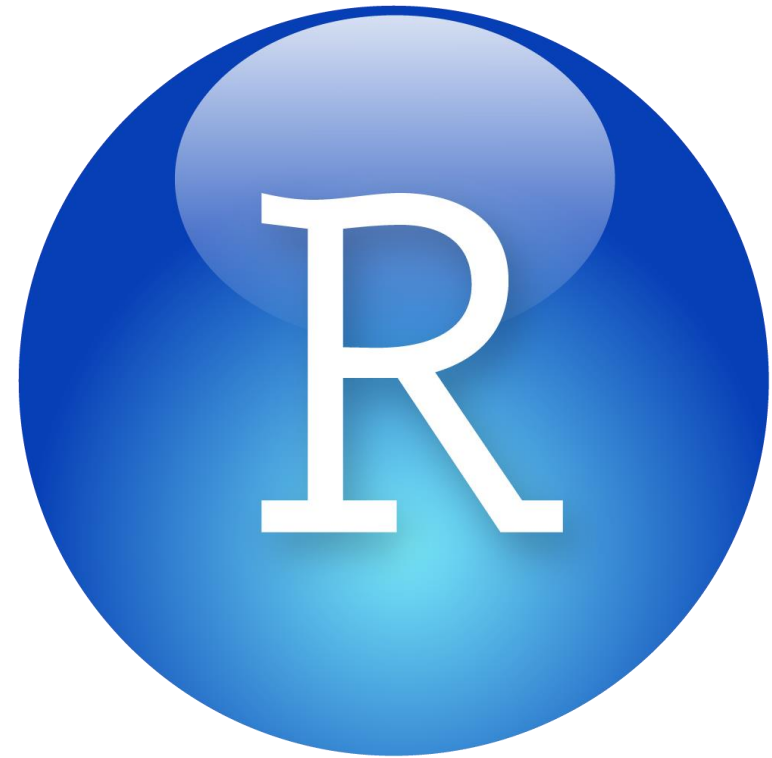


Information Retrieval – begleitendes Tutorium VIII

Thomas Schmidt

thomas.schmidt@stud.uni-regensburg.de

Heute



Allgemeines – R und R-Studio

- Freie Programmiersprache für statistisches Rechnen und Grafiken
- R-Studio liefert ein User-Interface zur vereinfachten Nutzung von R (auch frei verfügbar)
- Zahlreiche Pakete für verschiedene Anwendungsszenarien
- Konkurrenzprodukt SPSS
- → für euch: Nutzung in der Evaluationsphase des Projekts bei der systemzentrierten Evaluation

R und R-Studio

Um R-Studio nutzen zu können benötigt man erst R.

Download von R:

