Week 5: Functions

Quantitative Textanalyse mit R

Nils Reiter

May 22, 2020

R so far

- ► Vector, list, matrix, data.frame
 - ► Vectors are 1-based (i.e., start with 1)
- ► Types: numeric, character, logical
- Expressions: Assignments of values to variable names
- ► Selecting elements in containers: []

R so far

- ► Vector, list, matrix, data.frame
 - ▶ Vectors are 1-based (i.e., start with 1)
- ► Types: numeric, character, logical
- Expressions: Assignments of values to variable names
- Selecting elements in containers: []

```
(1:100)[c(TRUE,FALSE)] # every second element
m <- matrix(1:100, nrow=25)
m[4,5] # row 4 col 5
m[4,] # row 4
m[,5] # col 5</pre>
```

Grouped expressions: $expr_1, expr_2, ..., expr_n$ $expr_1, expr_2, ..., expr_n$

- Commands can be grouped with curly braces
- ▶ The value of the whole group is the value of the last expression within the group
- ▶ Grouped expressions do not form a scope on their own

Grouped expressions: $expr_1, expr_2, ..., expr_n$ $expr_1, expr_2, ..., expr_n$

- Commands can be grouped with curly braces
- ► The value of the whole group is the value of the last expression within the group
- Grouped expressions do not form a scope on their own

Conditional execution: if (expr_1) expr_2 else expr_3

- expr_2 only executed if expr_1 evaluates to a single logical TRUE value
- Implicit use of as.logical(expr_1)[1]
- Warning if length(expr_1)>1

```
Loops: for (name in expr) expr_2
```

- expr_2 repeatedly evaluated, while name is set to the elements of expr
- ▶ In R, no difference between 'foreach' and 'for'
- ▶ for i in 1:100 print(i) shows their equivalence

Loops: for (name in expr) expr_2

- expr_2 repeatedly evaluated, while name is set to the elements of expr
- ▶ In R, no difference between 'foreach' and 'for'
- ▶ for i in 1:100 print(i) shows their equivalence
- Loops are used much less in R compared to other languages
 - Most operations work in a vectorized fashion: We call a function that gets executed to every element of the vector
 - ▶ This vectorization is much faster, because it can be processed en bloc
 - Rule of thumbs: Avoid writing for-loops
 - There is even a vectorized version of if: ifelse(condition,a,b)

Functions

- ► Functions are mini-programs
- Named collections of commands
- ► Similar to other programming languages
- Syntax
 - ► Function definition: name <- function(arg_1, arg_2, ...) expression
 - ► Function call: name(arg_1, arg_2, ...)

Functions

- ► Functions are mini-programs
- Named collections of commands
- Similar to other programming languages
- Syntax
 - ► Function definition: name <- function(arg_1, arg_2, ...) expression
 - ► Function call: name(arg_1, arg_2, ...)
 - ▶ If the name starts and ends with %, and the function takes two arguments, it can be used as a binary operator

```
"%add%" <- function(x,y) {sum(x,y)} 5 %add% 7 # returns 12
```

Example

- ► Accuracy: Percentage of correctly classified instances
- ► Input: Two lists of labels
- Output: Number

```
accuracy <- function(gold, system) {
    # let's do that together
}</pre>
```

Example

- ► Accuracy: Percentage of correctly classified instances
- ► Input: Two lists of labels
- Output: Number

```
accuracy <- function(gold, system)
   sum(gold == system)/length(gold)</pre>
```

Any remaining issues? What could go wrong?

Example

Edge cases

Not the same length

We should verify that the vectors have the same length

Different types

- Types are coerced to character if necessary
 - Depending on use case: Maybe issue a warning

Different factors

▶ Not a problem, factor levels can be compared as character vectors

Named Arguments

- Function arguments can be named
- ▶ Named arguments can be given in any order
 - ▶ If both named and positional arguments are mixed, positional arguments go first
- Using named arguments increases readability of your code accuracy(system=listOfLabels, gold=realLabels)

Default Values

Specified in the function definition

```
accuracy <- function(gold, system, times=100)
   times * (sum(gold == system)/length)</pre>
```

