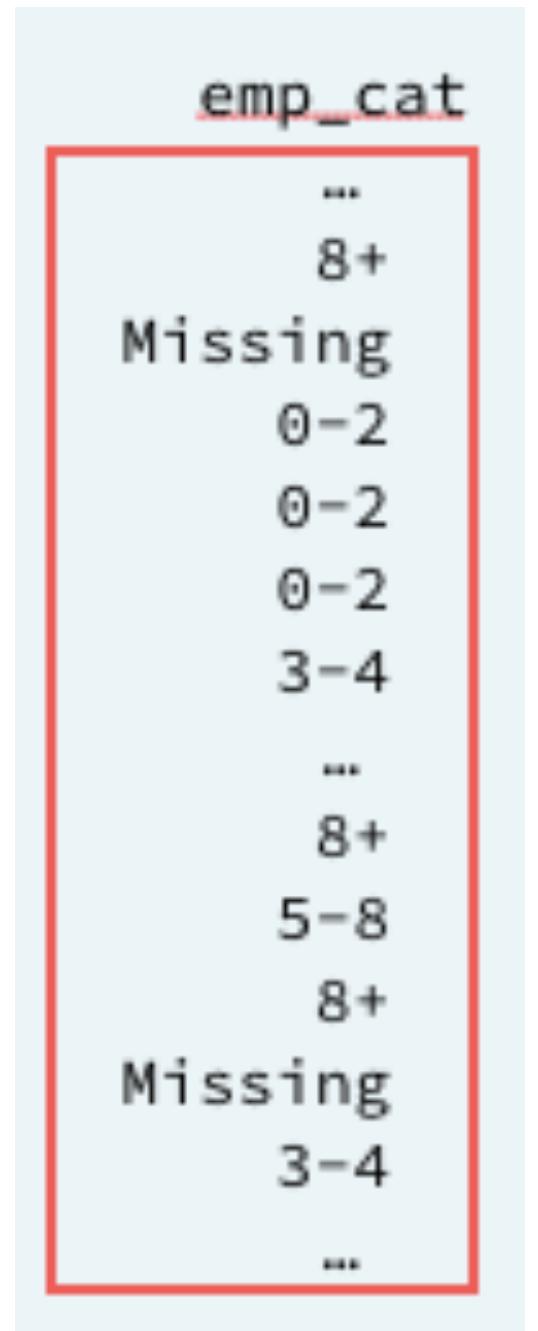
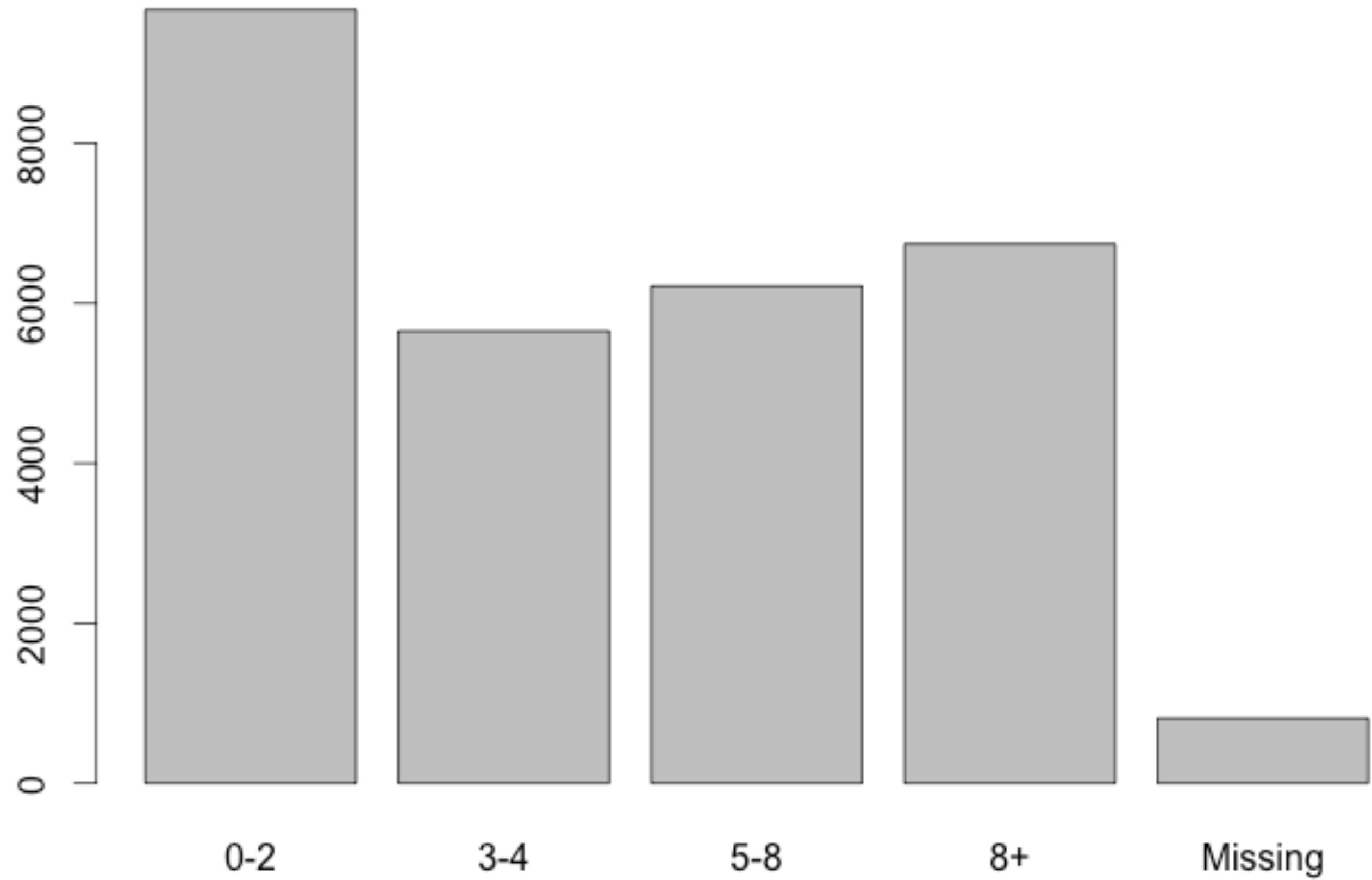
# Bin frequencies

```
plot(loan_data$emp_cat)
```



# Bin frequencies

```
plot(loan_data$emp_cat)
```



# Final remarks

	CONTINUOUS	CATEGORICAL
DELETE	Delete rows (observations with NAs) Delete column (entire variable)	Delete rows (observations with NAs) Delete column (entire variable)
REPLACE	replace using median	replace using most frequent category
KEEP	keep as NA (not always possible) keep using coarse classification	NA category



CREDIT RISK MODELING IN R

**Let's practice!**

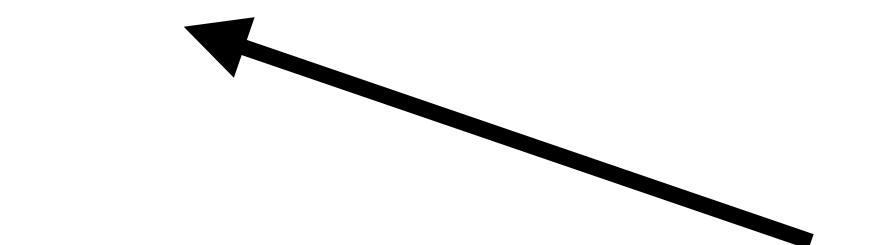
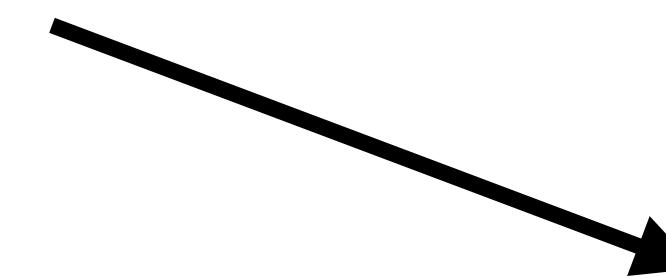


CREDIT RISK MODELING IN R

# Data splitting and confusion matrices

# Start analysis

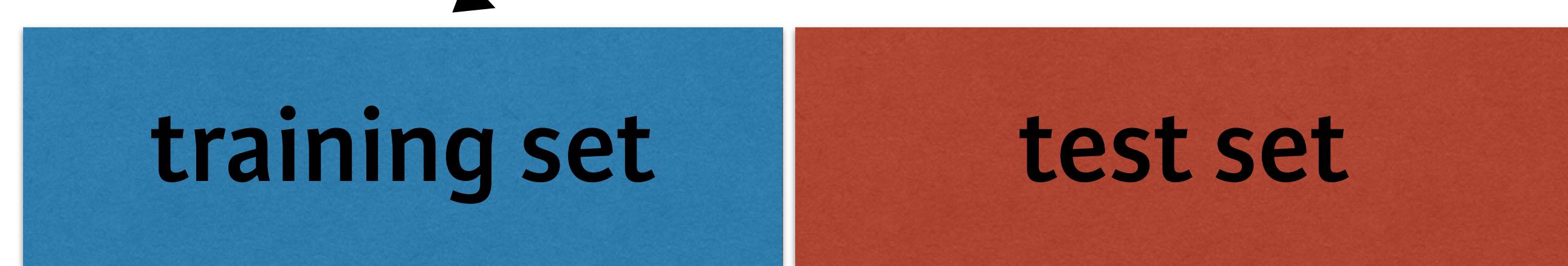
Run the model



evaluate the result

# training and test set

Run the model



evaluate the result

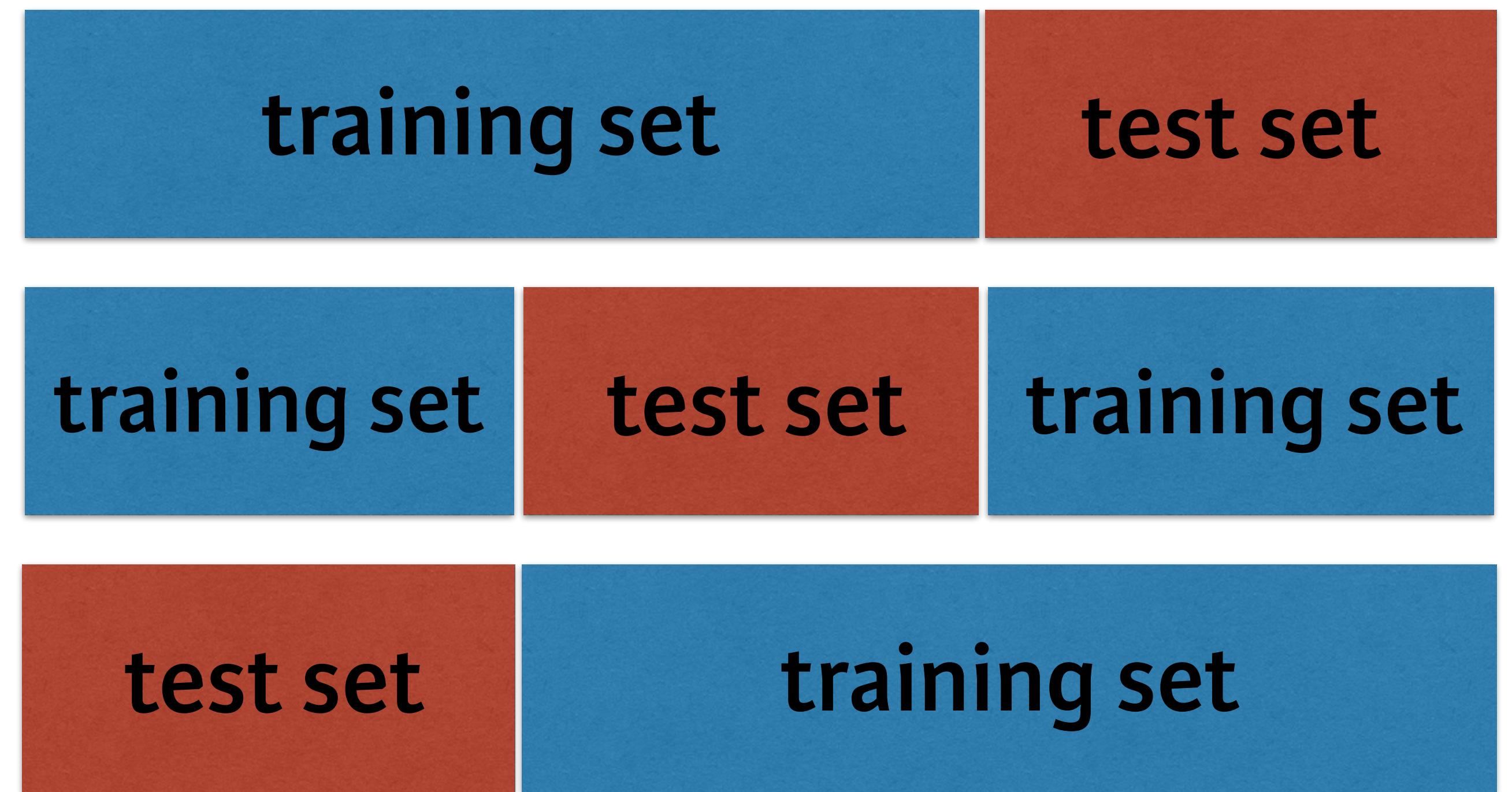
# training and test set

Run the model



evaluate the result

# cross-validation



# evaluate a model

	test_set\$loan_status	model_prediction
[8066, ]	...	...
[8067, ]	1	0
[8068, ]	0	0
[8069, ]	0	0
[8070, ]	0	0
[8071, ]	0	1
[8072, ]	1	0
[8073, ]	1	1
[8074, ]	0	0
[8075, ]	0	0
[8076, ]	0	0
[8077, ]	1	1
[8078, ]	0	0
[8079, ]	0	1
...	...	...

actual  
loan  
status

model prediction

	no default (0)	default (1)
no default (0)	8	2
default (1)	1	3

# evaluate a model

	test_set\$loan_status	model_prediction
[8066, ]	...	...
[8067, ]	1	0
[8068, ]	0	0
[8069, ]	0	0
[8070, ]	0	0
[8071, ]	0	1
[8072, ]	1	0
[8073, ]	1	1
[8074, ]	0	0
[8075, ]	0	0
[8076, ]	0	0
[8077, ]	1	1
[8078, ]	0	0
[8079, ]	0	1
...	...	...

actual  
loan  
status

model prediction

	no default (0)	default (1)
no default (0)	TN	FP
default (1)	FN	TP

# some measures...

- Accuracy =  $(8 + 3) / 14 = 78.57\%$
- Sensitivity =  $3 / (1 + 3) = 75 \%$
- Specificity =  $8 / (8 + 2) = 80\%$

		model prediction	
		no default (0)	default (1)
actual loan status	no default (0)	8	2
	default (1)	1	3



CREDIT RISK MODELING IN R

**Let's practice!**



CREDIT RISK MODELING IN R

# Logistic regression: introduction

# Final data structure

```
> str(training_set)

'data.frame': 19394 obs. of  8 variables:
 $ loan_status    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ loan_amnt      : int  25000 16000 8500 9800 3600 6600 3000 7500 6000 22750 ...
 $ grade          : Factor w/ 7 levels "A","B","C","D",...: 2 4 1 2 1 1 1 2 1 1 ...
 $ home_ownership: Factor w/ 4 levels "MORTGAGE","OTHER",...: 4 4 1 1 1 3 4 3 4 1 ...
 $ annual_inc     : num  91000 45000 110000 102000 40000 ...
 $ age            : int  34 25 29 24 59 35 24 24 26 25 ...
 $ emp_cat        : Factor w/ 5 levels "0-15","15-30",...: 1 1 1 1 1 2 1 1 1 1 ...
 $ ir_cat         : Factor w/ 5 levels "0-8","11-13.5",...: 2 3 1 4 1 1 1 4 1 1 ...
```

# What is logistic regression?

A regression model with output between 0 and 1

$$P(\text{loan\_status} = 1 \mid x_1, \dots, x_m) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_m x_m)}}$$

$x_1, \dots, x_m$

loan\_amnt      grade      age      annual\_inc  
home\_ownership      emp\_cat      irr\_cat

$\beta_0, \dots, \beta_m$

Parameters to be estimated

$\beta_0 + \beta_1 x_1 + \dots + \beta_m x_m$

Linear predictor

# Fitting a logistic model in R

```
> log_model <- glm(loan_status ~ age , family= "binomial", data = training_set)
> log_model
```

```
Call: glm(formula = loan_status ~ age, family = "binomial", data = training_set)
```

Coefficients:

(Intercept)   
-1.793566

age   
-0.009726

Degrees of Freedom: 19393 Total (i.e. Null); 19392 Residual

Null Deviance: 13680

Residual Deviance: 13670 AIC: 13670

$\hat{\beta}_0$

$\hat{\beta}_1$

$$P(\text{loan\_status} = 1 \mid \text{age}) = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 \text{age})}}$$