<http://www.r-project.org/>

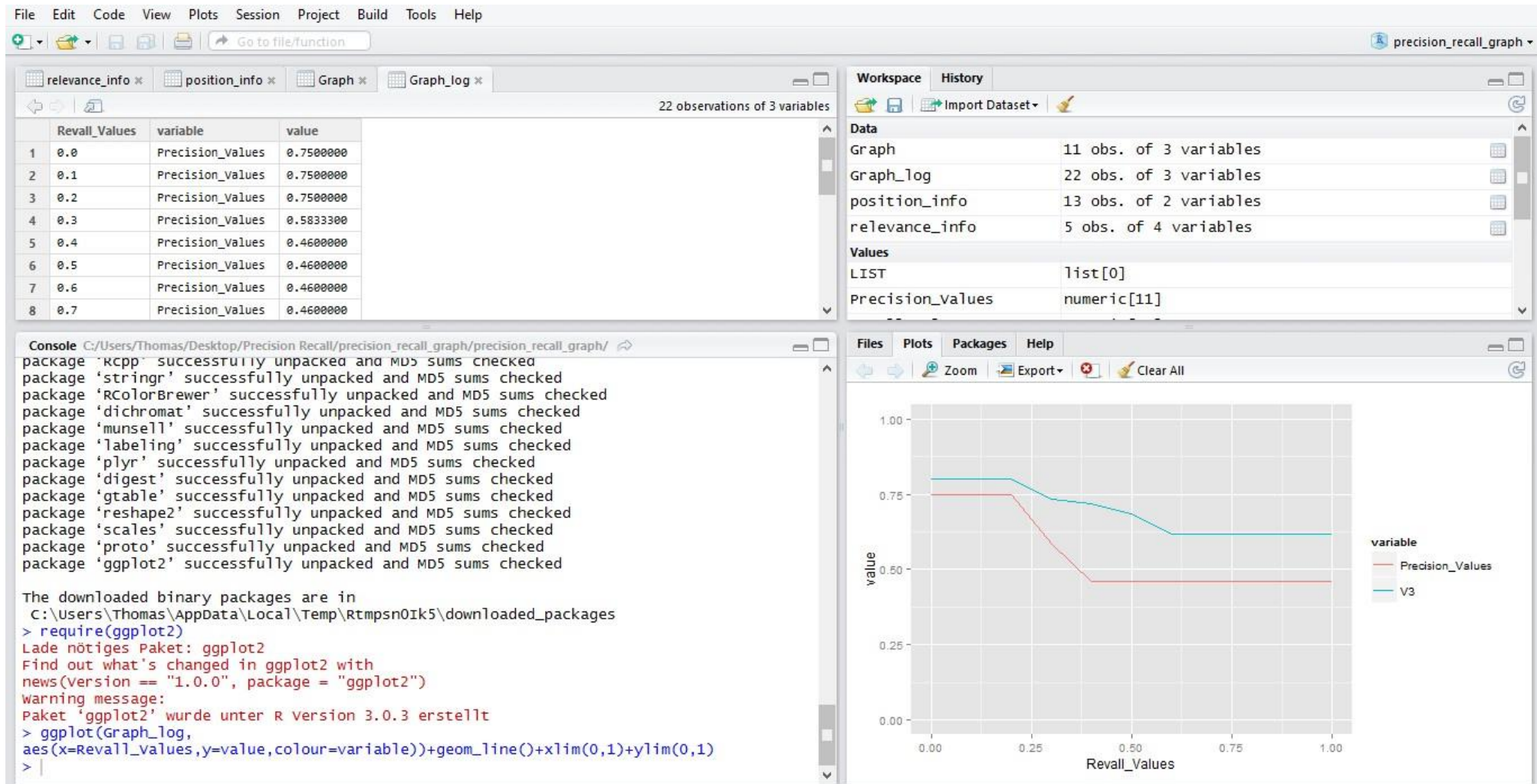
Download von R-Studio:

<http://www.rstudio.com/products/RStudio/>

Ablauf

1. „R-Studio in a Nutshell“ – Live-Beispiel
2. Anwendungsbeispiel: Recall-Precision-Graph
3. Anwendungsbeispiel: T-Test
4. Überblick über weitere mögliche Auswertungen
5. Letzte Tipps für das Projekt

R-Studio



Recall-Precision-Graph

- notwendig für Gruppe Web, Twitter und News
- Funktion **precision_recall_graph(relevance_info, position_info)** in Grips oder bestehende Beispielprojekte verwenden
- Gewisse Vorarbeiten müssen geleistet werden!

CSV-Dateien: position_info und relevance_info

- Müssen „händisch“ erstellt werden z.B. in Excel
- Export als CSV
- Format und Benennung muss eingehalten werden
- Ergebnisse basieren auf systemzentrierter Evaluation (Relevance Assessment, Auswertung usw.)
- Da Vergleich verschiedener Systemzustände/Suchmaschinen je ein paar für jeden Zustand: Also 2 mal position_info (A und B) und 2 mal relevance_info (A und B) für den jeweiligen Zustand A und B

relevance-info

- Query (hier über eine id)
- Anzahl zurückgelieferter Dokumente (N_back)
- Anzahl zurückgelieferter relevanter Dokumente (N_rel)
- Anzahl aller relevanten Dokumente insgesamt (REL_total)

	A	B	C	D
1	Query	N_back	N_rel	REL_total
2	1	10	5	5
3	2	10	3	3
4	3	12	2	10
5	4	30	2	10
6	5	23	1	10
7				

position-info

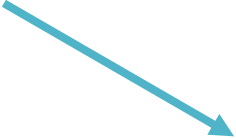
- Query (hier über eine id, Query_pos)
- Position des relevanten Dokuments im Ranking (Pos_pos)

	A	B
1	Query_pos	Pos_pos
2	1	1
3	1	3
4	1	6
5	1	9
6	1	10
7	2	2
8	2	5
9	2	7
10	3	2
11	3	4
12	4	1
13	4	3
14	5	1
15		

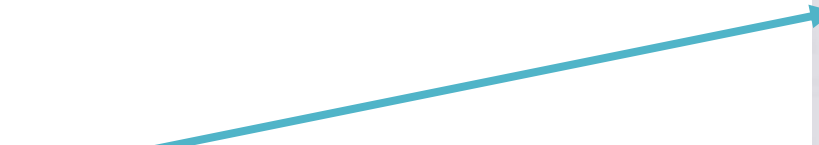
Import in R-Studio

- Tools → Import Dataset → From Text File → Datei auswählen (z.B. position_info_A) → Import