Can be defined based on other arguments (!)

```
accuracy <- function(gold, system, limit=1:length(gold))
    sum(gold[limit] == system[limit])/length(limit)</pre>
```

Passing on arguments

- ➤ Sometimes we want to pass on arguments to other functions, in particular if we write 'wrapper functions' (i.e., functions that wrap another function)
- ▶ In these cases, we can use the ... argument (yes, it's three regular dots)
- ▶ All arguments not defined by our function are passed on to another function

```
partOfRandomVector <- function(n, x, ...) {
    v <- runif(n, ...)
    v[v<x]
}</pre>
```

We can now use all (named) arguments of runif as arguments for partOfRandomVector without further declaration

Anonymous functions (and apply)

- ► Similar to passing literal arguments to functions (as we have done before), we can pass anonymous functions
- ► They look the same, but are not assigned to a name: function(arg_1, arg_2) expression
- ► The function apply() expects a function as third argument
 - apply() applies this function to each row or column of a given matrix

Example

```
m <- matrix(1:100, ncol=5)
apply(m, 1, function(x) { sum(x**2) } )</pre>
```

Exercise 5

- ► At the usual place: https: //github.com/idh-cologne-sprachverarbeitung-mit-r/exercise-05
 - Includes these slides
- Less exercises, but a bit more complicated

ATM

- When we withdraw money from an ATM, it has to decide which bills it gives
- ► We will implement this function in R
- ▶ The function atm(v) takes a numeric value and returns a named list
 - ► The names are the bill types (i.e., 50)
 - ► The values are the number of bills to give us
- ► How to decide?
 - We want as few bills as possible (this is not what real ATMs to ...)

ATM

- When we withdraw money from an ATM, it has to decide which bills it gives
- We will implement this function in R
- ▶ The function atm(v) takes a numeric value and returns a named list
 - ► The names are the bill types (i.e., 50)
 - ► The values are the number of bills to give us
- ► How to decide?
 - We want as few bills as possible (this is not what real ATMs to ...)

Example

```
> unlist(atm(50))
500 200 100 50 20 10 5
0 0 0 1 0 0 0
> unlist(atm(195))
500 200 100 50 20 10 5
0 0 1 1 2 0 1
```

Fleiss' Kappa

▶ Fleiss' Kappa is a metric to calculate inter-annotator agreement

Word	Α	В	C	D
1	2	1		
2			1	2
3	1	1		1
4	3			

- Observed agreement: How many agreements have been achieved?
- Expected agreement: How many agreements can be expected if everyone guesses?

Observed Agreement

Normalized observed agreement for item i

Problem: k categories, n annotators, N items

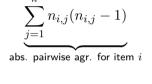
3	3	1 :	1	-	1
	N = 4	, k =	4, n	= 3	

Word

Observed Agreement

Normalized observed agreement for item \emph{i}

Problem: k categories, n annotators, N items



	VVord	А	В	C	D
	1	2	1		
	2			1	2
	3	1	1		1
	4	3			
N = 4, k = 4, n = 3					

Observed Agreement

Normalized observed agreement for item i

Problem: k categories, n annotators, N items

$$\underbrace{\frac{1}{n(n-1)}}_{\text{Scaling for annotators}} \times \underbrace{\sum_{j=1}^{j} n_{i,j}(n_{i,j}-1)}_{\text{abs. pairwise agr. for item } i}$$

Word	Α	В	С	D	
1	2	1			
2			1	2	
3	1	1		1	
4	3				
N = 4, k = 4, n = 3					

Observed Agreement

Normalized observed agreement for item \emph{i}

Problem: k categories, n annotators, N items

$$P_i = \underbrace{\frac{1}{n(n-1)}}_{ ext{Scaling for annotators}} imes \underbrace{\sum_{j=1}^{n} n_{i,j}(n_{i,j}-1)}_{ ext{abs. pairwise agr. for item }i$$

VVord	А	В	C	D	
1	2	1			
2			1	2	
3	1	1		1	
4	3				
N = 4, k = 4, n = 3					

Observed Agreement

Normalized observed agreement for item i

Problem: k categories, n annotators, N items

$$\hat{P}_i = \underbrace{\frac{1}{n(n-1)}}_{\text{Scaling for annotators}} \times \underbrace{\sum_{j=1}^{\kappa} n_{i,j}(n_{i,j}-1)}_{\text{abs. pairwise agr. for item } i}$$