# Probabilities of default

$$P(\text{loan\_status} = 1 \mid x_1, \dots, x_m) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_m x_m)}} = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_m x_m}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_m x_m}}$$

$$P(\text{loan\_status} = 0 \mid x_1, \dots, x_m) = 1 - \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_m x_m}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_m x_m}} = \frac{1}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_m x_m}}$$

$$\frac{P(\text{loan\_status} = 1 \mid x_1, \dots, x_m)}{P(\text{loan\_status} = 0 \mid x_1, \dots, x_m)} = e^{\beta_0 + \beta_1 x_1 + \dots + \beta_m x_m} \rightarrow \text{odds in favor of loan\_status=1}$$

# Interpretation of coefficient

If variable  $x_j$  goes up by 1 → The odds are multiplied by  $e^{\beta_j}$

$\beta_j < 0$  →  $e^{\beta_j} < 1$  → The odds decrease as  $x_j$  increases

$\beta_j > 0$  →  $e^{\beta_j} > 1$  → The odds increase as  $x_j$  increases

## Applied to our model

If variable age goes up by 1 → The odds are multiplied by  $e^{-0.009726}$   
→ The odds are multiplied by 0.991



CREDIT RISK MODELING IN R

**Let's practice!**



CREDIT RISK MODELING IN R

# **Logistic regression: predicting the probability of default**

# An example with “age” and “home ownership” $\hat{\beta}_0$

```
> log_model_small <- glm(loan_status ~ age + home_ownership, family = "binomial", data = training_set)
> log_model_small
```

```
Call: glm(formula = loan_status ~ age + home_ownership, family = "binomial",
          data = training_set)
```

Coefficients:

	(Intercept)	age	home_ownershipOTHER	home_ownershipOWN	home_ownershipRENT
	-1.886396	-0.009308	0.129776	-0.019384	0.158581

Degrees of Freedom: 19393 Total (i.e. Null); 19389 Residual

Null Deviance: 13680

Residual Deviance: 13660 AIC: 13670

$$P(\text{loan\_status} = 1 \mid \text{age}, \text{home\_ownership}) = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 \text{age} + \hat{\beta}_2 \text{OTHER} + \hat{\beta}_3 \text{OWN} + \hat{\beta}_4 \text{RENT})}}$$

# Test set example

$$P(\text{loan\_status} = 1 \mid \text{age} = 33, \text{home\_ownership} = \text{RENT})$$

$$= \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 * 33 + \hat{\beta}_2 * 0 + \hat{\beta}_3 * 0 + \hat{\beta}_4 * 1)}}$$

$$= \frac{1}{1 + e^{-( -1.886396 + (-0.009308) * 33 + (0.158581) * 1)}}$$

$$= 0.115579$$

# Making predictions in R

```
> test_case <- as.data.frame(test_set[1,])  
  
> test_case  
  loan_status loan_amnt grade home_ownership annual_inc    age emp_cat  ir_cat  
1        0       5000     B      RENT        24000  33 0-15   8-11
```

```
> predict(log_model_small, newdata = test_case)
```

1  
-2.03499


$$\hat{\beta}_0 + \hat{\beta}_1 \text{age} + \hat{\beta}_2 \text{OTHER} + \hat{\beta}_3 \text{OWN} + \hat{\beta}_4 \text{RENT}$$

```
> predict(log_model_small, newdata = test_case, type = "response")  
1  
0.1155779
```



CREDIT RISK MODELING IN R

**Let's practice!**



CREDIT RISK MODELING IN R

# Evaluating the logistic regression model result

# Recap: model evaluation

test_set\$loan_status	model_prediction
[8066,]	...
[8067,]	1
[8068,]	0
[8069,]	0
[8070,]	0
[8071,]	0
[8072,]	1
[8073,]	1
[8074,]	0
[8075,]	0
[8076,]	0
[8077,]	1
[8078,]	0
[8079,]	0
...	...

actual  
loan  
status

model prediction

	no default (0)	default (1)
no default (0)	8	2
default (1)	1	3

# In reality...

```
test_set$loan_status
```

```
[8066,]      ...  
[8067,]      1  
[8068,]      0  
[8069,]      0  
[8070,]      0  
[8071,]      0  
[8072,]      1  
[8073,]      1  
[8074,]      0  
[8075,]      0  
[8076,]      0  
[8077,]      1  
[8078,]      0  
[8079,]      0  
...  
...
```

```
model_prediction
```

```
...  
0.09881492  
0.09497852  
0.21071984  
0.04252119  
0.21110838  
0.08668856  
0.11319341  
0.16662207  
0.15299176  
0.08558058  
0.08280463  
0.11271048  
0.08987446  
0.08561631  
...
```

actual  
loan  
status

model prediction

	no default (0)	default (1)
no default (0)	?	?
default (1)	?	?

# In reality...

test_set\$loan_status	model_prediction
[8066,] ...	0.09881492
[8067,] 1	0.09497852
[8068,] 0	0.21071984
[8069,] 0	0.04252119
[8070,] 0	0.21110838
[8071,] 0	0.08668856
[8072,] 1	0.11319341
[8073,] 1	0.16662207
[8074,] 0	0.15299176
[8075,] 0	0.08558058
[8076,] 0	0.08280463
[8077,] 1	0.11271048
[8078,] 0	0.08987446
[8079,] 0	0.08561631
...	...

Cutoff  
or  
threshold value  
**between 0 and 1**

# Cutoff = 0.5

test_set\$loan_status	model_prediction
[8066,]	...
[8067,]	1
[8068,]	0
[8069,]	0
[8070,]	0
[8071,]	0
[8072,]	1
[8073,]	1
[8074,]	0
[8075,]	0
[8076,]	0
[8077,]	1
[8078,]	0
[8079,]	0
...	...

model prediction

actual  
loan  
status

	no default (0)	default (1)
no default (0)	10	0
default (1)	4	0

$$\text{Accuracy} = 10 / (10 + 4 + 0 + 0) = 71.4\%$$

$$\text{Sensitivity} = 0 / (4 + 0) = 0\%$$

# Cutoff = 0.1

test_set\$loan_status	model_prediction
[8066,]	...
[8067,]	1
[8068,]	0
[8069,]	0
[8070,]	0
[8071,]	0
[8072,]	1
[8073,]	1
[8074,]	0
[8075,]	0
[8076,]	0
[8077,]	1
[8078,]	0
[8079,]	0
...	...

model prediction

actual  
loan  
status

	no default (0)	default (1)
no default (0)	7	3
default (1)	1	3

**Accuracy =  $10/(10+4+0+0) = 71.4\%$**

**Sensitivity =  $3/(3+1) = 75\%$**



CREDIT RISK MODELING IN R

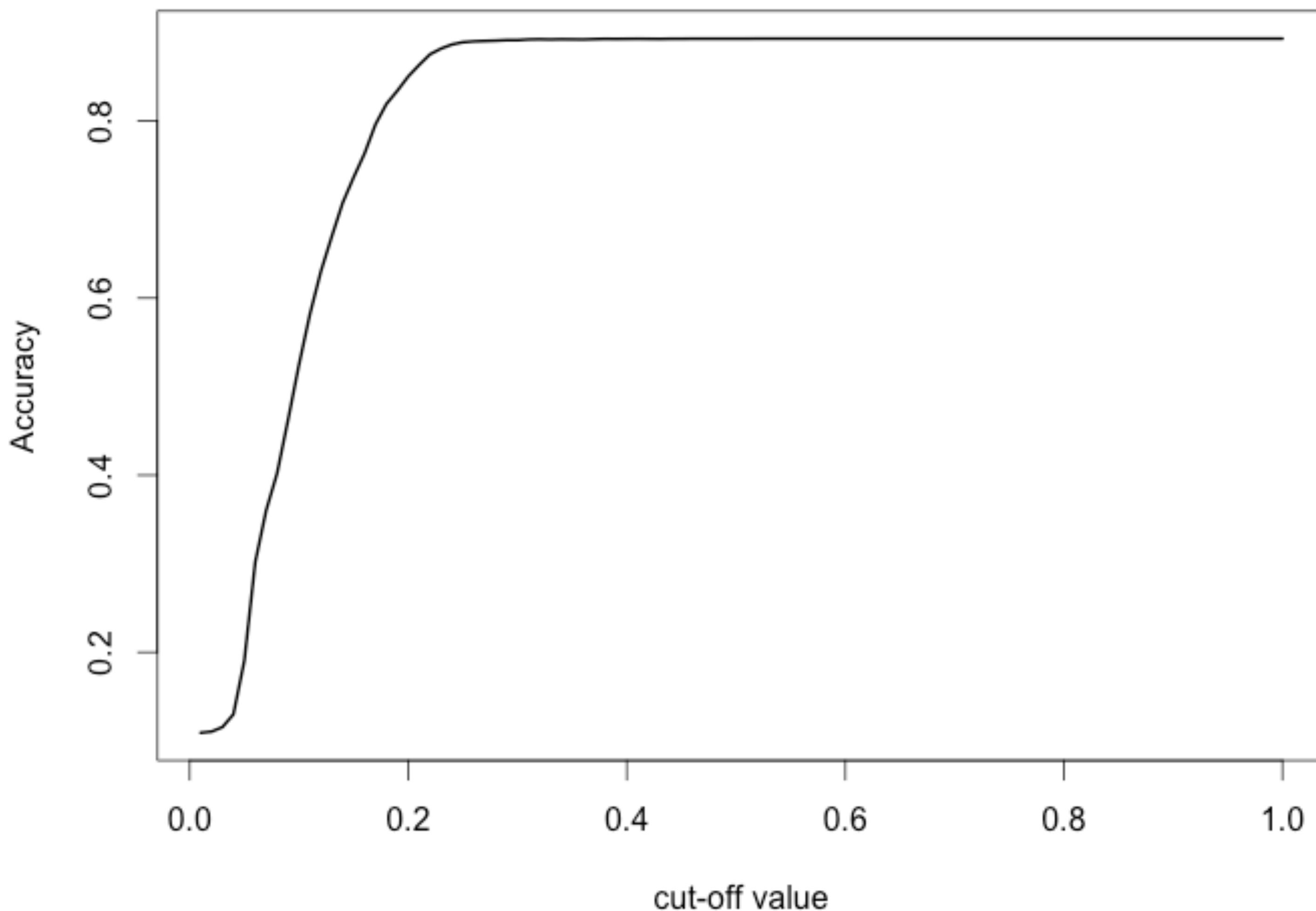
**Let's practice!**



CREDIT RISK MODELING IN R

# wrap-up and remarks

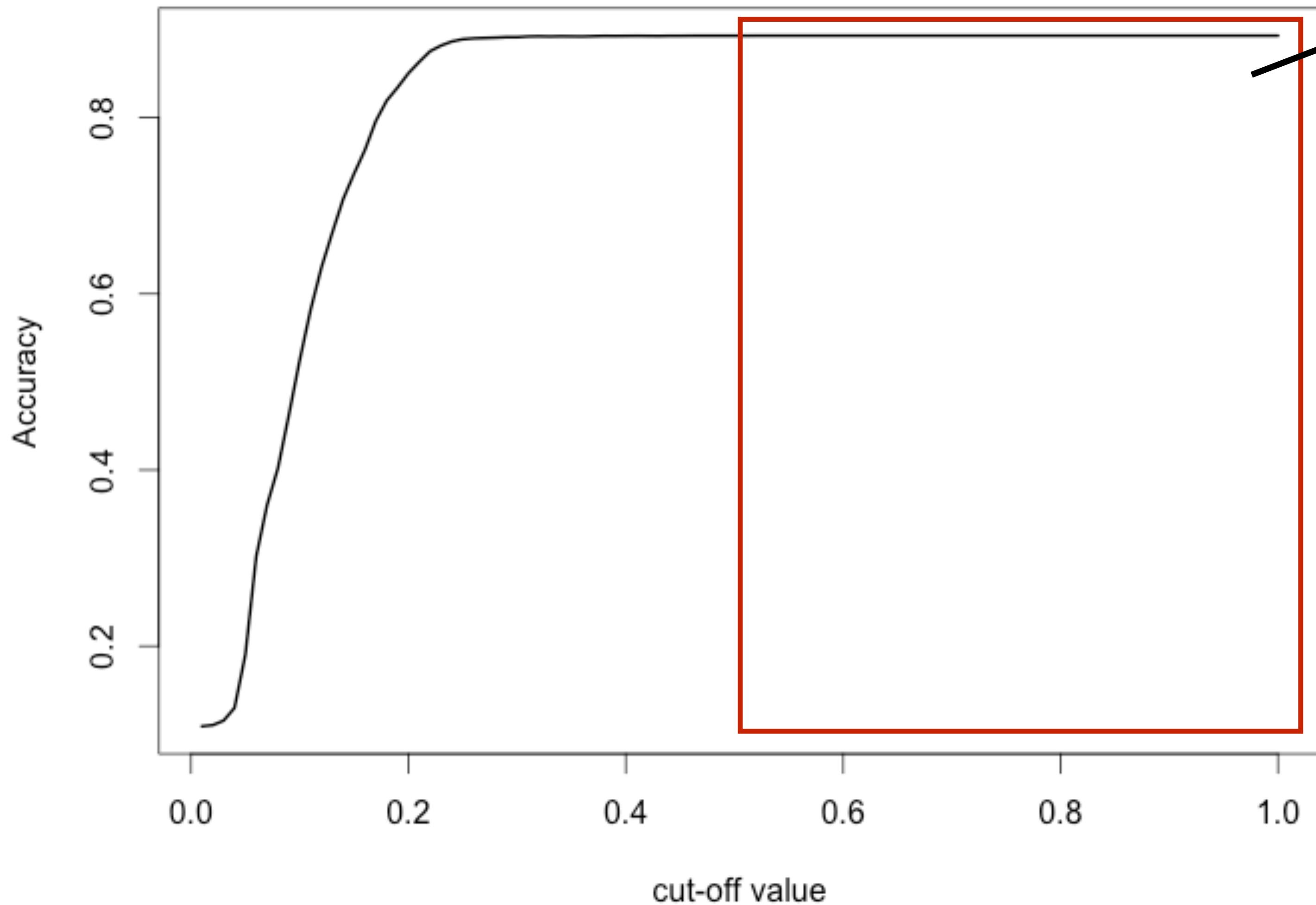
# best cut-off for accuracy?



$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

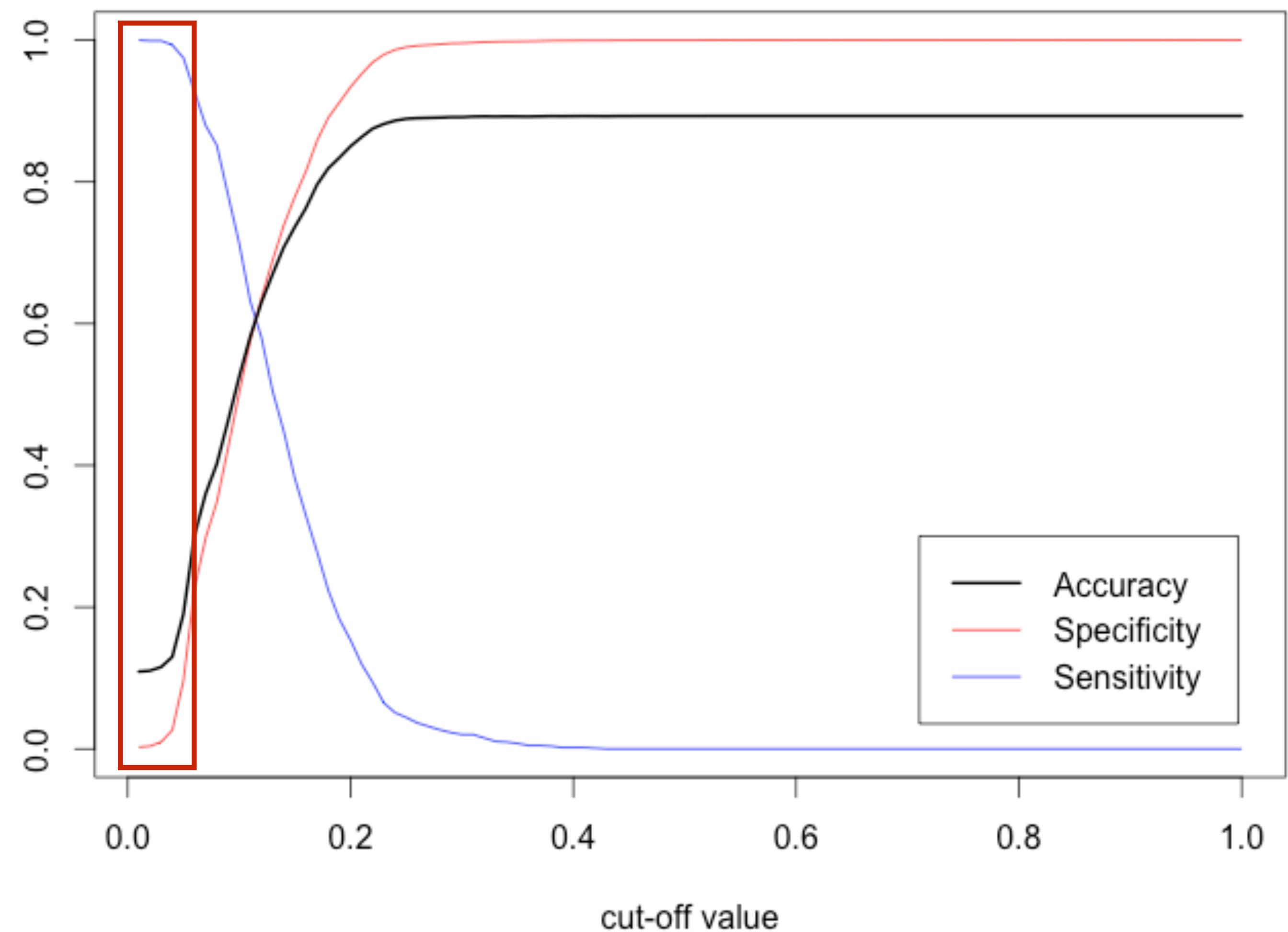
# best cut-off for accuracy?