Data	
position_info_A	13 obs. of 2 variables
relevance_info_A	5 obs. of 4 variables



	Query	N_back	N_rel	REL_total
1	1	10	5	5
2	2	10	3	3
3	3	12	2	10
4	4	30	2	10
5	5	23	1	10



	Query_pos	Pos_pos
1	1	1
2	1	3
3	1	6
4	1	9
5	1	10
6	2	2
7	2	5
8	2	7
9	3	2
10	3	4
11	4	1
12	4	3
13	5	1

2 verschiedene Zustände

Data	
position_info_A	13 obs. of 2 variables
position_info_B	17 obs. of 2 variables
relevance_info_A	5 obs. of 4 variables
relevance_info_B	5 obs. of 4 variables

Recall_Values und Precision_Values erstellen

```
> Recall_values <- c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1)
```

→ 11-point-Recall-Precision-Graph

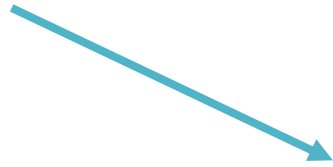
```
> Precision_values_A <- precision_recall_graph(relevance_info_A, position_info_A)
```



```
structure(c(0.8, 0.8, 0.8, 0.7333333333333333, 0.719047619047619,  
0.685714285714286, 0.619047619047619, 0.619047619047619, 0.619047619047619,  
0.619047619047619, 0.619047619047619), .Names = c("V1", "V2",  
"V3", "V4", "V5", "V6", "V7", "V8", "V9", "V10", "V11"))|
```

Data-Frame aus Recall_Values und den beiden Precision_Values erstellen

```
> Graph <- data.frame(Recall_Values, Precision_Values_A, Precision_Values_B)
```



	row.names	Recall_Values	Precision_Values_A	Precision_Values_B
1	V1	0.0	0.8000000	0.5800000
2	V2	0.1	0.8000000	0.5800000
3	V3	0.2	0.8000000	0.5800000
4	V4	0.3	0.7333333	0.5800000
5	V5	0.4	0.7190476	0.5800000
6	V6	0.5	0.6857143	0.5800000
7	V7	0.6	0.6190476	0.4600000
8	V8	0.7	0.6190476	0.4400000
9	V9	0.8	0.6190476	0.4400000
10	V10	0.9	0.6190476	0.4314286
11	V11	1.0	0.6190476	0.4314286

Vom Wide-Format zum Long-Format

reshape-Package importieren (install.packages(„reshape“))

```
> require(reshape)
> Graph_log <- melt(Graph, id="Recall_values")
```


	row.names	Recall_Values	Precision_Values_A	Precision_Values_B
1	V1	0.0	0.8000000	0.5800000
2	V2	0.1	0.8000000	0.5800000
3	V3	0.2	0.8000000	0.5800000
4	V4	0.3	0.7333333	0.5800000
5	V5	0.4	0.7190476	0.5800000
6	V6	0.5	0.6857143	0.5800000
7	V7	0.6	0.6190476	0.4600000
8	V8	0.7	0.6190476	0.4400000
9	V9	0.8	0.6190476	0.4400000
10	V10	0.9	0.6190476	0.4314286
11	V11	1.0	0.6190476	0.4314286