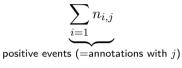
Normalized observed agreement for all items

$$P_o = \frac{1}{N} \sum_{i=1}^{N} \hat{P}_i$$

(this is just the arithmetic mean, a.k.a. average)

Expected Agreement

 $lackbox{ Probability that category } j$ gets selected (by one annotator)



Word	Α	В	С	D
1	2	1		
2 3	1	1	1	2 1
4	3			
N =	= 4, k	= 4,	n = 3	3

Expected Agreement

lacktriangleright Probability that category j gets selected (by one annotator)

VVord	А	В	C	D
1 2	2	1	1	2
3	1	1		1
4	3			
N	= 4, k	= 4,	n = 3	3

 $\underbrace{\frac{1}{nN}}_{\text{Possible events (all annotations)}} \times \underbrace{\sum_{i=1}^{N} n_{i,j}}_{\text{positive events (=annotations with } j)}$

Expected Agreement

lacktriangleright Probability that category j gets selected (by one annotator)

$p_j =$	\overline{nN} Possible events (all annotations)	$ imes \sum_{i=1}^{n_{i,j}} n_{i,j}$ positive events (=annotations with j)

Expected Agreement

▶ Probability that category *j* gets selected (by one annotator)

	rossible events (all afflotations)	positive events (=annotations with j)
▶ P	robability that two annotators select cat	egory j

Possible events (all annotations)

 $p_i \times p_i = p_i^2$

Expected Agreement

lacktriangle Probability that category j gets selected (by one annotator)

Possible events (all annotations)
$$\underbrace{\sum_{i=1}^{j-1}}_{\text{positive events (=annotations with } j)}$$

 $lackbox{ Probability that two annotators select category } j$

$$p_j \times p_j = p_j^2$$

Probability that two annotators are in agreement (over all categories):

$$P_e = \sum_{i=1}^k p_j^2$$

```
Probability for j
Observed agreement for i
```

```
p_j Probability for j
\hat{P}_i Observed agreement for i
P_o = \frac{1}{N} \sum_{i=1}^{N} \hat{P}_i
```

```
p_j Probability for j \hat{P}_i Observed agreement for i P_o = \frac{1}{N} \sum_{i=1}^N \hat{P}_i P_e = \sum_{j=1}^k p_j^2
```

$$\begin{array}{ll} p_j & \text{Probability for } j \\ \hat{P}_i & \text{Observed agreement for } i \\ P_o & = & \frac{1}{N} \sum_{i=1}^N \hat{P}_i \\ \\ P_e & = & \sum_{j=1}^k p_j^2 \\ \\ \kappa & = & \frac{P_o - P_e}{1 - P_e} \end{array}$$

$$p_j$$
 Probability for j
 \hat{P}_i Observed agreement for i
 $P_o = \frac{1}{N} \sum_{i=1}^N \hat{P}_i$
 $P_e = \sum_{j=1}^k p_j^2$
 $\kappa = \frac{P_o - P_e}{1 - P_e}$

- $ightharpoonup P_o P_e$: Tatsächlich erreichtes, nicht-zufälliges Agreement
- $ightharpoonup 1-P_e$: Maximal erreichbares, nicht-zufälliges Agreement

$$p_j$$
 Probability for j
 \hat{P}_i Observed agreement for i
 $P_o = \frac{1}{N} \sum_{i=1}^N \hat{P}_i$
 $P_e = \sum_{j=1}^k p_j^2$
 $\kappa = \frac{P_o - P_e}{1 - P_e}$

- $ightharpoonup P_o P_e$: Tatsächlich erreichtes, nicht-zufälliges Agreement
- $ightharpoonup 1 P_e$: Maximal erreichbares, nicht-zufälliges Agreement
- ▶ $-\infty < \kappa < 1$: Je höher desto besser
 - ► Extremfälle?