Accuracy = 89.31 %



ACTUAL defaults in test set=  
10.69 % = (100 - 89.31) %

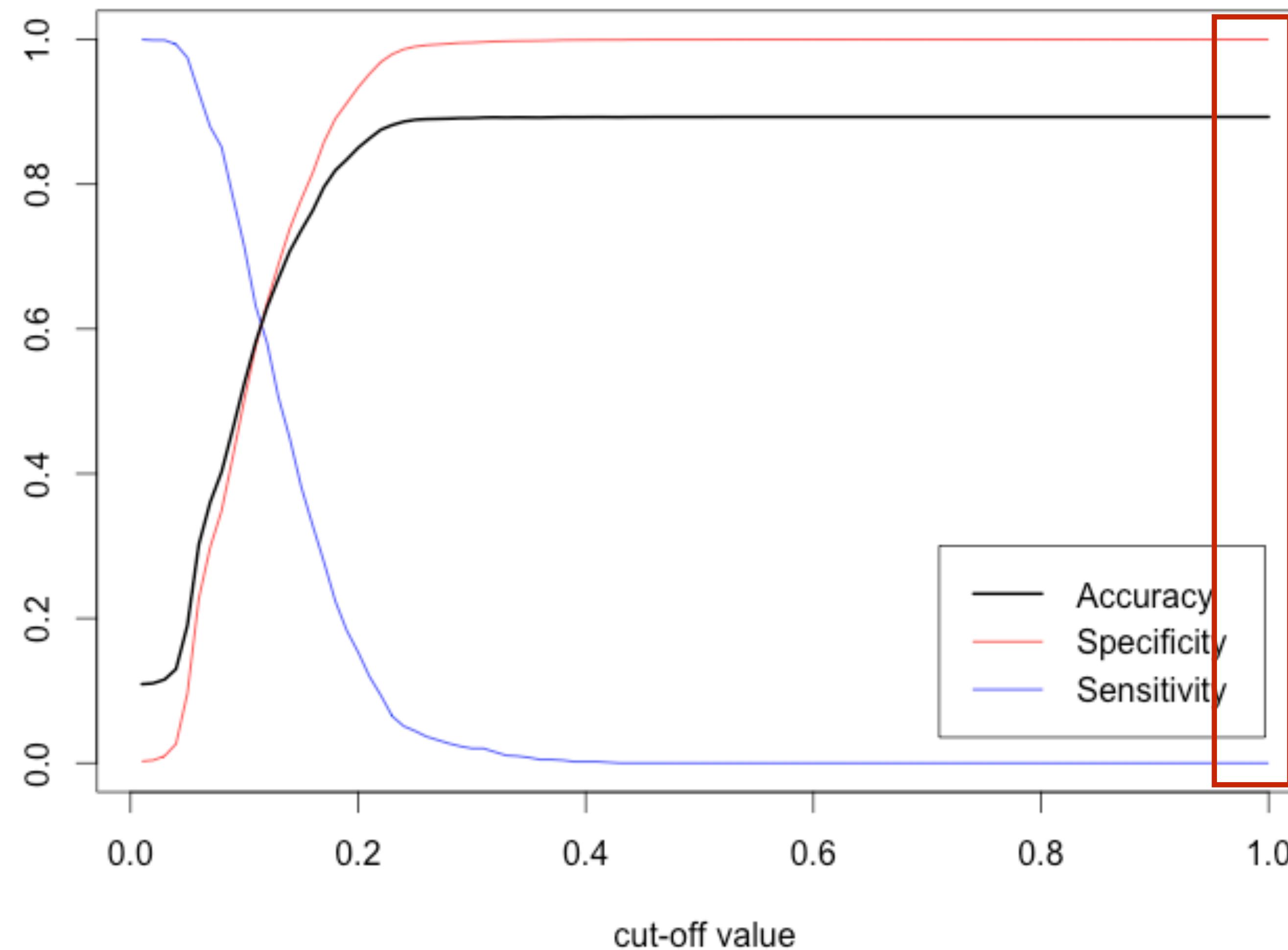
# What about sensitivity or specificity?



Sensitivity =  $1037 / (1037 + 0) = 100\%$

Specificity =  $0 / (0 + 864) = 0\%$

# What about sensitivity or specificity?



Sensitivity =  $0 / (0 + 1037) = 0\%$

Specificity =  $8640 / (8640 + 0) = 100\%$

# About logistic regression...

```
log_model_full <- glm(loan_status ~ ., family = "binomial", data = training_set)
```

is the same as

```
log_model_full <- glm(loan_status ~ ., family = binomial(link = logit),  
data = training_set)
```

recall

$$P(\text{loan\_status} = 1 \mid x_1, \dots, x_m) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_m x_m)}}$$

# Other logistic regression models

```
log_model_full <- glm(loan_status ~ ., family = binomial(link = probit),  
data = training_set)
```

```
log_model_full <- glm(loan_status ~ ., family = binomial(link = cloglog),  
data = training_set)
```

$\beta_j < 0 \rightarrow$  The probability of default decreases as  $x_j$  increases

$\beta_j > 0 \rightarrow$  The probability of default decreases as  $x_j$  increases

BUT

$$P(\text{loan\_status} = 1 | x_1, \dots, x_m) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_m x_m)}}$$



CREDIT RISK MODELING IN R

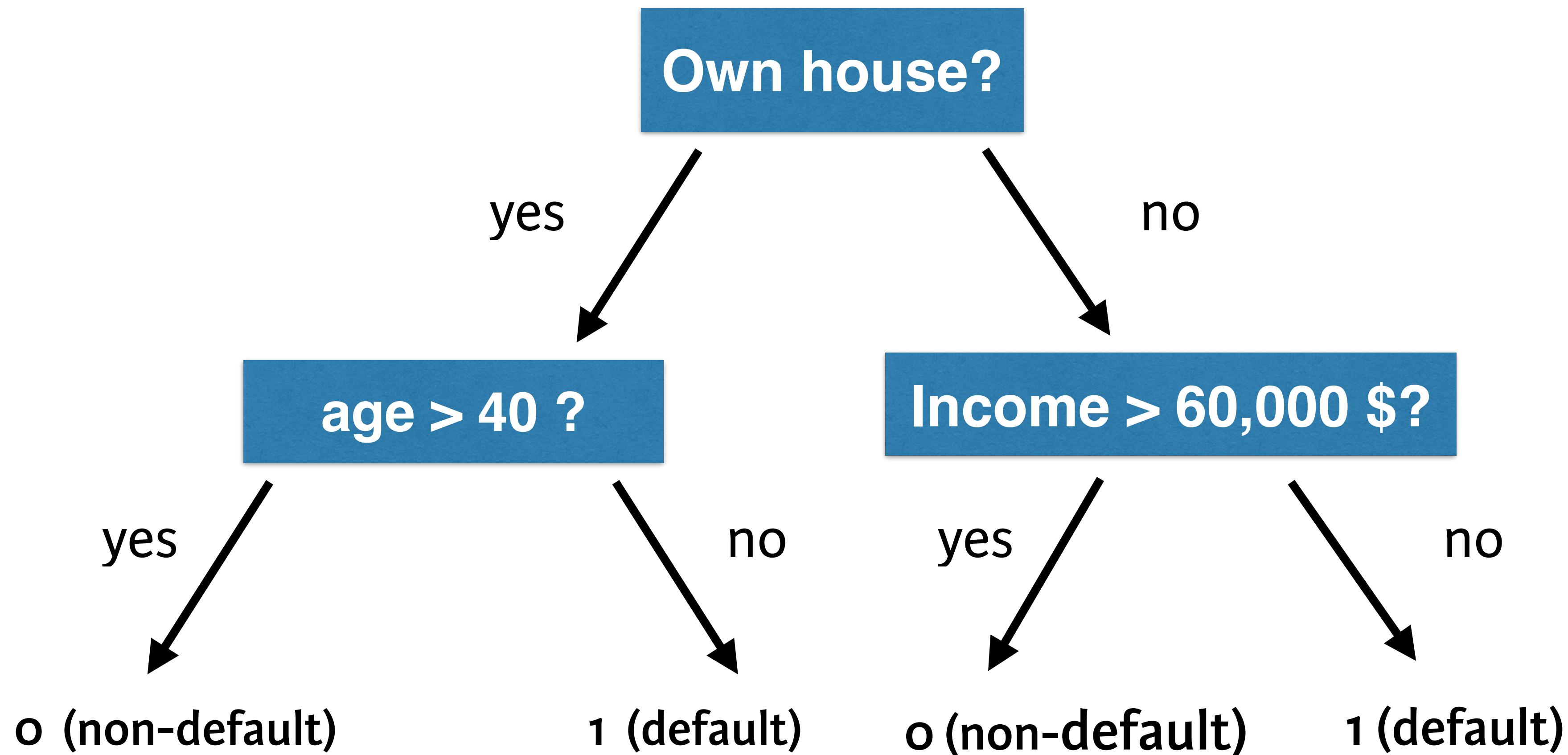
**Let's practice!**



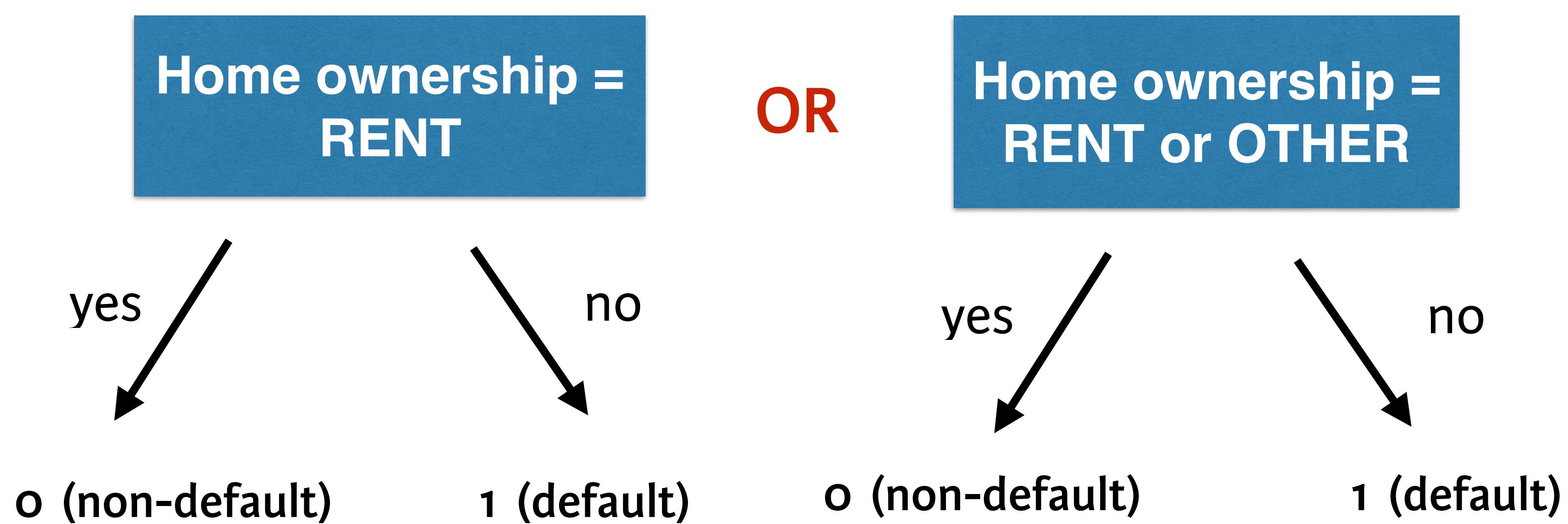
CREDIT RISK MODELING IN R

# What is a decision tree?

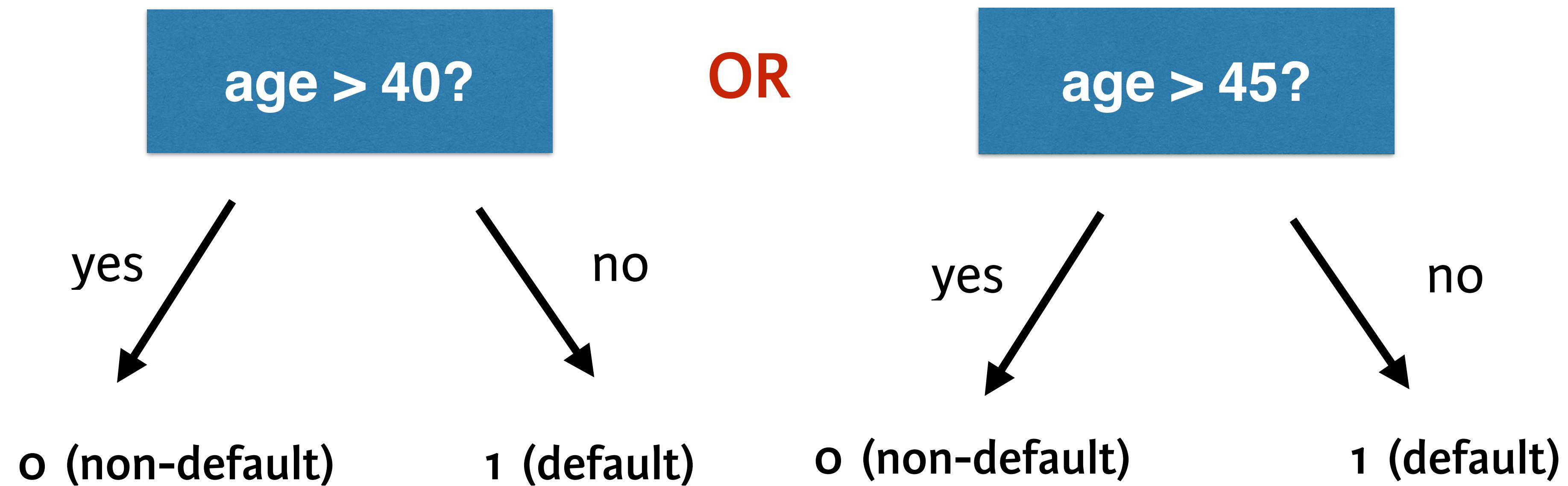
# Decision tree example



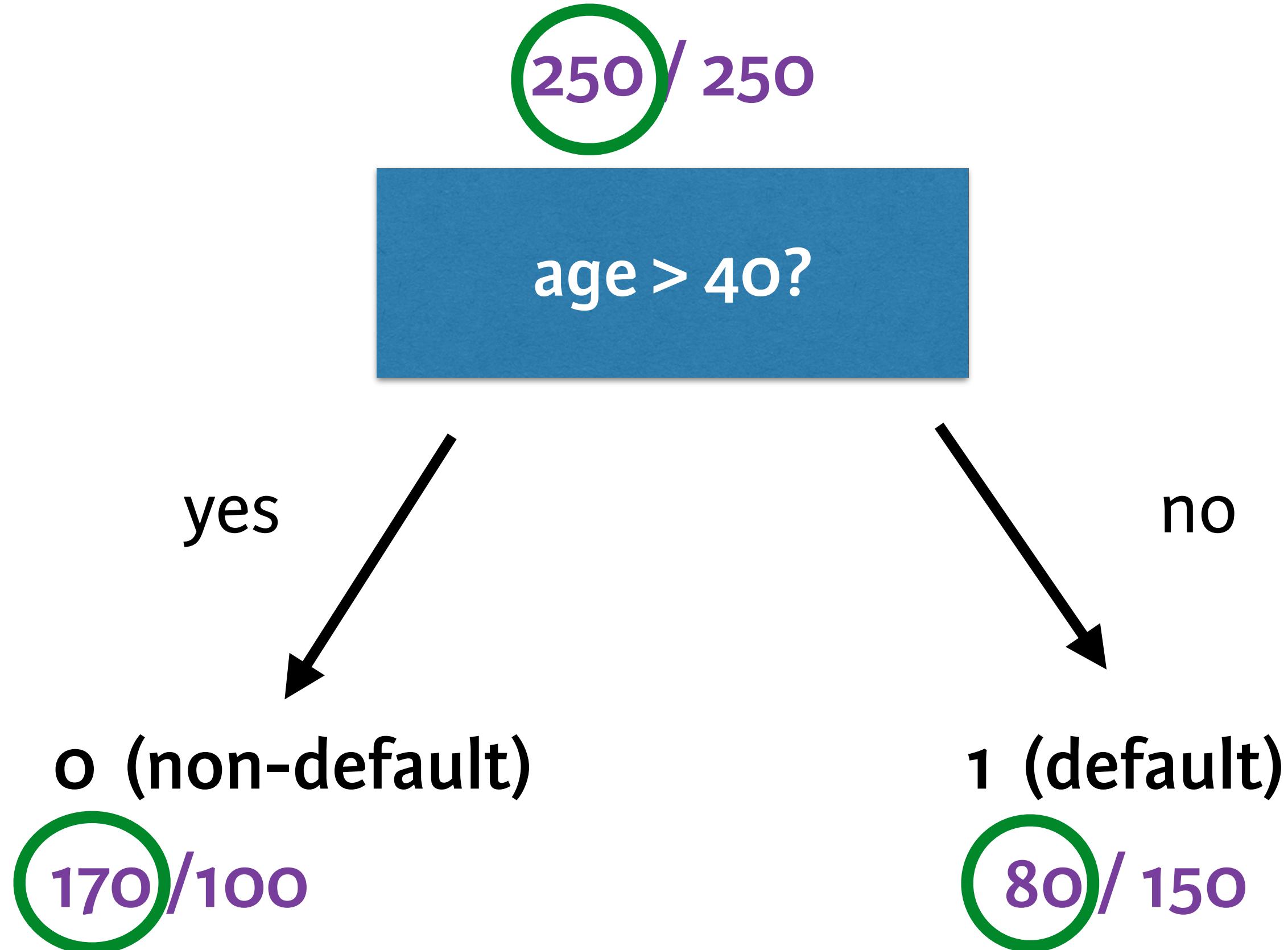
# How to make splitting decision?