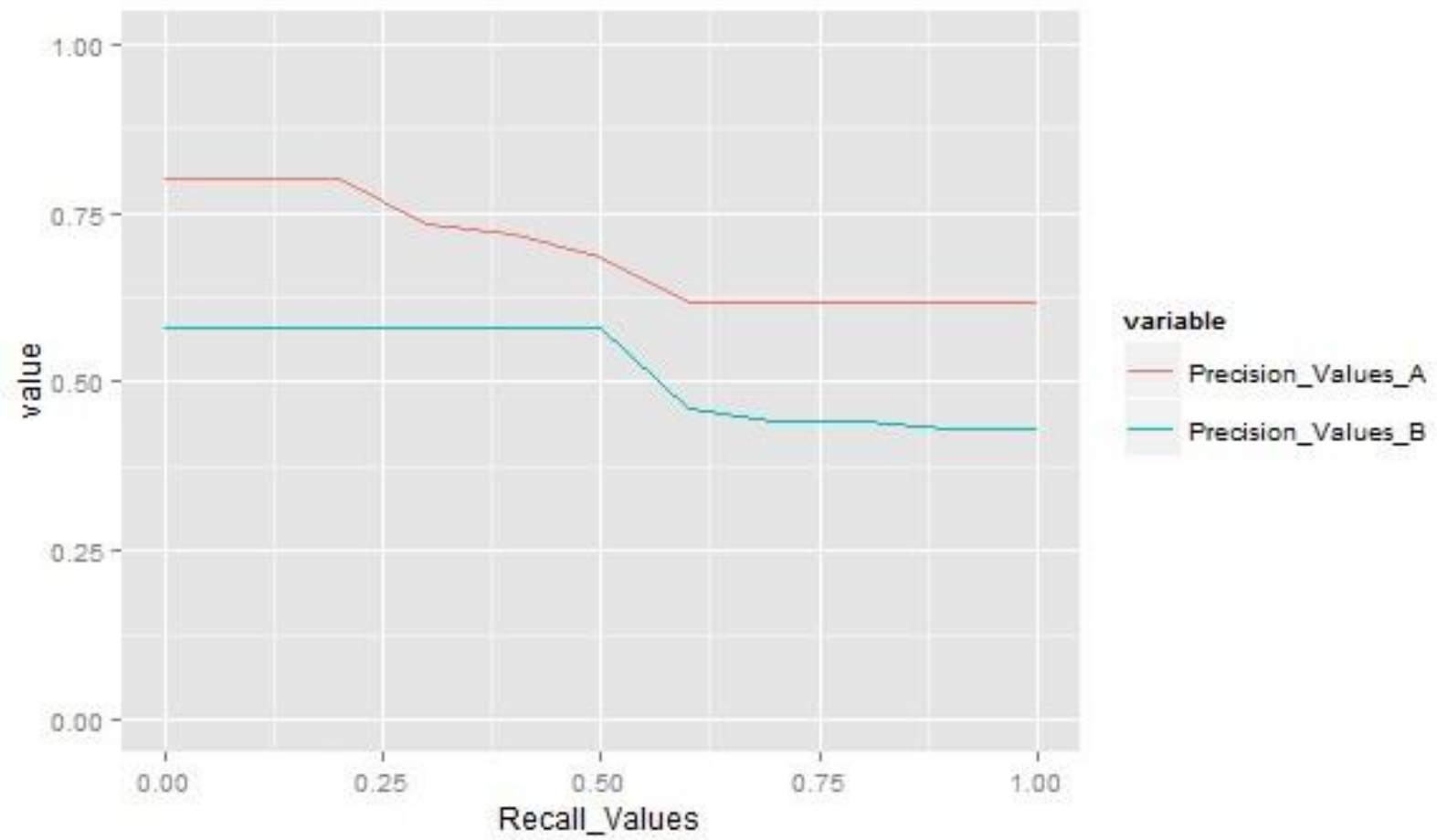
	Recall_Values	variable	value
1	0.0	Precision_Values_A	0.8000000
2	0.1	Precision_Values_A	0.8000000
3	0.2	Precision_Values_A	0.8000000
4	0.3	Precision_Values_A	0.7333333
5	0.4	Precision_Values_A	0.7190476
6	0.5	Precision_Values_A	0.6857143
7	0.6	Precision_Values_A	0.6190476
8	0.7	Precision_Values_A	0.6190476
9	0.8	Precision_Values_A	0.6190476
10	0.9	Precision_Values_A	0.6190476
11	1.0	Precision_Values_A	0.6190476
12	0.0	Precision_Values_B	0.5800000
13	0.1	Precision_Values_B	0.5800000
14	0.2	Precision_Values_B	0.5800000
15	0.3	Precision_Values_B	0.5800000
16	0.4	Precision_Values_B	0.5800000
17	0.5	Precision_Values_B	0.5800000
18	0.6	Precision_Values_B	0.4600000
19	0.7	Precision_Values_B	0.4400000
20	0.8	Precision_Values_B	0.4400000
21	0.9	Precision_Values_B	0.4314286
22	1.0	Precision_Values_B	0.4314286

Plotten

mit ggplot2-Paket (install.packages("ggplot2"))

```
> require(ggplot2)
Lade nötiges Paket: ggplot2
Find out what's changed in ggplot2 with
news(version == "1.0.0", package = "ggplot2")
warning message:
Paket 'ggplot2' wurde unter R version 3.0.3 erstellt
> ggplot(Graph_log, aes(x=Recall_values, y=value, colour=variable))+geom_line()+xlim(0,1)
  )+ylim(0,1)
```