Section 3

Fleiss' Kappa

- ► Warum quantitativ messen?
 - Weil wir vergleichen wollen, über verschiedene Konfigurationen hinweg
 - ▶ (5 oder 6 Kategorien, 2 oder 3 AnnotatorInnen, 10 oder 20 Instanzen)

- ► Warum quantitativ messen?
 - Weil wir vergleichen wollen, über verschiedene Konfigurationen hinweg
 - ▶ (5 oder 6 Kategorien, 2 oder 3 AnnotatorInnen, 10 oder 20 Instanzen)
- ▶ Wie messen?
 - Problem: Wir wissen nicht was richtig ist
 - IAA ist ausschließlich eine Aussage über die Übereinstimmung, nicht über die Korrektheit

- Warum quantitativ messen?
 - Weil wir vergleichen wollen, über verschiedene Konfigurationen hinweg
 - ▶ (5 oder 6 Kategorien, 2 oder 3 AnnotatorInnen, 10 oder 20 Instanzen)
- ▶ Wie messen?
 - Problem: Wir wissen nicht was richtig ist
 - IAA ist ausschließlich eine Aussage über die Übereinstimmung, nicht über die Korrektheit
 - Verfahren das für beliebige Zahlen von Kategorien, AnnotatorInnen oder Instanzen funktioniert
 - Wie oft haben zwei AnnotatorInnen gleich gewählt? (paarweise Übereinstimmung)

Datenstrukturen

Word	A1	A2	А3
Zwei	ART	ART	CARD
Hunde	NN	NNS	NNS
bellen	VVFIN	VVINF	VVFINX
	\$.	\$.	\$.

Datenstrukturen

Word	A1	A2	А3
Zwei	ART	ART	CARD
Hunde	NN	NNS	NNS
bellen	VVFIN	VVINF	VVFINX
	\$.	\$.	\$.

 \Downarrow Konversion \Downarrow

Word	ART	CARD	NN	NNS	VVFIN	VVINF	VVFINX	\$.
Zwei Hunde bellen	2	1	1	2	1	1	1	3

Paarungen

Word	ART	CARD	NN	NNS	VVFIN	VVINF	VVFINX	\$.
Zwei	2	1						
Hunde			1	2				
bellen					1	1	1	2
•								3

▶ Wieviele paarweise Übereinstimmungen gibt es (absolut gemessen)?

Paarungen

Word	ART	CARD	NN	NNS	VVFIN	VVINF	VVFINX	\$.
Zwei	2	1						
Hunde			1	2				
bellen					1	1	1	
								3

▶ Wieviele paarweise Übereinstimmungen gibt es (absolut gemessen)?

$$1 + 1 + 0 + 3 = 5$$

Paarungen

Word	ART	CARD	NN	NNS	VVFIN	VVINF	VVFINX	\$.
Zwei Hunde bellen	2	1	1	2	1	1	1	2
								3

- ▶ Wieviele paarweise Übereinstimmungen gibt es (absolut gemessen)?
 - 1 + 1 + 0 + 3 = 5
- Warum 3? Binomialkoeffizient!

Anzahl möglicher k-elementiger Teilmengen aus n Dingen

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Anzahl möglicher k-elementiger Teilmengen aus n Dingen

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Fakultät:

$$n! = n(n-1)(n-2)(n-3)\cdots 1$$

Anzahl möglicher k-elementiger Teilmengen aus n Dingen

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Fakultät:

$$n! = n(n-1)(n-2)(n-3)\cdots 1$$

$$\binom{n}{2} = \frac{n!}{(n-2)!2!}$$

Anzahl möglicher k-elementiger Teilmengen aus n Dingen

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Fakultät:

$$n! = n(n-1)(n-2)(n-3)\cdots 1$$

$$\binom{n}{2} = \frac{n!}{(n-2)!2!} = \frac{1}{2} \frac{n!}{(n-2)!}$$

Anzahl möglicher k-elementiger Teilmengen aus n Dingen

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Fakultät:

$$n! = n(n-1)(n-2)(n-3)\cdots 1$$

$$\binom{n}{2} = \frac{n!}{(n-2)!2!} = \frac{1}{2} \frac{n!}{(n-2)!}$$
$$= \frac{1}{2} \frac{n(n-1)(n-2)(n-3)\cdots 1}{(n-2)(n-3)\cdots 1}$$