# How to make splitting decision?

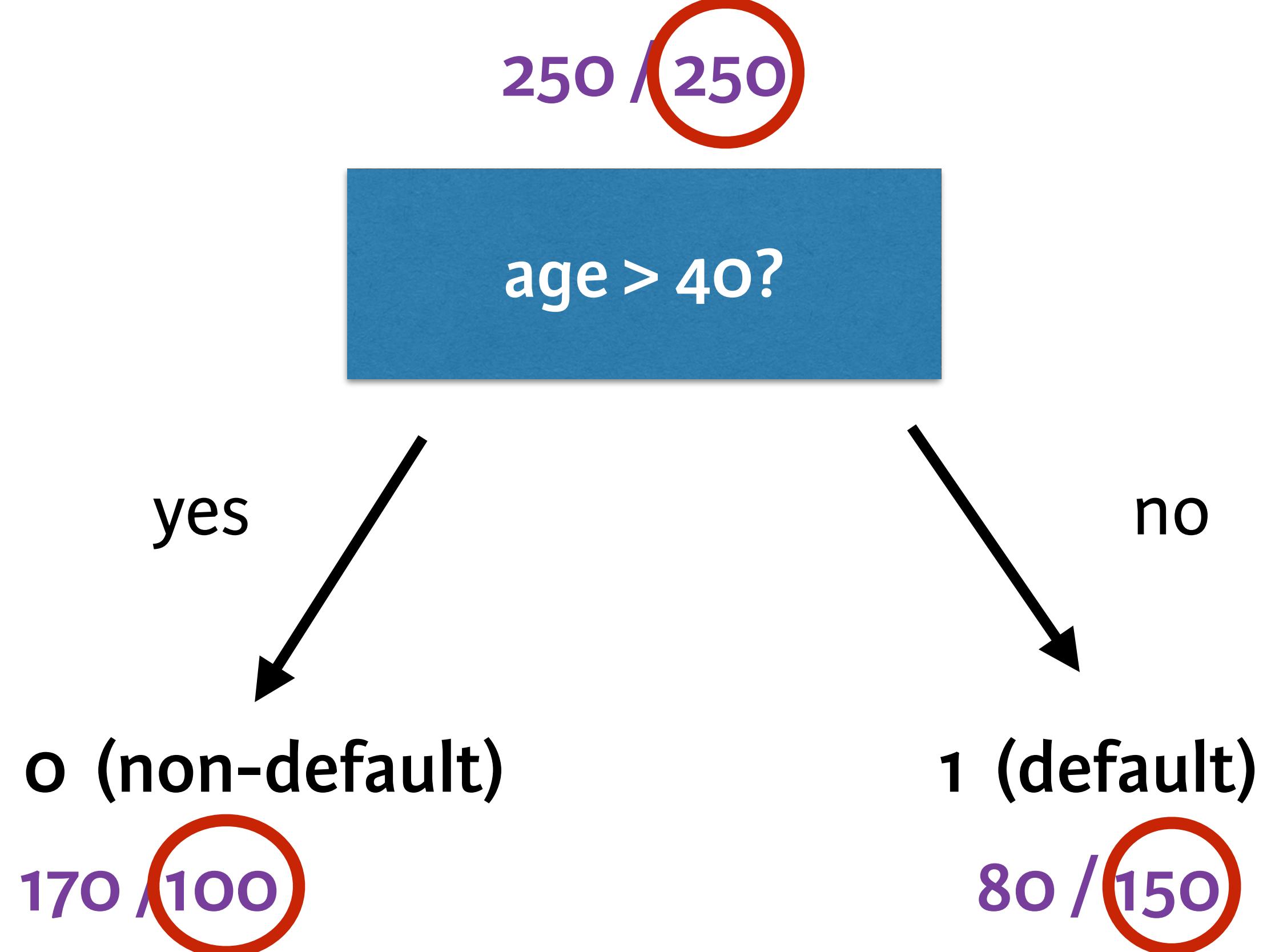


# Example



Actual non-defaults in this node using this split

# Example



Actual defaults in this node  
using this split

# Example

250 / 250

= IDEAL SCENARIO

age > 40?

yes

0 (non-default)

~~170 / 100~~

250/0

no

1 (default)

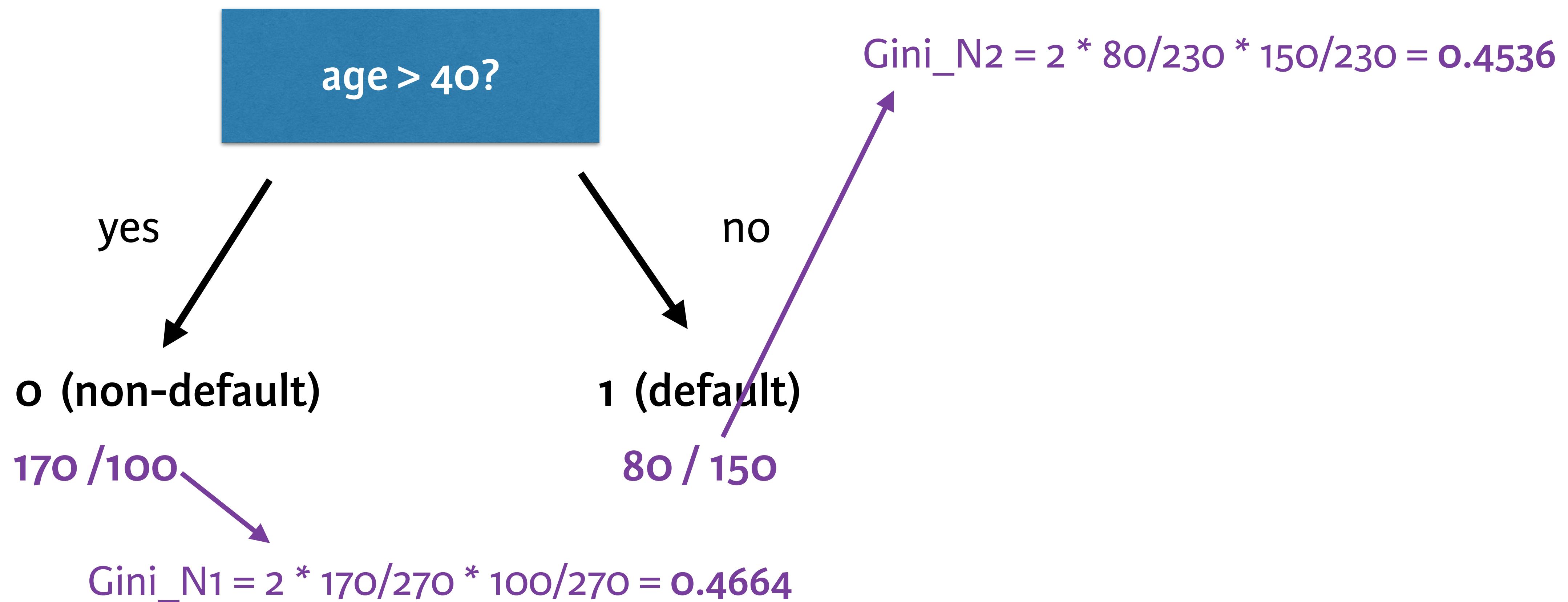
~~80 / 150~~

0/250

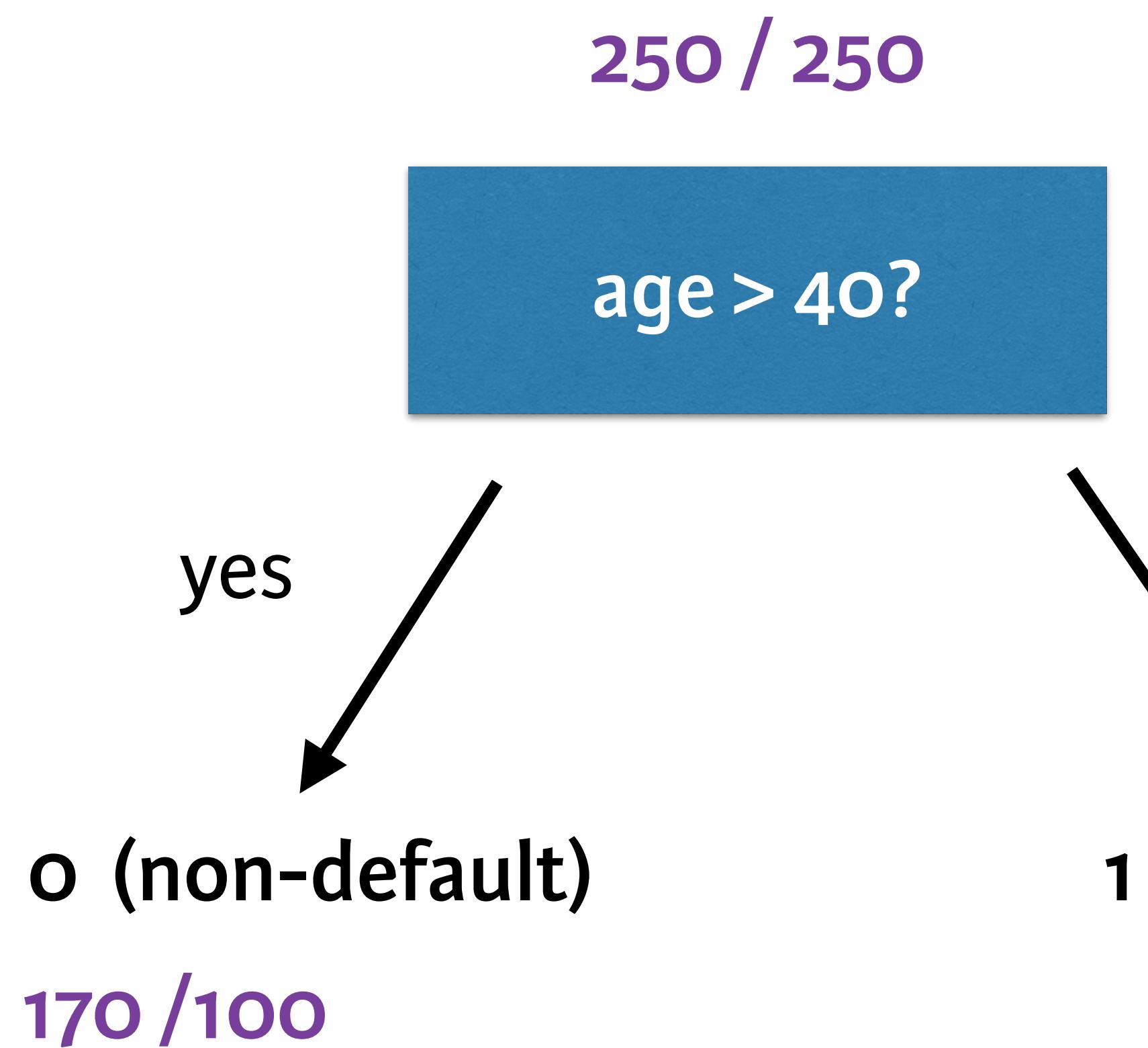
# Example

$$\text{Gini} = 2 * \text{prop(default)} * \text{prop(non-default)}$$

$$250 / 250 \longrightarrow \text{Gini}_R = 2 * 250/500 * 250/500 = 0.5$$



# Example



$$\text{Gain} = \text{Gini}_R - \text{prop}(\text{cases in } N_1) * \text{Gini}_{N_1}$$

