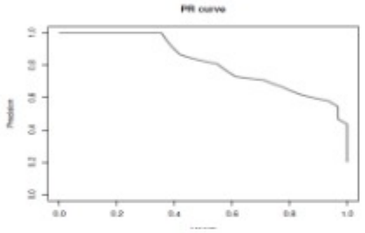
Precision-recall curves are often used for imbalanced applications

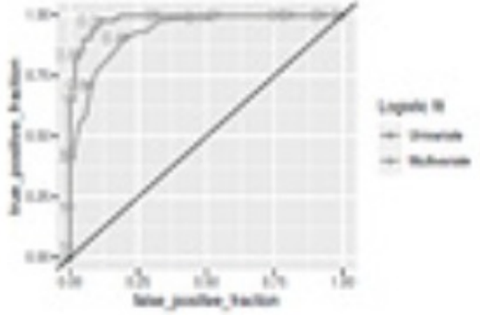
- Precision ($\frac{TP}{P}$) against the recall ($\frac{TP}{TP+FP}$).
- To plot a precision-recall curve we can use the package PRROC.

```
library(PRROC)

PRROC_obj <- pr.curve(scores.class0 = imbalanced_heights$mu_hat,
                      weights.class0=imbalanced_heights$y,
                      curve=TRUE)
plot(PRROC_obj, auc.main=FALSE , color=2)
```



Area under the ROC curve

- The area under the ROC curve (AUC) captures the overall performance.
- AUC = 0.5 : random
- AUC = 1 : perfect



H_0 rejected
$\Leftrightarrow$
p-value $\leq \alpha$
$\Leftrightarrow$
$P_{H_0}(H_0 \text{ rejected}) \leq \alpha$

a) $\mathbb{E}[\text{false positive}] = 5000 \cdot 0{,}05 = 250$ ✓

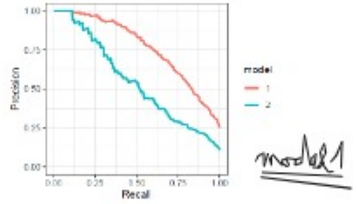
5000: number of cases where H_0 is true

0,05: probability that H_0 is rejected although H_0 true

b) $\mathbb{E}[\text{false positive}] = 250$ ✓

Problem 14 (1 credit)

Consider the precision-recall curves provided below, which compare the performance of two models.



model 1

Based on the provided curves, is it possible to determine which of the two models has superior performance?

height sizes

1,80 m

1,9 w

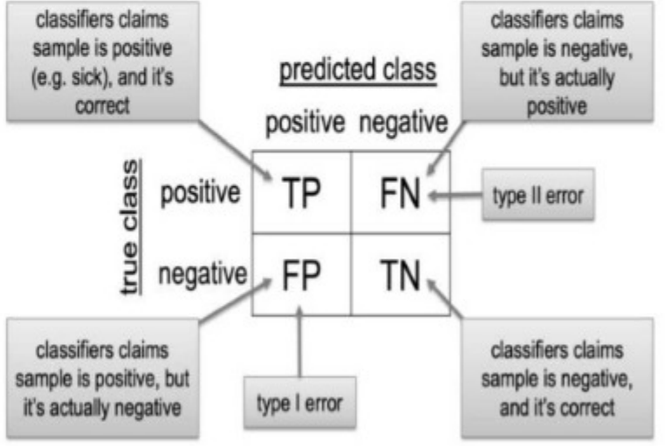
1,7 m

false positive

false negative

You say, person is male if height > 1,75.

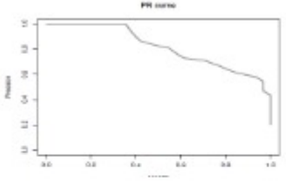
Confusion Matrix



Precision-recall curves are often used for imbalanced applications

- Precision ($\frac{TP}{P}$) against the recall ($\frac{TP}{TP+FP}$).
- To plot a precision-recall curve we can use the package PRROC.