Anzahl möglicher k-elementiger Teilmengen aus n Dingen

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Fakultät:

$$n! = n(n-1)(n-2)(n-3)\cdots 1$$

$$\binom{n}{2} = \frac{n!}{(n-2)!2!} = \frac{1}{2} \frac{n!}{(n-2)!}$$
$$= \frac{1}{2} \frac{n(n-1)(n-2)(n-3)\cdots 1}{(n-2)(n-3)\cdots 1}$$
$$= \frac{1}{2} n(n-1)$$

Anzahl möglicher k-elementiger Teilmengen aus n Dingen

Reihenfolge: Sind (a,b) und (b,a) das gleiche Paar?

- ▶ Ja: $\frac{1}{2}n(n-1)$
- ▶ Nein: n(n-1)

Paarungen

$\binom{n}{2}$
0
1
3
6
10
45
4.950

Table: $\binom{n}{2}$ Werte für steigende n

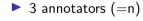
Problems

Word	Α	В	С	D
1	2	1		
2			1	2
3	1	1		1
4	3			

- ▶ 3 annotators (=n)
- ▶ 5 pairwise agreements

Problems

<u>Situatio</u>	<u>n 1:</u>				
Word	Α	В	C	D	
1	2	1			
2			1	2	
3	1	1		1	
4	3				



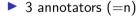
5 pairwise agreements

Situation 2:								
Word	Α	В	C	D				
1	2	1	1					
2		1	1	2				
3	1	1	1	1				
4	3	1						

4 annotators (=n)5 pairwise agreements

Problems

<u>Situatio</u>	<u>n 1:</u>				
Word	Α	В	C	D	
1	2	1			
2			1	2	
3	1	1		1	
4	3				



▶ 5 pairwise agreements

Situation 2:								
Word	Α	В	C	D				
1	2	1	1					
2		1	1	2				
3	1	1	1	1				
4	3	1						

▶ 4 annotators (=n)

▶ 5 pairwise agreements

How much worse is Situation 2 compared to 1?

 $\to \mathsf{Scaling!}$

Inter-Annotator Agreement Scaling

- Sometimes, values have different scales
 - i.e., different min and max values
- Scaling: Apply a function to values so that they are comparable
 - ► Simplest way: Divide by the (theoretical) maximum

Inter-Annotator Agreement Scaling

- Sometimes, values have different scales
 - i.e., different min and max values
- Scaling: Apply a function to values so that they are comparable
 - Simplest way: Divide by the (theoretical) maximum
- ▶ What's the theoretical maximum here?
 - ▶ If all annotators agree: $\binom{n}{2} = \frac{1}{2}n(n-1)$

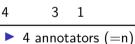
Inter-Annotator Agreement Scaling

- Sometimes, values have different scales
 - i.e., different min and max values
- Scaling: Apply a function to values so that they are comparable
 - Simplest way: Divide by the (theoretical) maximum
- ▶ What's the theoretical maximum here?
 - ▶ If all annotators agree: $\binom{n}{2} = \frac{1}{2}n(n-1)$
- ightharpoonup Scaling \simeq Normalization

<u>ituatio</u>		_		_
Word	Α	В	C	D
1	2	1		
2			1	2
3	1	1		1
4	3			

$$=\frac{5}{100}=0.416$$

Scaled:
$$\frac{5}{4\binom{3}{2}} = \frac{5}{4 \times 3} = \frac{5}{12} = 0.416$$



Situation 2:

Word A B C

5 pairwise agreements

► Scaled: $\frac{5}{4\binom{4}{2}} = \frac{5}{4\times 6} = \frac{5}{24} = 0.208$

Observed Agreement

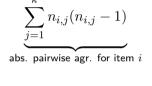
Normalized observed agreement for item iProblem: k categories, n annotators, N items 2 1

Word

Observed Agreement

Normalized observed agreement for item \emph{i}

Problem: k categories, n annotators, N items



Word	А	В	C	D
1 2	2	1	1	2
3	1	1	-	1
4	3			
N:	=4, k	= 4,	n = 3	3

Observed Agreement

Normalized observed agreement for item i

Problem: k categories, n annotators, N items

$$\underbrace{\frac{1}{n(n-1)}}_{\text{Scaling for annotators}} \times \underbrace{\sum_{j=1} n_{i,j}(n_{i,j}-1)}_{\text{abs. pairwise agr. for item } i}$$