$$- \text{prop}(\text{cases in } N_2) * \text{Gini}_{N_1}$$

**MAXIMIZE GAIN**

$$= 0.5 - 270/500 * 0.4664$$

$$- 230/500 * 0.4536$$

$$= \mathbf{0.039488}$$



CREDIT RISK MODELING IN R

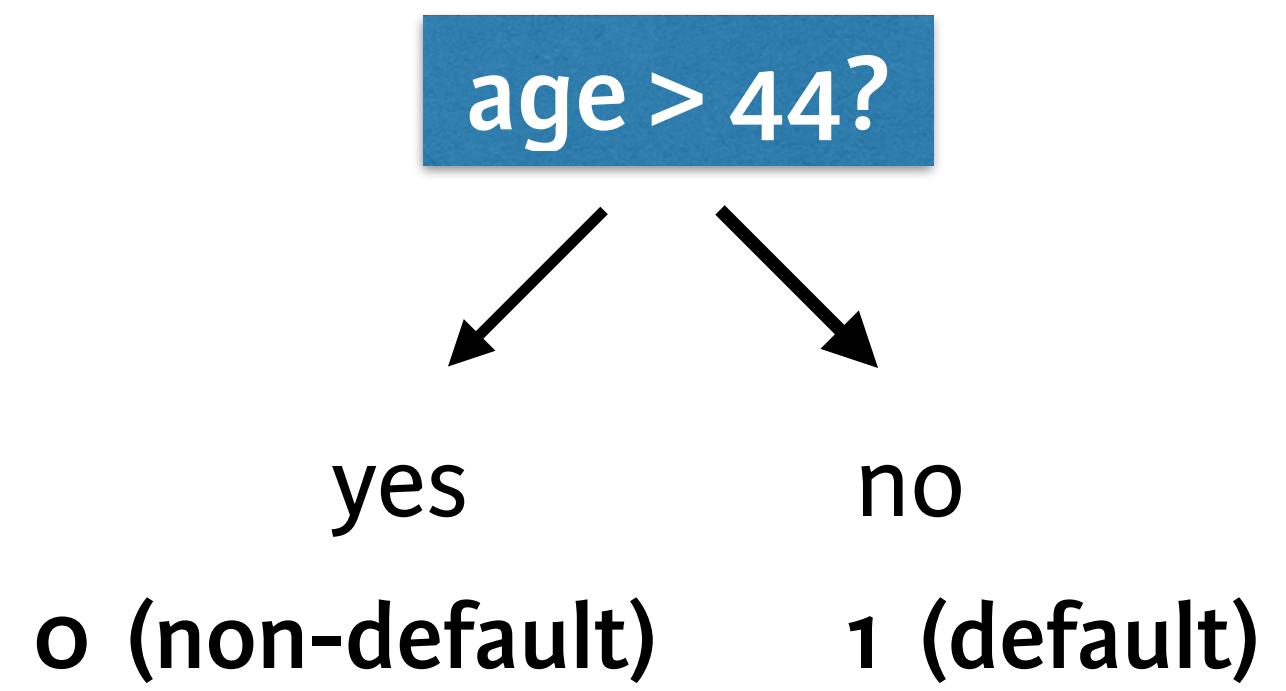
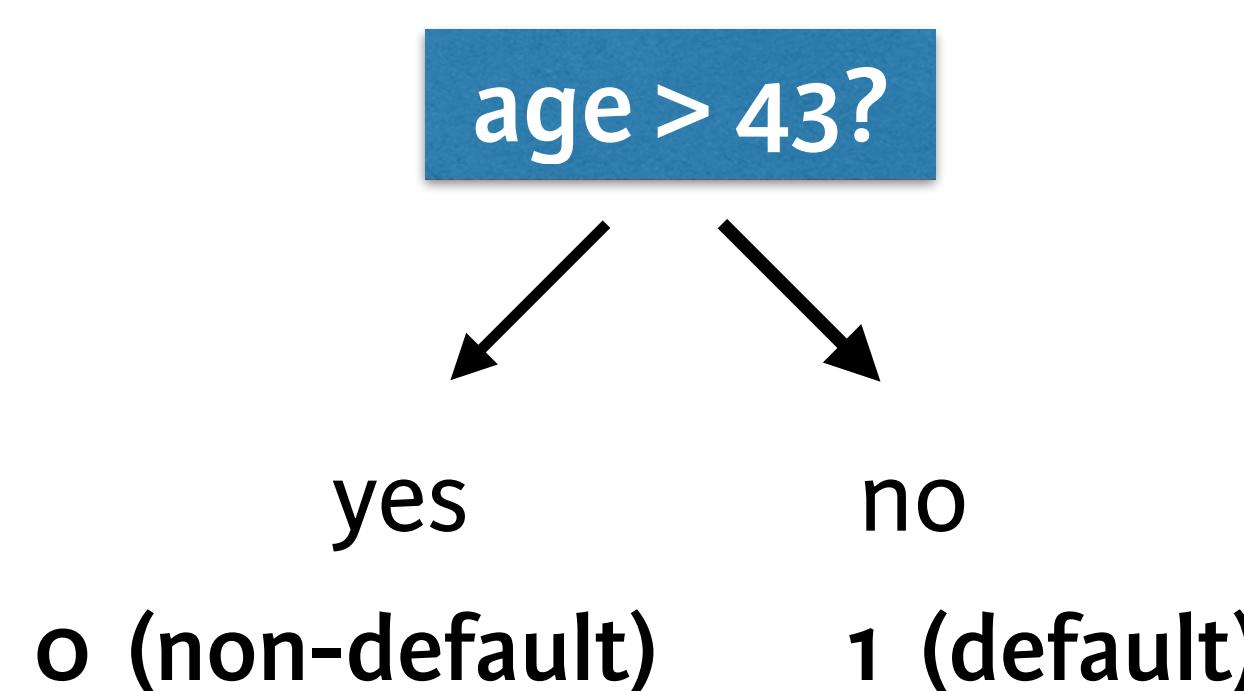
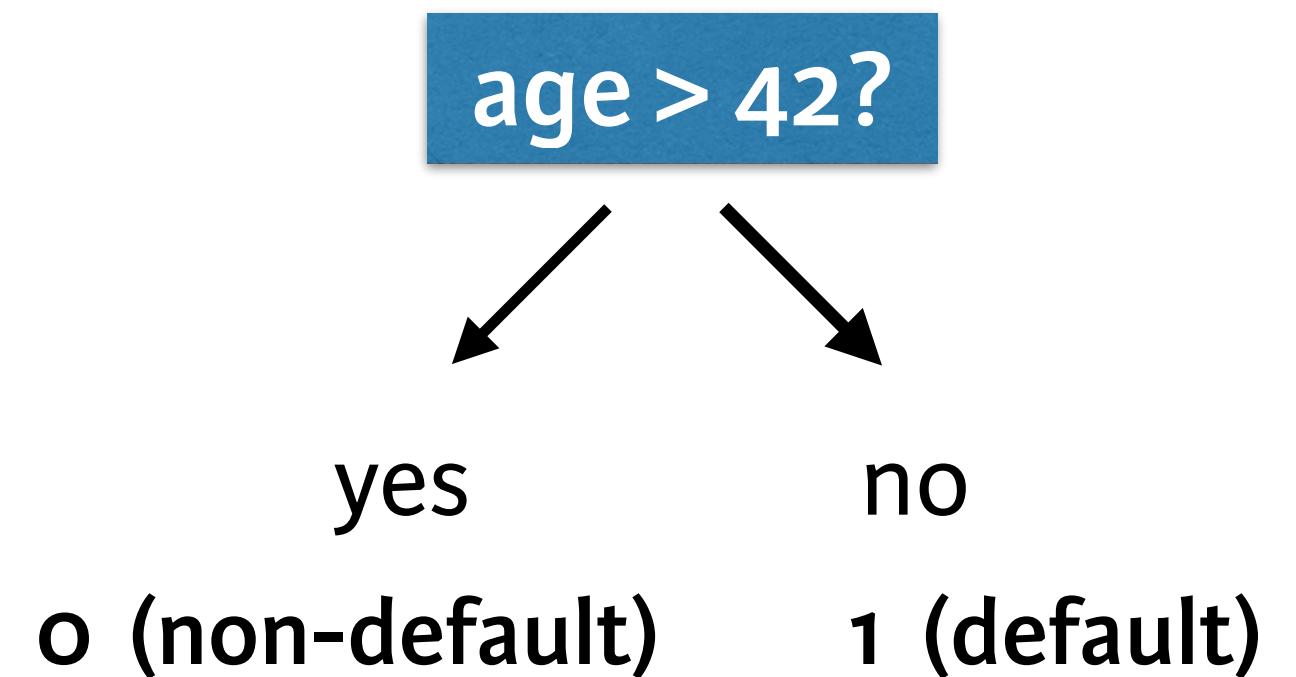
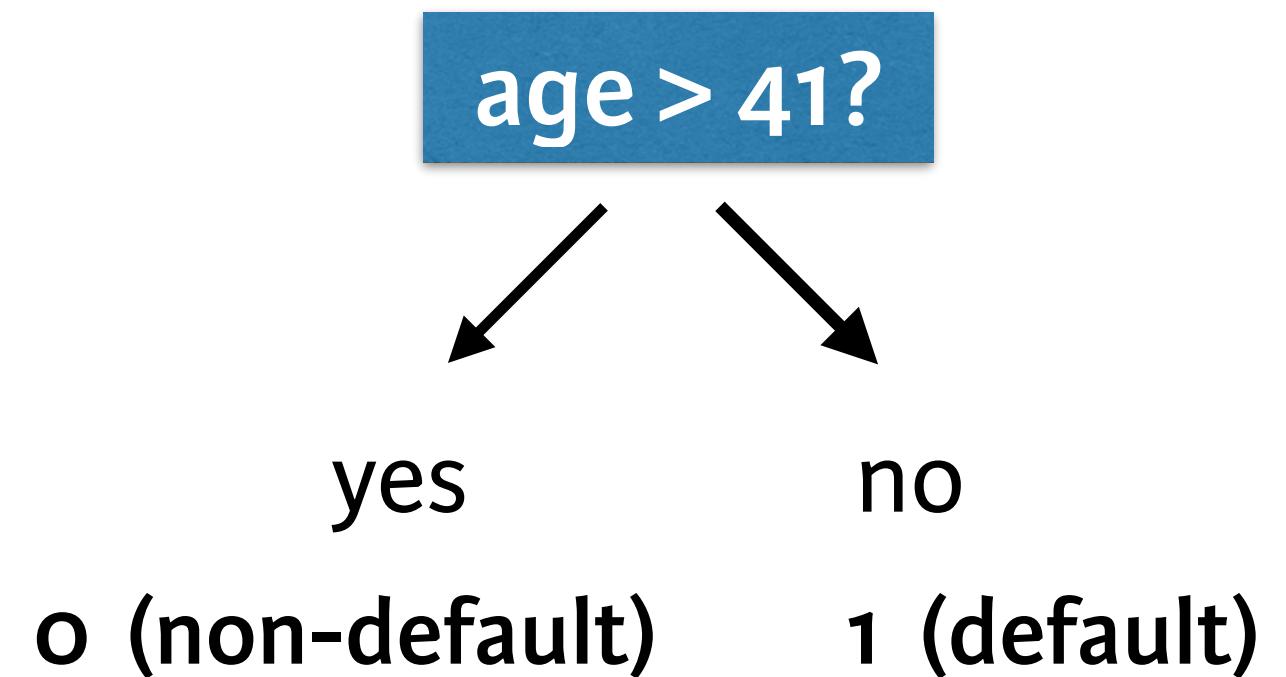
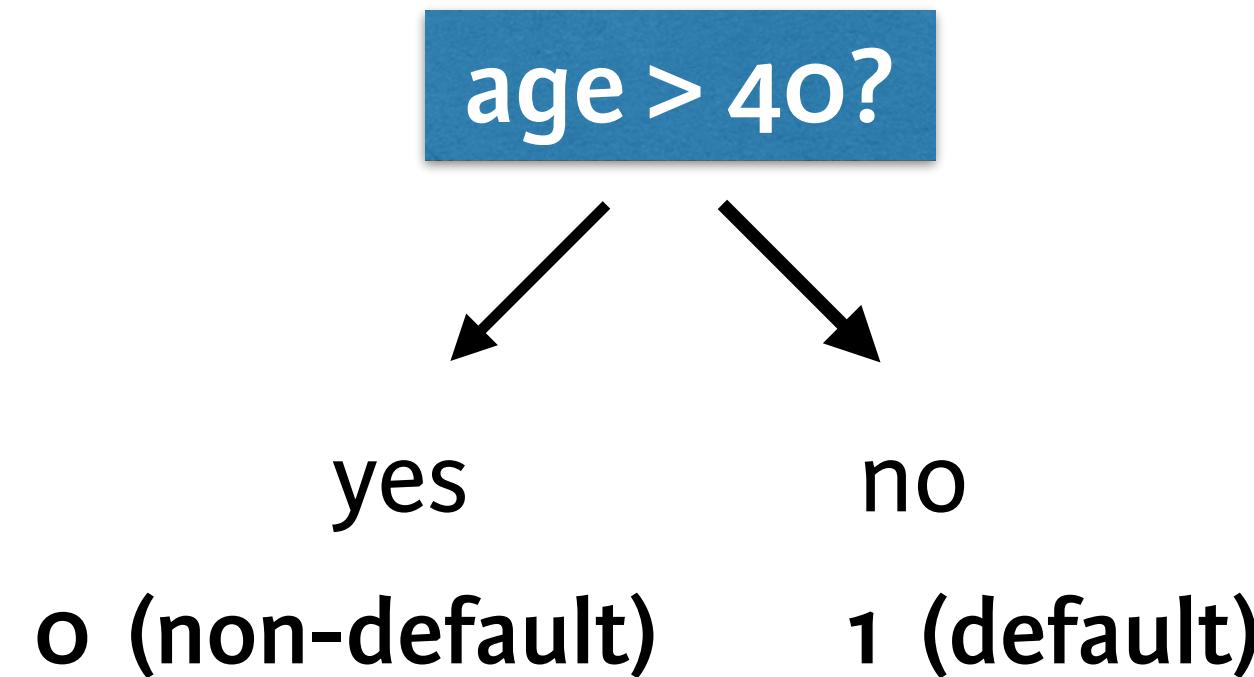
**Let's practice!**



CREDIT RISK MODELING IN R

# Building decision trees using the rpart()-package

# Imagine...



...

# rpart() package! But...

- hard building nice decision tree for credit risk data
- main reason: unbalanced data

```
> fit_default <- rpart(loan_status ~ ., method = "class",
  data = training_set)

> plot(fit_default)
Error in plot.rpart(fit_default) : fit is not a tree, just a root
```

# Three techniques to overcome unbalance

- Undersampling or oversampling
  - Accuracy issue will disappear
  - Only training set
- Changing the prior probabilities
- Including a loss matrix

Validate model to see what is best!



CREDIT RISK MODELING IN R

**Let's practice!**



CREDIT RISK MODELING IN R

# Pruning the decision tree

# Problems with large decision trees

- Too complex: not clear anymore
- Overfitting when applying to test set
- Solution: use printcp(), plotcp() for pruning purposes

# Printcp and tree\_undersample

```
> printcp(tree_undersample)

Classification tree:
rpart(formula = loan_status ~ ., data = undersampled_training_set, method = "class",
control = rpart.control(cp = 0.001))

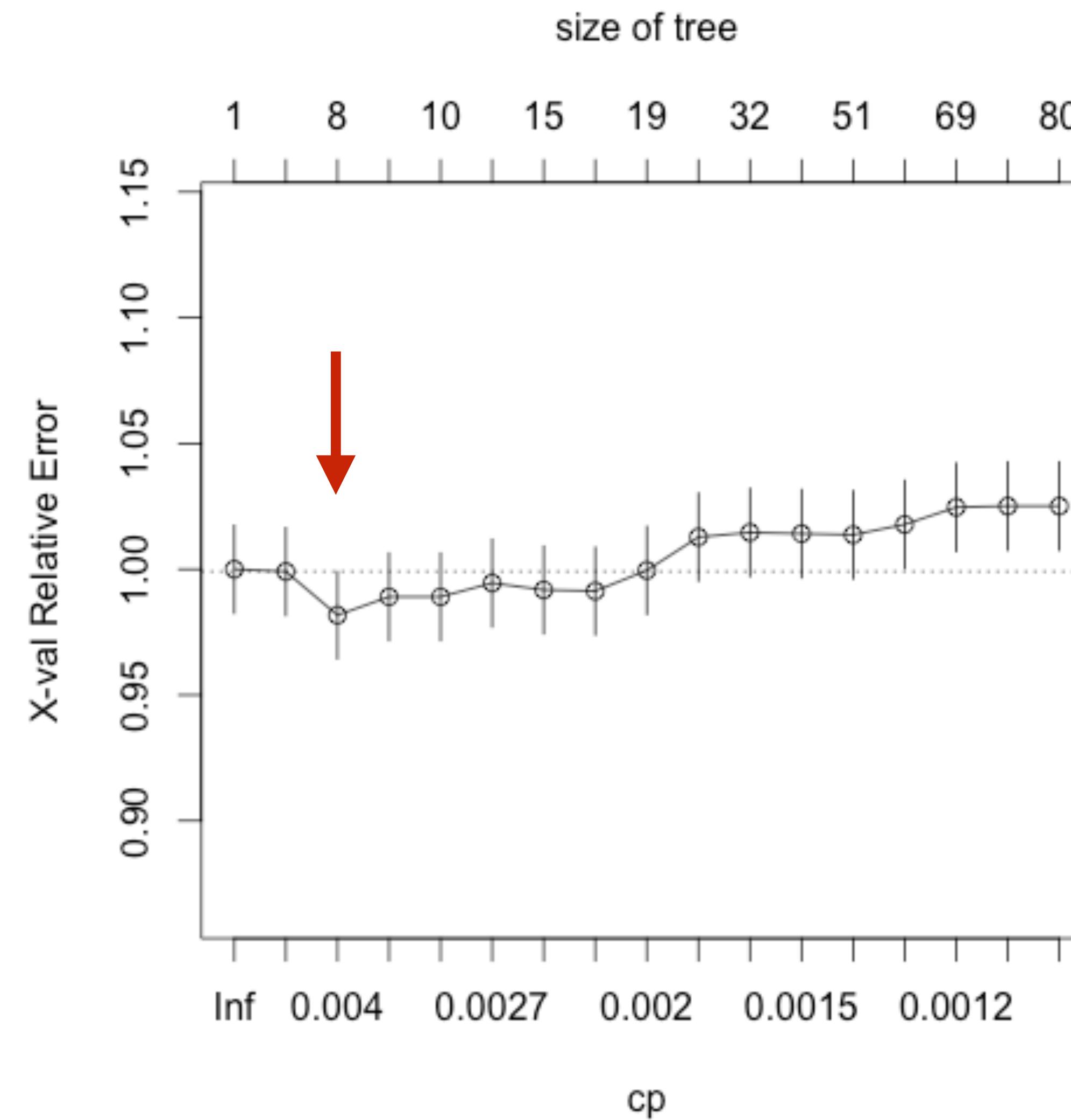
Variables actually used in tree construction:
[1] age      annual_inc    emp_cat     grade      home_ownership  ir_cat     loan_amnt

Root node error: 2190/6570 = 0.33333

n= 6570

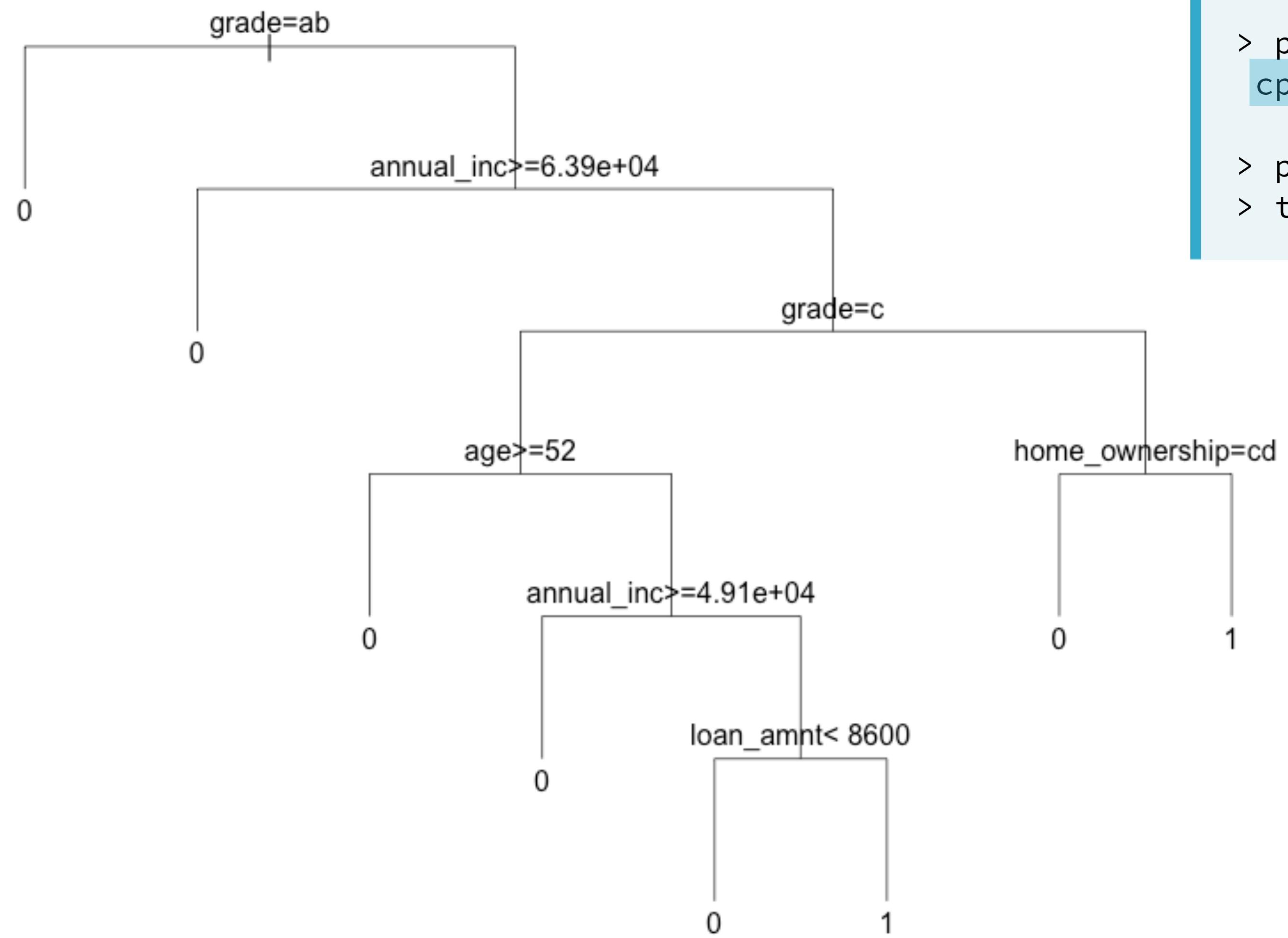
          CP    nsplit rel_error  xerror      xstd
1 0.0059361      0   1.00000  1.00000  0.017447
2 0.0044140      4   0.97443  0.99909  0.017443
3 0.0036530      7   0.96119  0.98174  0.017366
4 0.0031963      8   0.95753  0.98904  0.017399
                               ...
16 0.0010654     76   0.84247  1.02511  0.017554
17 0.0010000     79   0.83927  1.02511  0.017554
```