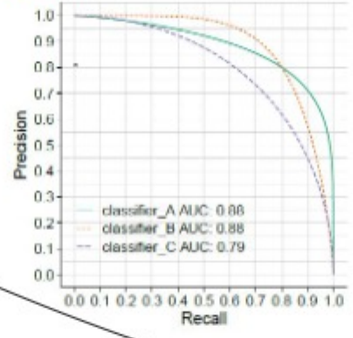
```
library(PRROC)

PRROC_obj <- pr.curve(scores.class0 = imbalanced_heights$mu_hat,
                      weights.class0=imbalanced_heights$y,
                      curve=TRUE)
plot(PRROC_obj, auc.main=FALSE , color=2)
```



positive = criminal

Problem 13 (1 credit)

A police department uses the help of a machine learning classifier to find criminals. They want to ensure that less than 10% of individuals classified as criminal are actually innocent. Which one of the three classifiers A, B or C shown in the Precision-Recall curves do you recommend to use? Justify your choice.



grüne Kurve = beste Methode

da die grüne Kurve ab Precision von 0.8 den höchsten Recall hat
Hoher Recall -> niederges FP -> wenige unschuldige werden als Kriminelle verurteilt

Classification performance metrics

- The *sensitivity* refers to the fraction of actual positives that is predicted to be positive:

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

The sensitivity is also referred to as "recall", "true positive rate", or "power".

- The *specificity* refers to the fraction of actual negatives that is predicted to be negative:

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

The specificity is also known as "true negative rate" or "sensitivity of the negative class"

- The *precision* refers to the fraction of predicted positives that are indeed positives:

Recall = Precision $= \frac{TP}{TP+FP}$ P

Problem 3 (5 points)

abhängig? yes!

Problem 7 (2 credits)

You have a dataset named **dt**, which includes information on cars from the **mtcars** dataset in R. The dataset contains information on miles per gallon (mpg) and transmission type (automatic or manual) of the cars.

mpg	transmission
21	Manual
21	Manual
23	Manual
21	Automatic
19	Automatic
18	Automatic
...	...

A colleague intends to investigate whether there is a significant difference in the median miles per gallon between cars with automatic and manual transmissions using a permutation test. The colleague visualizes a histogram of the test statistics across the permuted datasets alongside the value of the statistic T in the original unpermuted dataset (T reference). Examine the plot and the code used.

```r
# Permutation function
perm_test <- function(data, n_permutations = 1000) {
  # get reference statistics
  T_ref <- data[transmission == "Manual", median(mpg)] -
    data[transmission == "Automatic", median(mpg)]

  # get permutation statistics
  T_star <- sapply(1:n_permutations, function(x){
    # shuffle groups
    data[, transmission := sample(transmission), by='transmission']
    # compute statistics
    data[transmission == "Manual", median(mpg)] -
      data[transmission == "Automatic", median(mpg)] })

  g <- ggplot(data = data.table(T_star = T_star), aes(T_star)) +
    geom_histogram() +
    geom_vline(aes(xintercept=T_ref, color="T_ref")) + xlim(0, 10)
  print(g)
  return(list(T_ref=T_ref, T_star=T_star))
}
```

Problem 7 (2 credits)

You have a dataset named **dt**, which includes information on cars from the **mtcars** dataset in R. The dataset contains information on miles per gallon (mpg) and transmission type (automatic or manual) of the cars.

mpg	transmission
21	Manual
21	Manual
23	Manual
21	Automatic
19	Automatic
18	Automatic
...	...

Testgröße T_{obs} =
Durschnittsverbrauch der Gangschaltautos
- Durchschnittsverbrauch der Automatikschaltautos

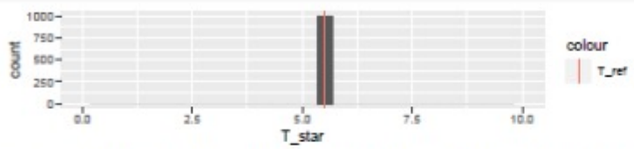
H_0 : Gangschaltautos verbrauchen weniger

Falls T_{obs} groß wird H_0 abgelehnt

Falls T_{obs} gegen 0 r klein!
$\Rightarrow \hat{p}$ klein
$\Rightarrow H_0$ wird abgelehnt