Word	Α	В	C	D
1	2	1	1	_
2 3	1	1	1	2 1
4	3			
N =	= 4, k	= 4,	n = 3	3

Observed Agreement

Normalized observed agreement for item \emph{i}

Problem: k categories, n annotators, N items

$$P_i = \underbrace{\frac{1}{n(n-1)}}_{\text{Scaling for annotators}} \times \underbrace{\sum_{j=1}^{n} n_{i,j}(n_{i,j}-1)}_{\text{abs. pairwise agr. for item } i}$$

Word	Α	В	C	D
1	2	1		
2			1	2
3	1	1		1
4	3			
N =	= 4, k	= 4,	n = 3	3

Observed Agreement

Normalized observed agreement for item i

Problem: k categories, n annotators, N items

$$\hat{P}_i = \underbrace{\frac{1}{n(n-1)}}_{\text{Scaling for annotators}} \times \underbrace{\sum_{j=1}^{\kappa} n_{i,j}(n_{i,j}-1)}_{\text{abs. pairwise agr. for item } i}$$

Normalized observed agreement for all items

$$P_o = \frac{1}{N} \sum_{i=1}^{N} \hat{P}_i$$

(this is just the arithmetic mean, a.k.a. average)

Word	Α	В	С	D
1 2 3	2	1	1	2
4	3			
N =	= 4, k	= 4,	n = 3	3

Large Operators

$$\sum_{i=0}^{n} f(i) = f(0) + f(1) + f(2) + \dots + f(n)$$

Large Operators

$$\sum_{i=0}^{n} f(i) = f(0) + f(1) + f(2) + \dots + f(n)$$

Example

$$\sum_{i=0}^{4} i^5 = 0^5 + 1^5 + 2^5 + 3^5 + 4^5 = 1300$$

Large Operators

$$\sum_{i=0}^{n} f(i) = f(0) + f(1) + f(2) + \dots + f(n)$$

Example

$$\sum_{i=0}^{4} i^5 = 0^5 + 1^5 + 2^5 + 3^5 + 4^5 = 1300$$

$$\prod_{i=1}^{3} (5i+1) = (5(1)+1) \times (5(2)+1) \times (5(3)+1)$$

$$= 6+11+16=33$$

Expected Agreement

\sim						- 1
_	+ 1	ın	+ 1	\sim	n	1
Si	LL	ıa	LI	u		1

Word	Α	В	C	D
1	2	1		
2			1	2
3	1	1		1
4	3			

- ightharpoonup n=3 annotators
- ightharpoonup k = 4 categories
- ▶ 5 pairwise agreements

0	٠.					\sim
\rightarrow	ıtı	ıa	Ť١	\sim	n	2:

Word	A	В	С
1	2	1	
2	2		1
3	1	1	1
4	3		

- ightharpoonup n=3 annotators
- ightharpoonup k = 3 categories
- ► 5 pairwise agreements

What situation had the better agreement? How much better?

Expected Agreement

ightharpoonup How likely is it (in general) that category A is selected?

Expected Agreement

▶ How likely is it (in general) that category *A* is selected?

Excurs: Relative Frequencies

► How often do you win a game?

Expected Agreement

▶ How likely is it (in general) that category *A* is selected?

Excurs: Relative Frequencies

- How often do you win a game?
- Number of positive events (= wins)

 Number of total events(= games played)

Expected Agreement

▶ How likely is it (in general) that category *A* is selected?

Excurs: Relative Frequencies

- How often do you win a game?
- Number of positive events (= wins)

 Number of total events(= games played)
- ightharpoonup p("selecting category A") = ?

Expected Agreement

▶ How likely is it (in general) that category *A* is selected?