# Plotcp and tree\_undersample



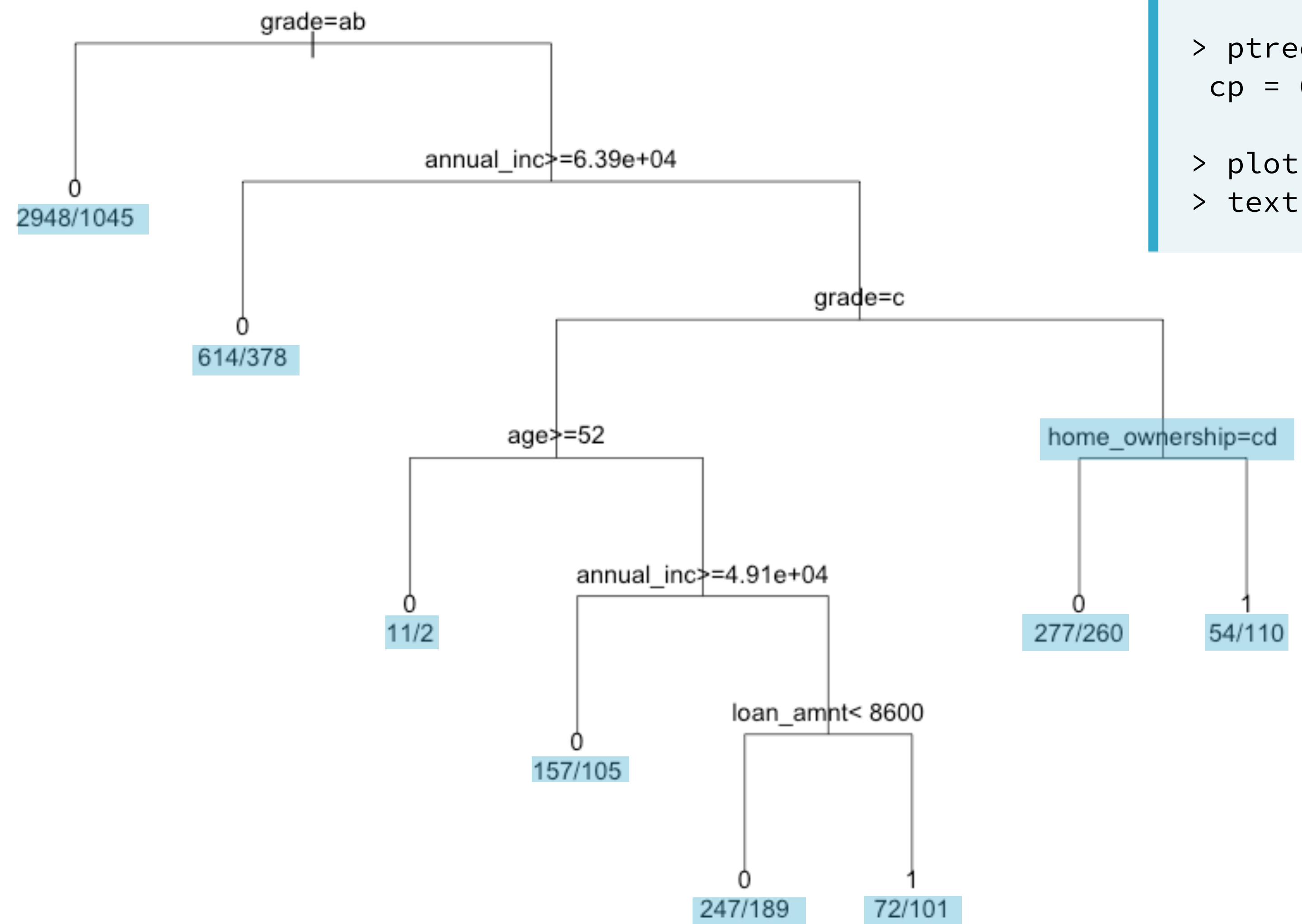
cp= 0.003653

# plot the pruned tree



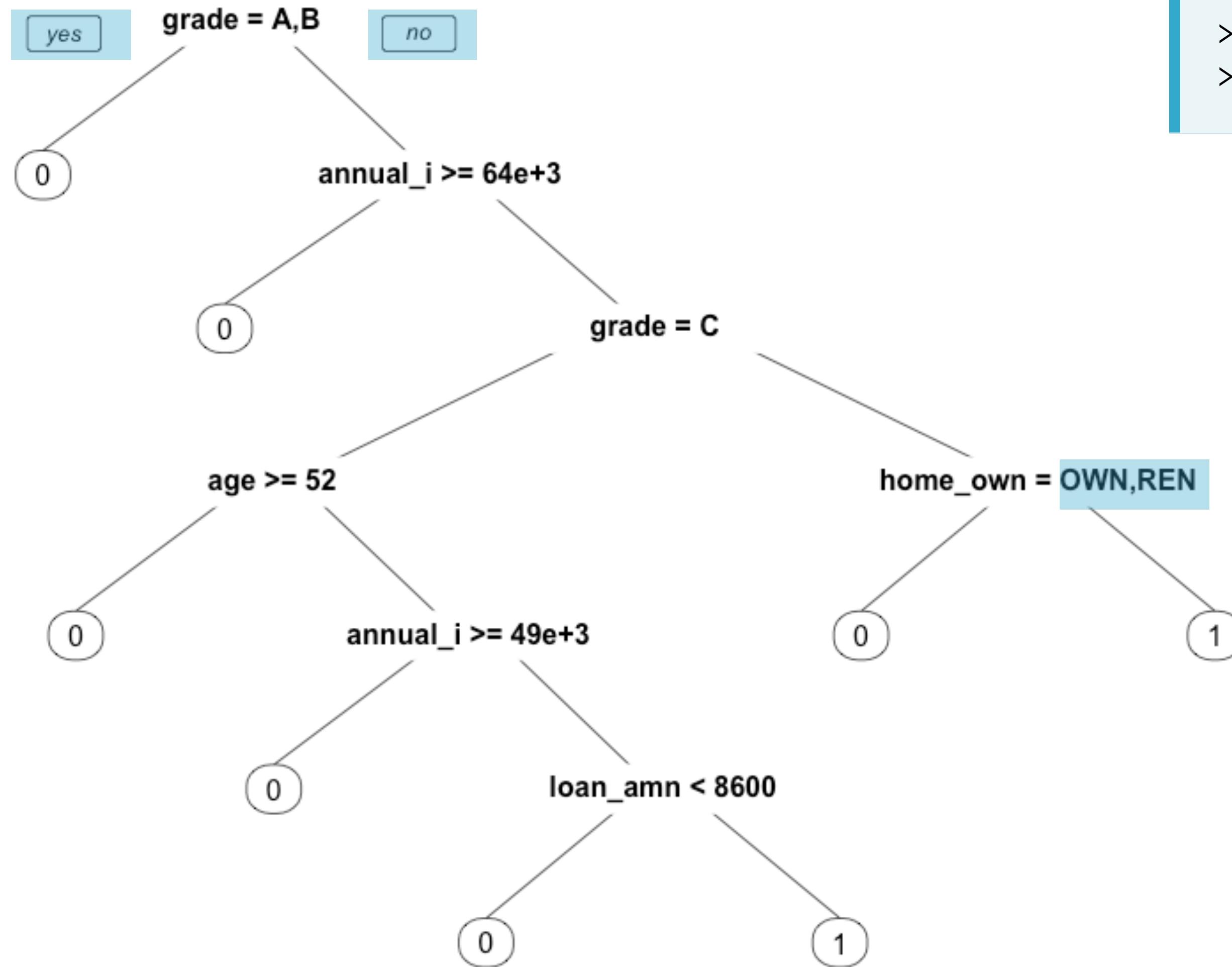
```
> ptree_undersample=prune(tree_undersample,  
  cp = 0.003653)  
  
> plot(ptree_undersample, uniform=TRUE)  
> text(ptree_undersample)
```

# plot the pruned tree



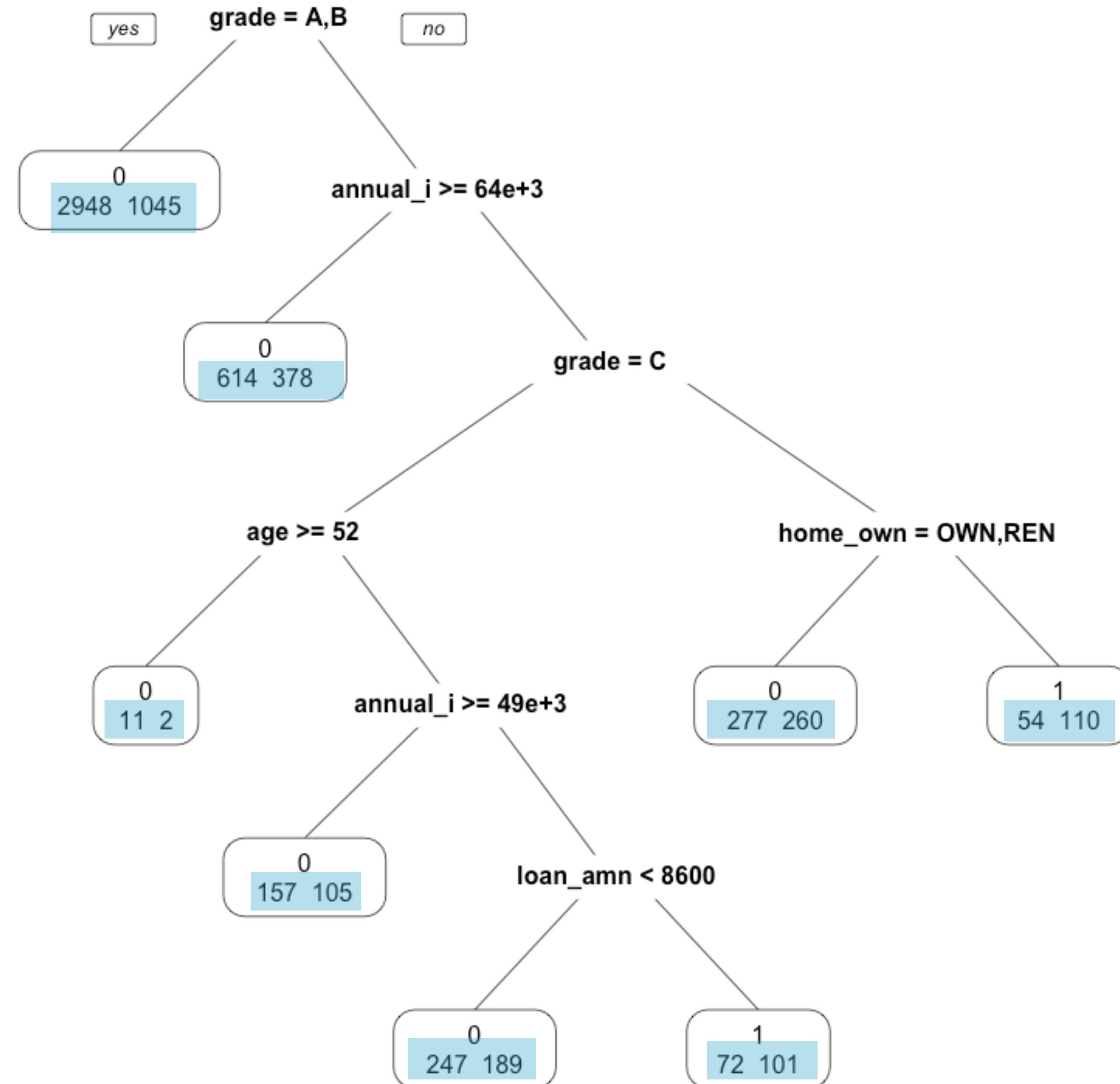
```
> ptree_undersample=prune(tree_undersample,  
  cp = 0.003653)  
  
> plot(ptree_undersample, uniform=TRUE)  
> text(ptree_undersample, use.n=TRUE)
```

# prp() in the rpart.plot-package



```
> library(rpart.plot)  
> prp(ptree_undersample)
```

# prp() in the rpart.plot-package



```
> library(rpart.plot)
> prp(ptree_undersample, extra = 1)
```



CREDIT RISK MODELING IN R

**Let's practice!**



CREDIT RISK MODELING IN R

**Other tree options and  
the construction of confusion matrices.**

# Other interesting rpart()-arguments

...in `rpart()`

- `weights`: include case weights

...in the control argument of `rpart` (`rpart.control`)

- `minsplit`: minimum number of observations for split attempt
- `minbucket`: minimum number of observations in leaf node

# Making predictions using the decision tree

```
> pred_undersample_class = predict(ptree_undersample, newdata = test_set,  
type =“class”)
```

1	2	3	...	29073	29079	29084	29090	29091
0	0	0	...	1	0	0	0	0

OR

```
> pred_undersample = predict(ptree_undersample, newdata = test_set)
```

	0	1
1	0.7382920	0.2617080
2	0.5665138	0.4334862
3	0.5992366	0.4007634
	...	...
29073	0.4161850	0.5838150
29079	0.6189516	0.3810484
29084	0.7382920	0.2617080
29090	0.7382920	0.2617080
29091	0.7382920	0.2617080

# Constructing a confusion matrix

```
> table(test_set$loan_status, pred_undersample_class)

pred_undersample_class
  0    1
0 8314  346
1  964   73
```



CREDIT RISK MODELING IN R

**Let's practice!**



CREDIT RISK MODELING IN R

# Finding the right cut-off: the strategy curve

# Constructing a confusion matrix

```
> predict(log_reg_model, newdata = test_set, type = "response")
```

1	2	3	4	5	...
0.08825517	0.3502768	0.28632298	0.1657199	0.11264550	...

```
> predict(class_tree, new data = test_set)
```

0	1
1 0.7873134	0.2126866
2 0.6250000	0.3750000
3 0.6250000	0.3750000
4 0.7873134	0.2126866
5 0.5756867	0.4243133

# Cut-off?

```
> pred_log_regression_model <- predict(log_reg_model, newdata = test_set,  
type = "response")  
  
> cutoff <- 0.14  
  
> class_pred_logit <- ifelse(pred_log_regression_model > cutoff, 1, 0)
```

?

# A certain strategy...

```
> log_model_full <- glm(loan_status ~ ., family = "binomial", data = training_set)
> predictions_all_full<- predict(log_reg_model, newdata = test_set, type = "response")

> cutoff <- quantile(predictions_all_full, 0.8)
cutoff
 80%
0.1600124

> pred_full_20 <- ifelse(predictions_all_full > cutoff, 1, 0)
```

# A certain strategy (continued)

```
> true_and_predval <- cbind(test_set$loan_status, pred_full_20)
true_and_predval
```

	test_set\$loan_status	pred_full_20
1	0	0
2	0	0
3	0	1
4	0	0
5	0	1
...	...	...