A colleague intends to investigate whether there is a significant difference in the median miles per gallon between cars with automatic and manual transmissions using a permutation test. The colleague visualizes a histogram of the test statistics across the permuted datasets alongside the value of the statistic T in the original unpermuted dataset (T reference). Examine the plot and the code used.

```r
# Permutation function
perm_test <- function(data, n_permutations = 1000) {
  # get reference statistics
  T_ref <- data[transmission == "Manual", median(mpg)] -
    data[transmission == "Automatic", median(mpg)]

  # get permutation statistics
  T_star <- sapply(1:n_permutations, function(x){
    # shuffle groups
    data[, transmission := sample(transmission), by='transmission']
    # compute statistics
    data[transmission == "Manual", median(mpg)] -
      data[transmission == "Automatic", median(mpg)] })

  g <- ggplot(data = data.table(T_star = T_star), aes(T_star)) +
    geom_histogram() +
    geom_vline(aes(xintercept=T_ref, color="T_ref")) + xlim(0, 10)
  print(g)
  return(list(T_ref=T_ref, T_star=T_star))
}
```



a) The code has a mistake. Clearly state the line of code that is wrong. Explain why it leads to the obtained histogram. Correct the line of code.
b) Assume the code is now correct. Give the formula to compute the P-value estimate using T_ref and T_star in this particular case. No R code is required.

b) $T_{ref} = T_{obs}$

$r = \#\{i \mid T\_star[i] \geq T_{obs}\}$

$$\hat{p} = \frac{r+1}{m+1}$$

length(T_star) // n_permutations

Permutation testing: P-value

permutation 1/22

- Let
  - m be the number of random (Monte Carlo) permutations
  - $r = \#\{T^* \geq T_{obs}\}$ be the number of these random permutations that produce a test statistic greater than or equal to that calculated for the actual data.
- The estimated one-sided P-value, $\hat{P}$ is:

$$\hat{P} = \frac{r+1}{m+1}$$

- Permutation P-values should never be zero.

Permutation testing: P-value

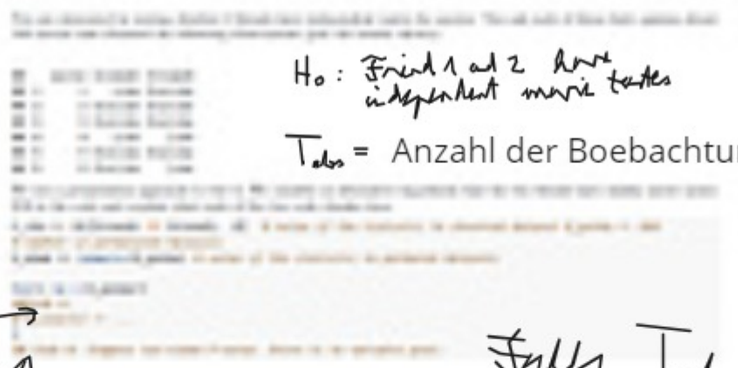
permutation 1/22

- Let
  - m be the number of random (Monte Carlo) permutations
  - $r = \#\{T^* \geq T_{obs}\}$ be the number of these random permutations that produce a test statistic greater than or equal to that calculated for the actual data.
- The estimated one-sided P-value, $\hat{P}$ is:

$$\hat{P} = \frac{r+1}{m+1}$$

- Permutation P-values should never be zero.

Problem 3



H_0: Friend 1 and 2 have independent movie tastes

T_{obs} = Anzahl der Boebachtungen, wo Freund 1 und 2 gleiche Tastes haben

$T_{obs} = 4$

Falls T_{obs} ganz, dann wird H_0 abgelehnt

$\Downarrow$

$r = \#\{T^* \geq T_{obs}\}$

$\Downarrow$

$\hat{p} = \frac{r+1}{m+1}$ ist klein

$\Downarrow$

H_0 wird abgelehnt

$T^*[i]$ = Wert der Teststatistik für die i-te Permutation

```
dt[ , friend2 := sample(friend2)]
t_star[i] = dt[friend == friend2, .N]
```

```
r = 0;
for i in 1:N_perm {
  if( T_star[i] >= T_ref ){
    r = r+1;
  }
}
p = (r+1) / (N_perm + 1)
```