Excurs: Relative Frequencies

- How often do you win a game?
- Number of positive events (= wins)

 Number of total events(= games played)
- ightharpoonup p("selecting category A") = ?
- Relative frequency is an estimate of the probability
 - Probability: Theoretical concept
 - ▶ Relative frequency: Result of actual experiments
 - Assumption: The more experiments I do, the more similar is the relative frequency to the probability

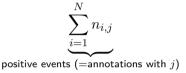
Expected Agreement

Situation 1:						<u>Situat</u>	Situation 2:	
	Word	Α	В	C	D	Wor	d A	В
	1	2	1			1	2	1
	2			1	2	2	2	
	3	1	1		1	3	1	1
	4	3				4	3	

- $\qquad \qquad p(\text{"selecting category A"}) = \frac{\text{Number of positive events}}{\text{Number of possible events}}$
 - ▶ Positive events: How often category A has been selected
 - Possible events: How many decisions have been made
- ightharpoonup p("selecting category A") =
 - Situation 1: $\frac{6}{}$
 - Situation 1: ⁶/₁₂
 Situation 2: ⁸/₁₂

Expected Agreement

▶ Probability that category *j* gets selected (by one annotator)



Expected Agreement

▶ Probability that category *j* gets selected (by one annotator)



Expected Agreement

▶ Probability that category *j* gets selected (by one annotator)

$$p_j = \underbrace{\frac{1}{nN}}_{\text{Possible events (all annotations)}} \times \underbrace{\sum_{i=1}^{N} n_{i,j}}_{\text{positive events (=annotations with } j)}$$

Expected Agreement

▶ Probability that category *j* gets selected (by one annotator)

$$p_j = \underbrace{\frac{1}{nN}}_{\text{Possible events (all annotations)}} \times \underbrace{\sum_{i=1}^{N} n_{i,j}}_{\text{positive events (=annotations with } j)}$$

lacktriangle Probability that two annotators select category j

$$p_j \times p_j = p_j^2$$

Expected Agreement

▶ Probability that category *j* gets selected (by one annotator)

$$p_j = \underbrace{\frac{1}{nN}}_{ ext{Possible events (all annotations)}} imes \underbrace{\sum_{i=1}^N n_{i,j}}_{ ext{positive events (=annotations with } j)}$$

ightharpoonup Probability that two annotators select category j

$$p_j \times p_j = p_j^2$$

▶ Probability that two annotators are in agreement (over all categories):

$$P_e = \sum_{i=1}^k p_j^2$$

```
Probability for j
Observed agreement for i
```

```
p_j Probability for j
\hat{P}_i Observed agreement for i
P_o = \frac{1}{N} \sum_{i=1}^{N} \hat{P}_i
```

```
p_j Probability for j \hat{P}_i Observed agreement for i P_o = \frac{1}{N} \sum_{i=1}^N \hat{P}_i P_e = \sum_{j=1}^k p_j^2
```

$$\begin{array}{ll} p_j & \text{Probability for } j \\ \hat{P}_i & \text{Observed agreement for } i \\ P_o & = & \frac{1}{N} \sum_{i=1}^N \hat{P}_i \\ \\ P_e & = & \sum_{j=1}^k p_j^2 \\ \\ \kappa & = & \frac{P_o - P_e}{1 - P_e} \end{array}$$

$$p_j$$
 Probability for j
 \hat{P}_i Observed agreement for i
 $P_o = \frac{1}{N} \sum_{i=1}^N \hat{P}_i$
 $P_e = \sum_{j=1}^k p_j^2$
 $\kappa = \frac{P_o - P_e}{1 - P_e}$

- $ightharpoonup P_o P_e$: Tatsächlich erreichtes, nicht-zufälliges Agreement
- $lacktriangleq 1-P_e$: Maximal erreichbares, nicht-zufälliges Agreement

$$p_j$$
 Probability for j
 \hat{P}_i Observed agreement for i
 $P_o = \frac{1}{N} \sum_{i=1}^N \hat{P}_i$
 $P_e = \sum_{j=1}^k p_j^2$
 $\kappa = \frac{P_o - P_e}{1 - P_c}$

- $ightharpoonup P_o P_e$: Tatsächlich erreichtes, nicht-zufälliges Agreement
- $ightharpoonup 1 P_e$: Maximal erreichbares, nicht-zufälliges Agreement
- ▶ $-\infty < \kappa < 1$: Je höher desto besser
 - ► Extremfälle?