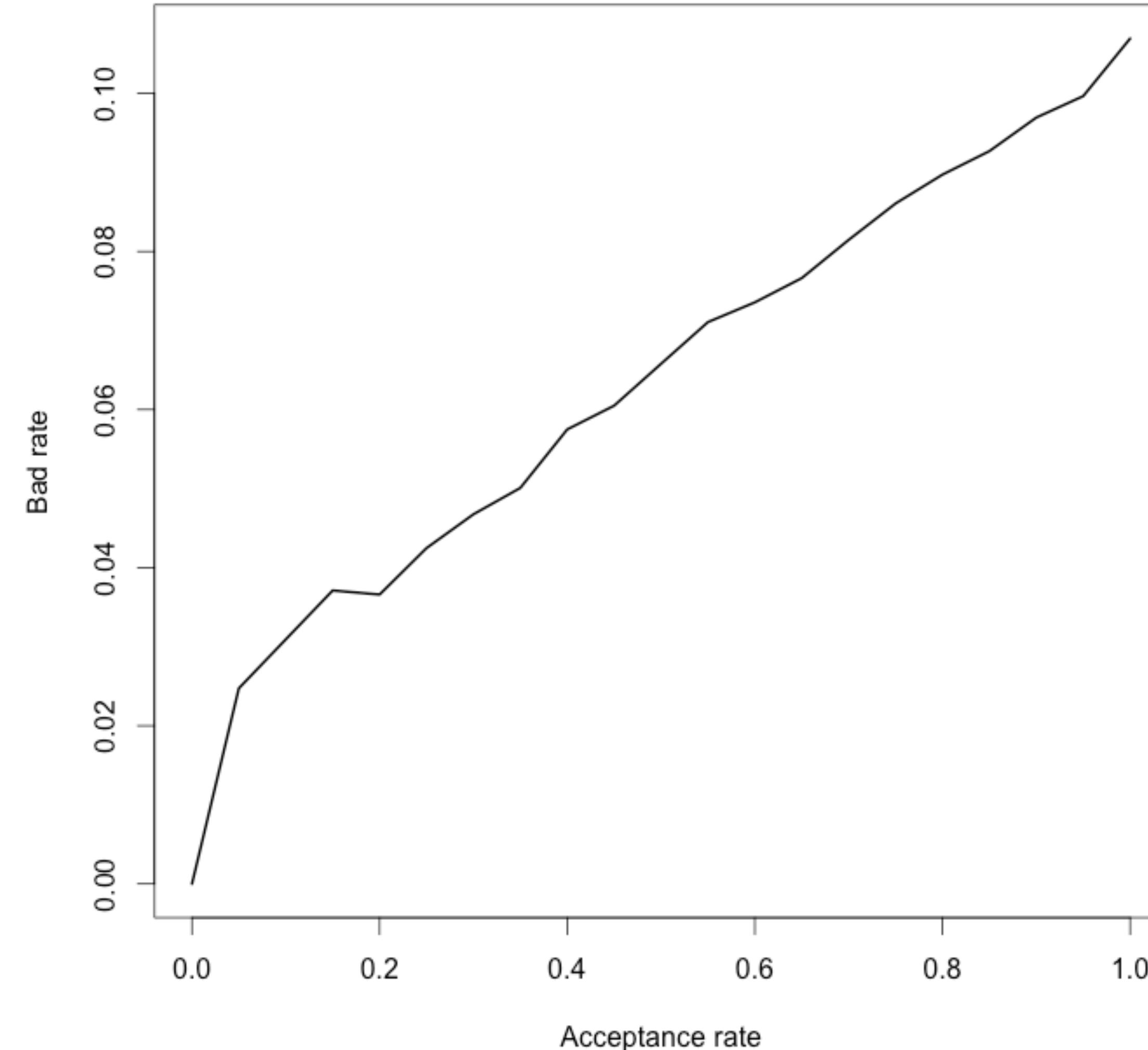
```
> accepted_loans <- pred_and_trueval[pred_full_20 == 0,1]
> bad_rate <- sum(accepted_loans)/length(accepted_loans)
> bad_rate
[1] 0.08972541
```

# The strategy table

	accept_rate	cutoff	bad_rate
[1, ]	1.00	0.5142	0.1069
[2, ]	0.95	0.2122	0.0997
[3, ]	0.90	0.1890	0.0969
[4, ]	0.85	0.1714	0.0927
[5, ]	0.80	0.1600	0.0897
[6, ]	0.75	0.1471	0.0861
[7, ]	0.70	0.1362	0.0815
[8, ]	0.65	0.1268	0.0766
...	...	...	...
[16, ]	0.25	0.0644	0.0425
[17, ]	0.20	0.0590	0.0366
[18, ]	0.15	0.0551	0.0371
[19, ]	0.10	0.0512	0.0309
[20, ]	0.05	0.0453	0.0247
[21, ]	0.00	0.0000	0.0000



# The strategy curve





CREDIT RISK MODELING IN R

**Let's practice!**



CREDIT RISK MODELING IN R

**Let's practice!**



CREDIT RISK MODELING IN R

# The ROC-curve

# Until now

- strategy table/curve : still make assumption
- what is “overall” best model?

# Confusion matrix

model prediction

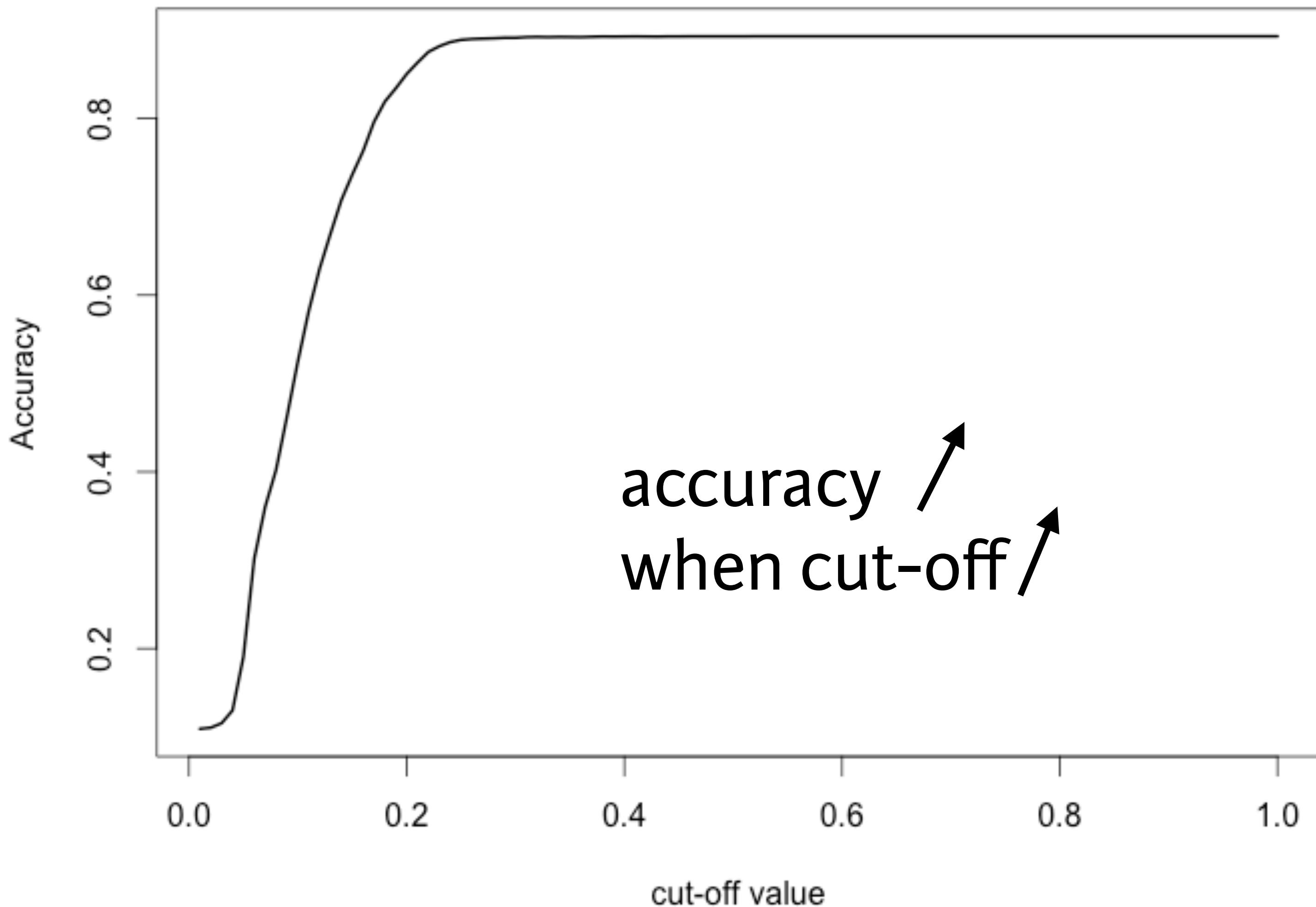
	no default (0)	default (1)
actual loan status	<b>TN</b>	<b>FP</b>
no default (0)	<b>FN</b>	<b>TP</b>
default (1)		

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

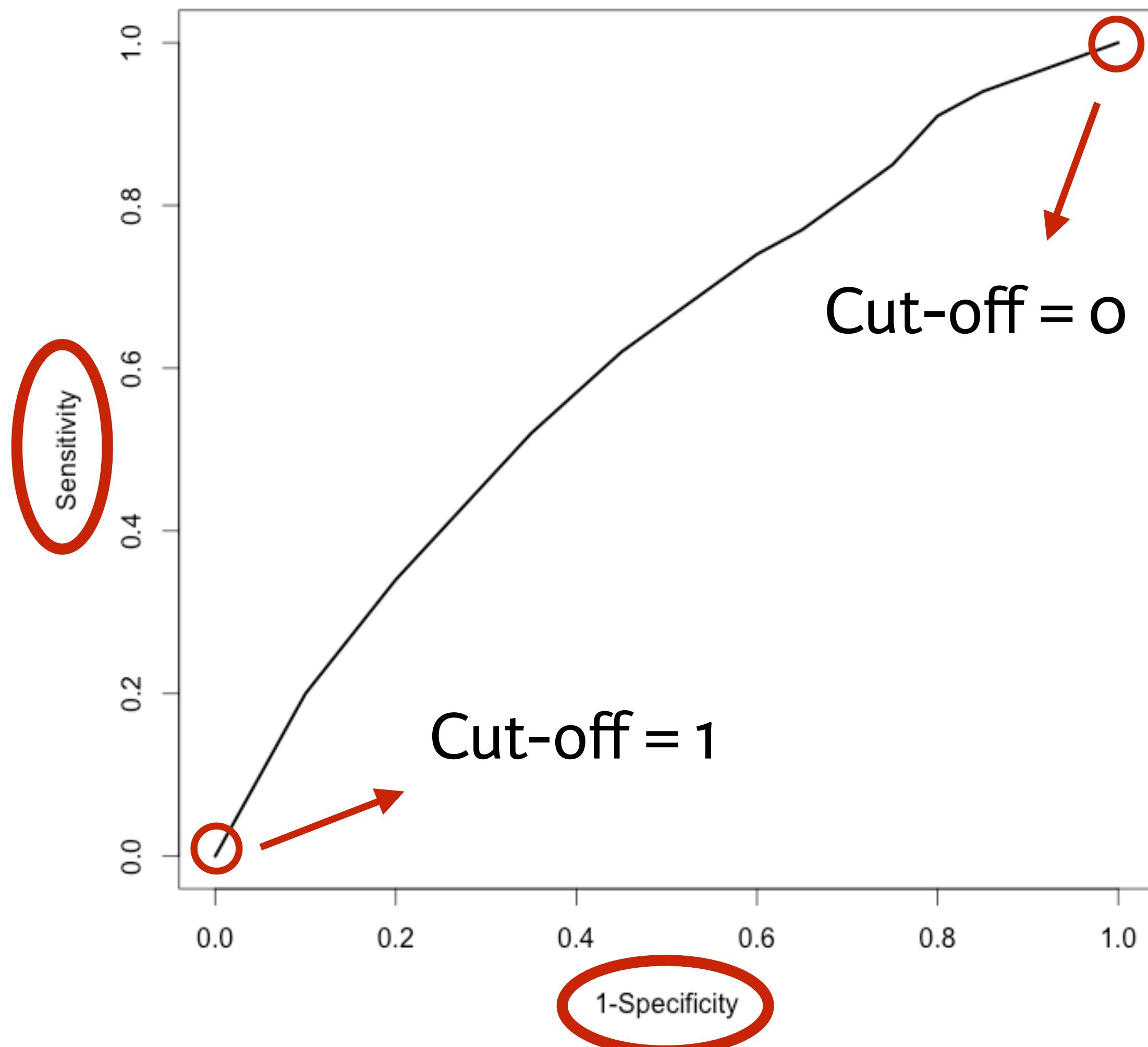
# Accuracy?