Beispiel



T-Test

- Zum Vergleich von 2 verschiedenen Systemen (oder Systemzuständen)
- Verschiedene Parameter zum Vergleich denkbar (P@10, P@1, MRR, Rang des ersten relevanten Doks. usw.)
- Für jede Anfrage wird der Parameter bei den einzelnen Systemen erfasst (je mehr Anfragen verglichen werden desto valider)
- T-Test → Mittelwertvergleich der beiden Systeme mit Signifikanzniveau
- Signifikant bei p-Wert ≤ 0.05 , marginal signifikant bei p-Wert ≤ 0.1
- Voraussetzung: Normalverteilung, Varianzhomogenität (nicht unbedingt) → t-Test (sehr robust)
- Vorher unbedingt Hypothesen aufstellen

Ausgangsdaten

Hier: fiktive Precision-Werte

CSV-Datei

Einlesen in R-Studio wie vorher

	Query	System_A	System_B
1	1	0.8	1.00
2	2	0.2	0.50
3	3	1.0	0.30
4	4	1.0	0.25
5	5	0.3	0.10
6	6	0.5	0.80
7	7	0.9	1.00
8	8	0.1	0.10
9	9	0.0	0.00
10	10	0.2	0.70

Numerische Vektoren erstellen

```
> system_A <- t_test_Testdaten[["system_A"]]  
> system_B <- t_test_Testdaten[["system_B"]]
```


Data	
t_test_Testdaten	10 obs. of 3 variables
Values	
system_A	numeric[10]
system_B	numeric[10]

T-Test durchführen

```
> t.test(system_A, system_B, paired=TRUE, alternative="greater")
```

Paired t-test

```
data: system_A and system_B  
t = 0.1895, df = 9, p-value = 0.4269  
alternative hypothesis: true difference in means is greater than 0  
95 percent confidence interval:  
 -0.2168235      Inf  
sample estimates:  
mean of the differences  
              0.025
```




Hier: nicht signifikant
Hypothese muss verworfen
werden

```
> t.test(system_A, system_B, paired=TRUE, alternative="less")
```

Paired t-test

```
data: system_A and system_B  
t = 0.1895, df = 9, p-value = 0.5731  
alternative hypothesis: true difference in means is less than 0  
95 percent confidence interval:  
 -Inf 0.2668235  
sample estimates:  
mean of the differences  
              0.025
```

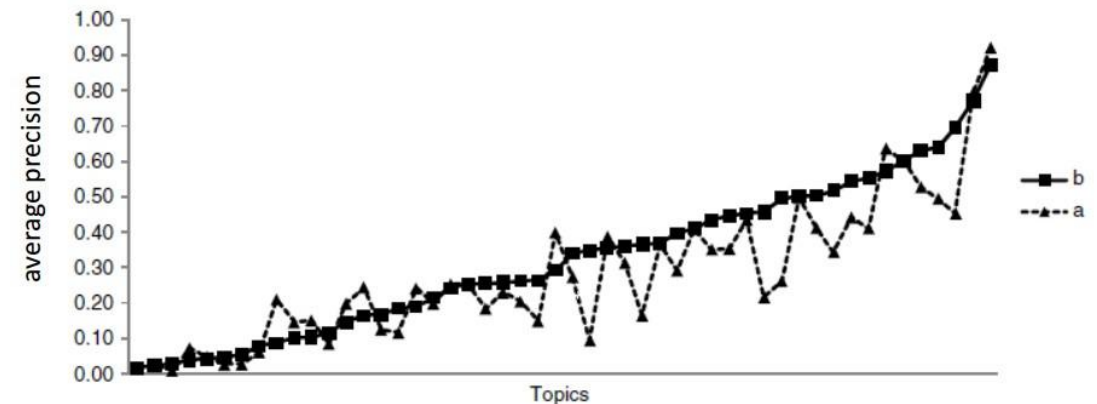


Sonstiges

- Tabellen und Visualisierungen von Parametern
- Topic-by-Topic-Vergleich
- Visualisierung vom t-Test z.B. Balkendiagramm der Mittelwerte mit Standardfehler
- Siehe auch Vorlesung zu weiteren Auswertungsmöglichkeiten

	WEB	TWITTER	NEWS	RECOMMENDER	PIM
P@10	X	X			
P@1	X				X
MRR					X
F-Maß			X	X	
R-Precision			X		

- Topic-by-Topic Vergleich



Viel Erfolg beim Projekt und eine schöne
vorlesungsfreie Zeit!