$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

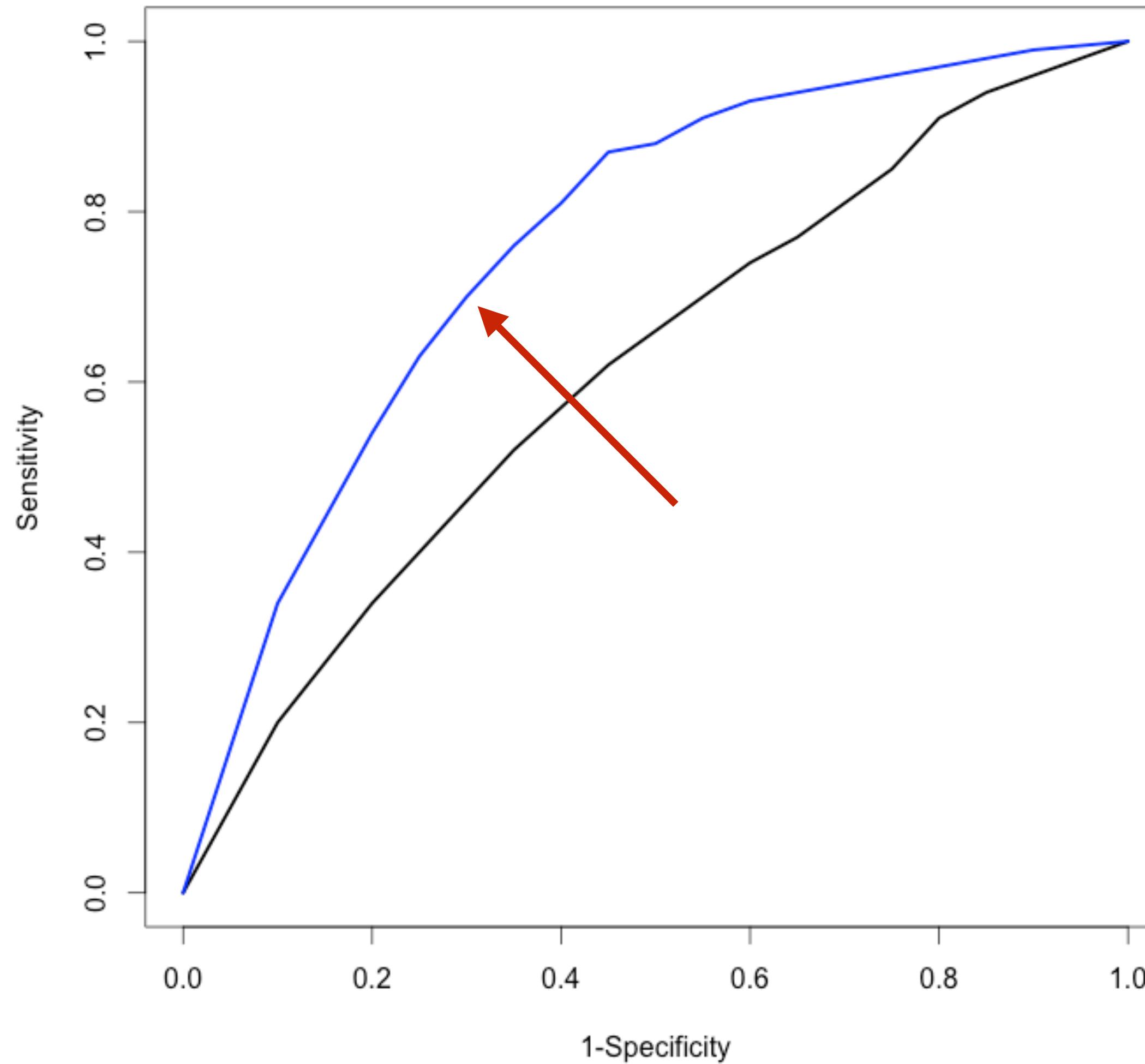
# The ROC-curve



$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

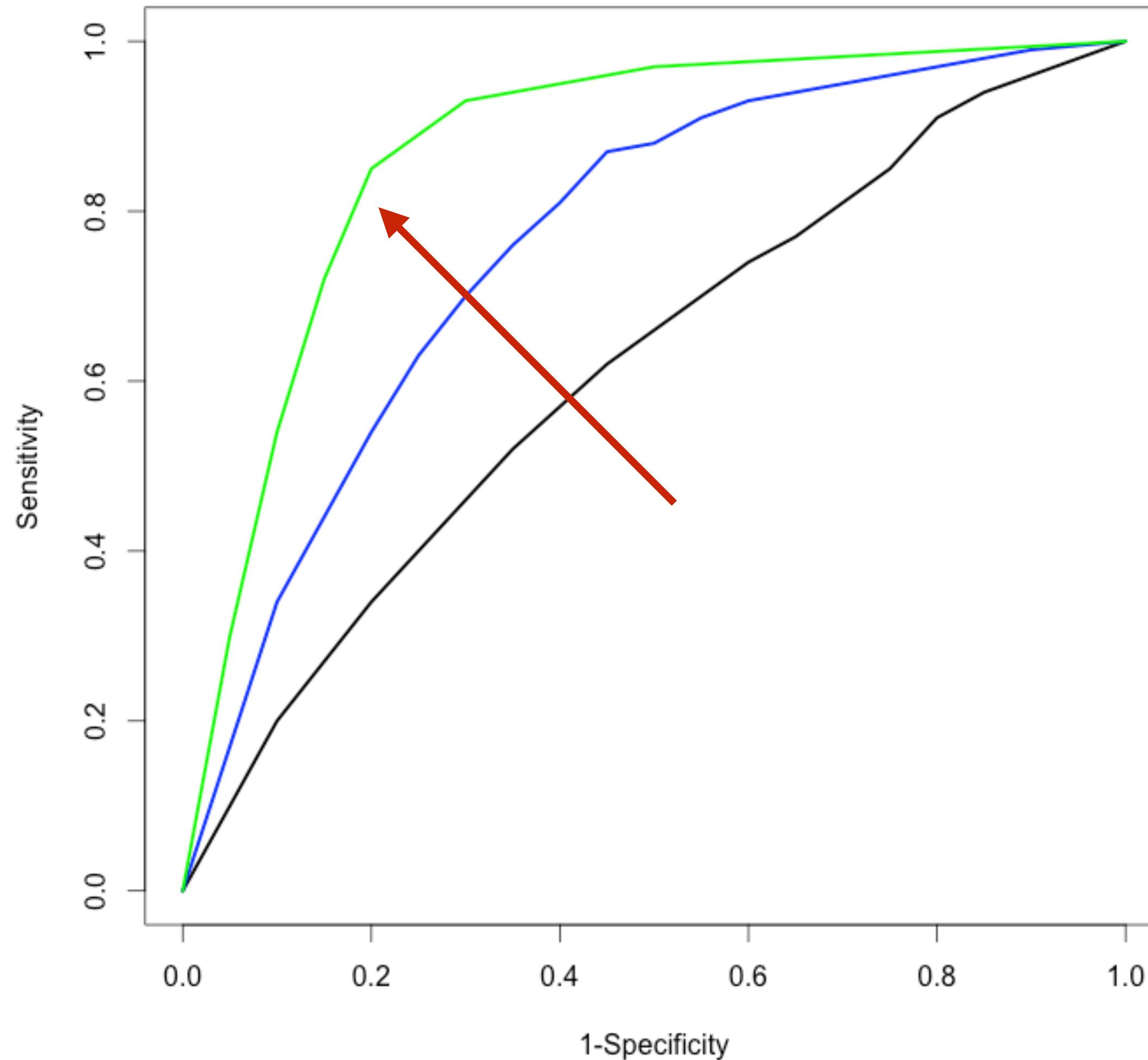
# The ROC-curve



$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

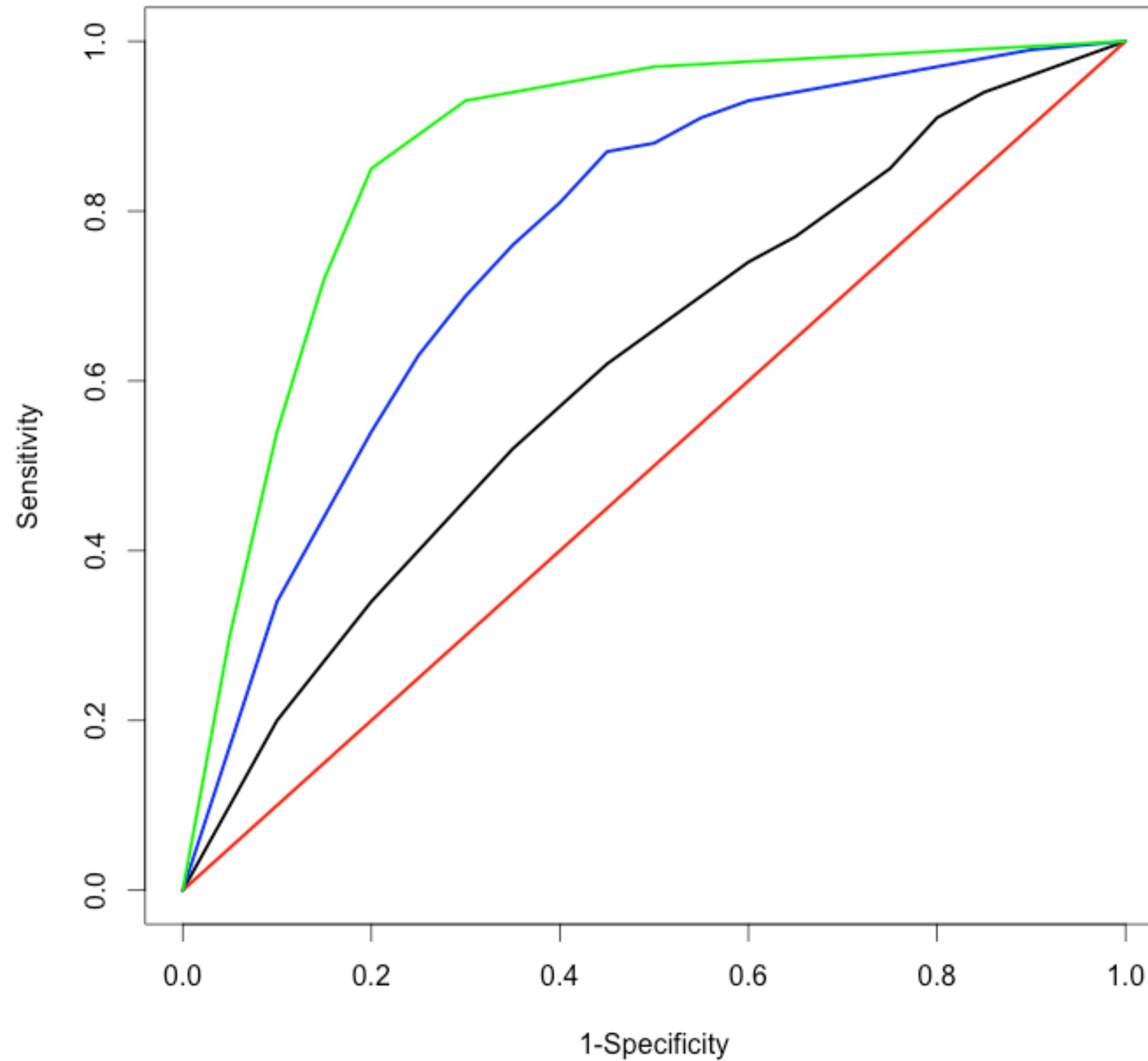
# The ROC-curve



$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

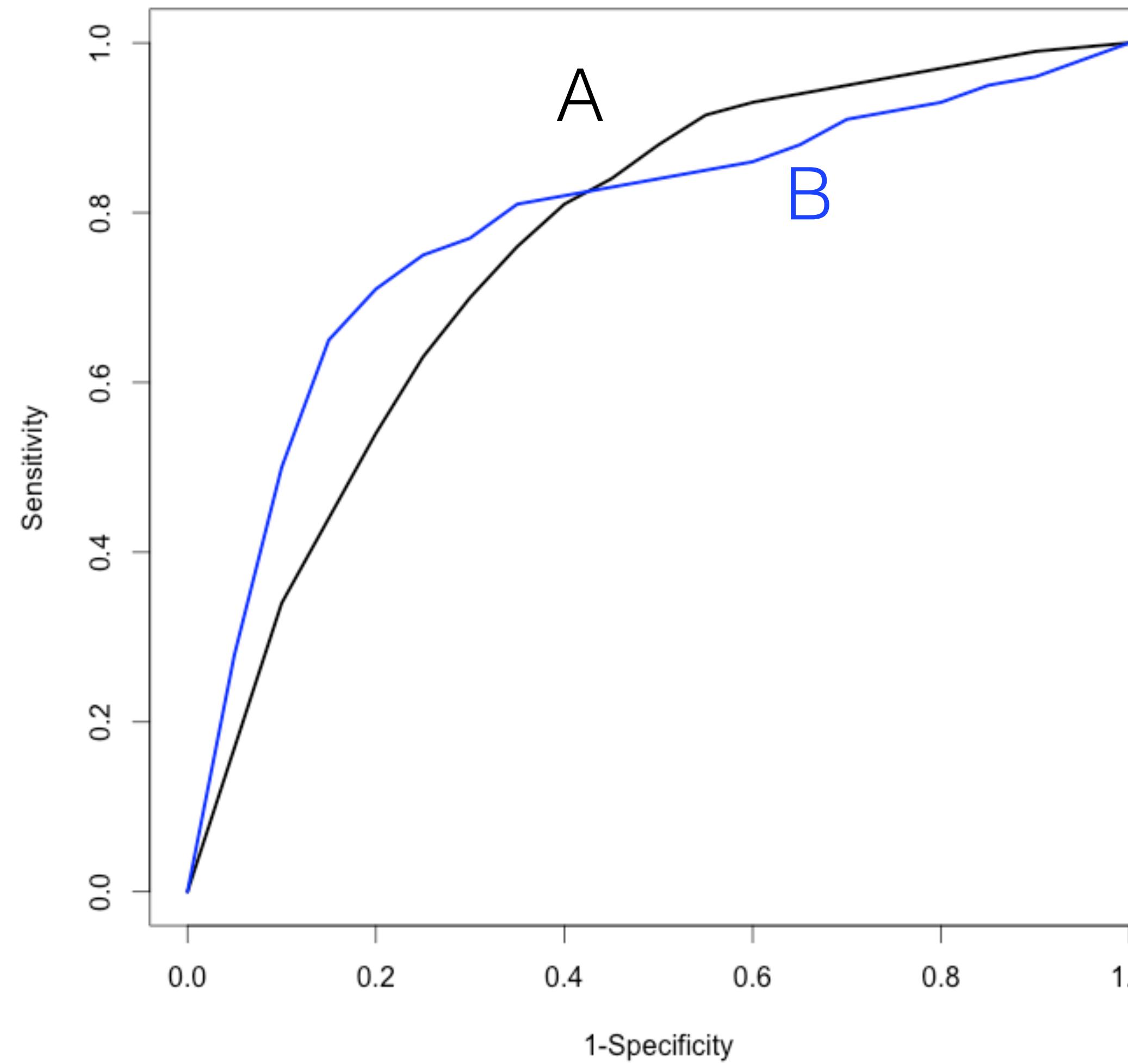
# The ROC-curve



$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

# Which one is better?



AUC ROC-curve A = 0.75

AUC ROC-curve B = 0.78



CREDIT RISK MODELING IN R

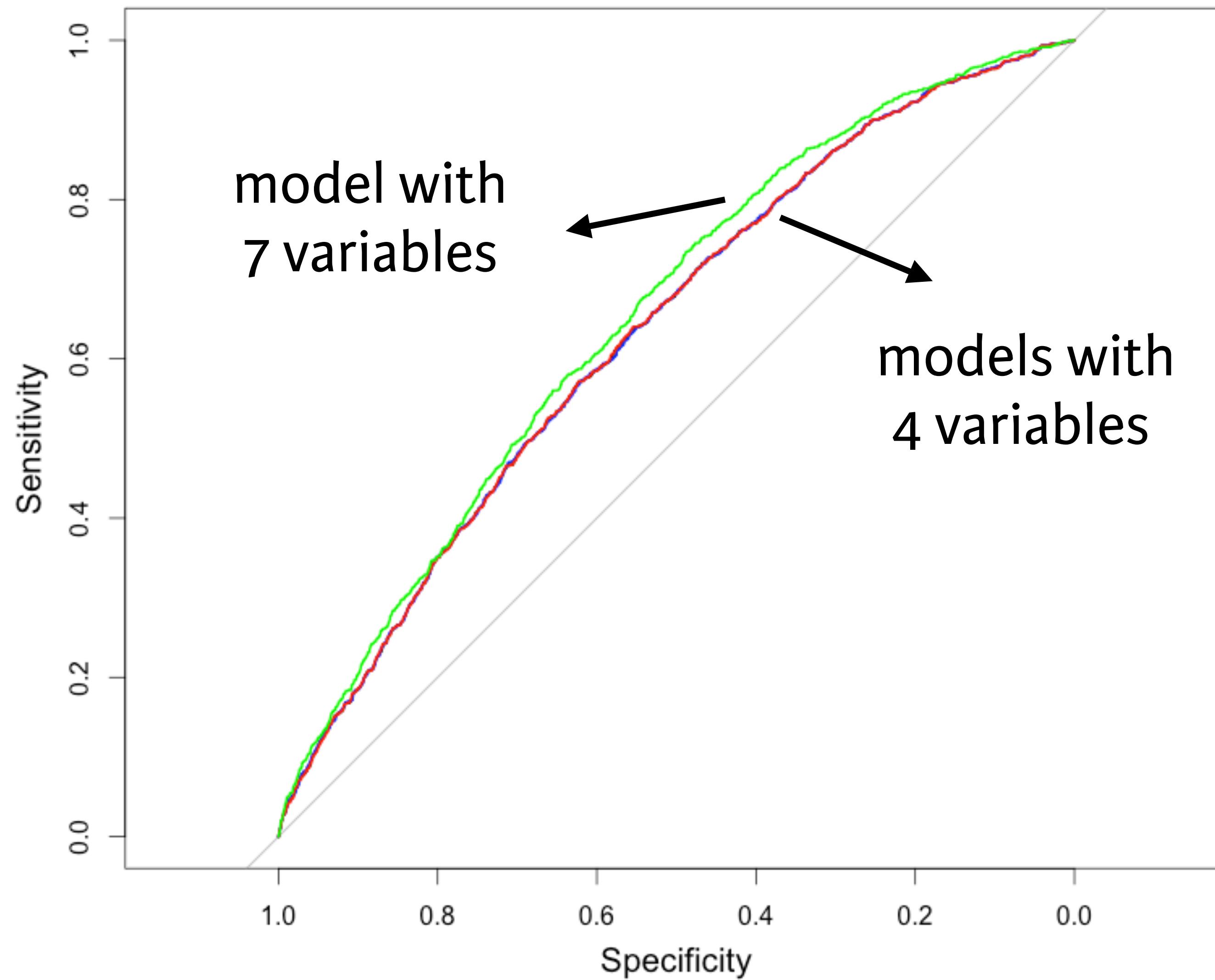
**Let's practice!**



CREDIT RISK MODELING IN R

# Input selection based on the AUC

# ROC curves for 4 logistic regression models



# AUC-based pruning

1) Start with a model including all variables (in our case, 7) and compute AUC

```
> log_model_full <- glm(loan_status ~ loan_amnt + grade + home_ownership +
  annual_inc + age + emp_cat + ir_cat, family = "binomial", data = training_set)

> predictions_model_full <- predict(log_model_full, newdata = test_set, type =
  "response")

> AUC_model_full <- auc(test_set$loan_status, predictions_model_full)
Area under the curve: 0.6512
```

# AUC-based pruning

2) Build 7 new models, where each time one of the variables is removed, and make PD-predictions using the test set

```
log_1_remove_amnt <- glm(loan_status ~ grade + home_ownership + annual_inc + age  
+ emp_cat + ir_cat, family = "binomial", data = training_set)  
  
log_1_remove_grade <- glm(loan_status ~ loan_amnt + home_ownership + annual_inc +  
age + emp_cat + ir_cat, family = "binomial", data = training_set)  
  
log_1_remove_home <- glm(loan_status ~ loan_amnt + grade + annual_inc + age +  
emp_cat + ir_cat, family = "binomial", data = training_set)  
  
...  
  
pred_1_remove_amnt <- predict(log_1_remove_amnt, newdata = test_set, type =  
"response")  
pred_1_remove_grade <- predict(log_1_remove_grade, newdata = test_set, type =  
"response")  
pred_1_remove_home <- predict(log_1_remove_home, newdata = test_set, type =  
"response")  
...
```

# AUC-based pruning

3) Keep the model that led to the best AUC (AUC full model: 0.6512)

```
> auc(test_set$loan_status, pred_1_remove_amnt)
Area under the curve: 0.6514

> auc(test_set$loan_status, pred_1_remove_grade)
Area under the curve: 0.6438

> auc(test_set$loan_status, pred_1_remove_home)
Area under the curve: 0.6537
...

```

Remove variable “home\_ownership”

4) Repeat until AUC decreases (significantly)



CREDIT RISK MODELING IN R

**Let's practice!**



CREDIT RISK MODELING IN R

# Course wrap-up

# Other methods

- Discriminant analysis
- Random forest
- Neural networks
- Support vector machines

# But... very classification-focused

- Timing aspect is neglected
- New popular method: survival analysis
  - PD's that change over time
  - time-varying covariates can be included



CREDIT RISK MODELING IN R

**The end!**