

# Leseanleitung für QIDB-Rechenregeln

**IQTIG-spezifische Variablen, Operatoren und Funktionen in R**

# Inhaltsverzeichnis

1	Einleitung .....	4
1.1	Grundlegende Konventionen .....	4
1.1.1	Datengrundlage .....	4
1.1.2	Mehrfachfelder .....	4
1.1.3	Operatorpräzedenz .....	5
1.1.4	Vorfilter .....	5
2	Variablen .....	6
2.1	VB.....	6
2.2	LST .....	6
3	Operatoren .....	7
3.1	Gleich-Operator .....	7
3.2	Ungleich-Operator .....	7
3.3	Kleiner-Gleich-Operator .....	8
3.4	Größer-Gleich-Operator .....	8
3.5	Kleiner-Operator .....	9
3.6	Größer-Operator .....	9
3.7	%all_in%.....	10
3.8	%all_like% .....	11
3.9	%any_in%.....	12
3.10	%any_like% .....	13
3.11	%between%.....	14
3.12	%group_by%.....	14
4	Funktionen .....	15
4.1	maximum.....	15
4.2	minimum .....	15
4.3	replace_na .....	16
4.4	row_sums .....	16
4.5	to_year .....	17
4.6	sdat_get.....	18
4.7	sdat_list.....	19

4.8	sdatt_code, sdatt_aufndatum, sdatt_entlgrund, ...	20
5	Darstellung der Ergebnisse für Vergleichsoperatoren	22
6	Import und Kombination von Ergebnissen (über mehrere Leistungsbereiche hinweg)	24
6.1	Import von Ergebnissen anderer QIs und Kennzahlen	24
6.2	Summen von Ergebnissen anderer QIs und Kennzahlen	24
6.3	Quoten-Indikatoren - Zähler und Nenner aus unterschiedlichen Quellen	25
	Hilfsfunktionen	26
7	Nutzung weiterer Datenquellen	29
8	Follow-Up-Kennzahlen	30
8.1	Follow-Up Typ "Rate"	30
8.2	Follow-Up Typ "Hazardverhältnis"	32
8.3	Follow-Up Typ "Sentinel Event"	34
9	Hilfsfunktionen für O/E-Indikatoren	35
9.1	Ergebnis O-Kennzahl	35
9.2	Ergebnis E-Kennzahl	35
	Impressum	36

# 1 Einleitung

Zur Erhöhung der Transparenz und einfachen Reproduzierbarkeit der Ergebnisse durch Dritte werden die Rechenregeln der Qualitätsindikatoren und Kennzahlen des IQTIG seit dem Erfassungsjahr 2018 in R veröffentlicht. R ist eine offene und freie Softwareumgebung für statistische Berechnungen. Um die Lesbarkeit der Rechenregeln zu erhöhen, werden zusätzlich zu den standardmäßig in R vorhandenen Funktionen und Operatoren auch IQTIG-spezifische Variablen, Funktionen und Operatoren verwendet. Diese werden im vorliegenden Dokument beschrieben. Alle darüber hinaus in Rechenregeln verwendeten Funktionen sind Teil von Standard R.

Bei Fragen können Sie sich gern per E-Mail an den Verfahrenssupport wenden.

## Ihr Ansprechpartner:

Institut für Qualitätssicherung und Transparenz im Gesundheitswesen  
Katharina-Heinroth-Ufer 1  
10787 Berlin

Telefon: (+49) 30 58 58 26 340

Fax: (+49) 30 58 58 26 341

[verfahrenssupport@iqtig.org](mailto:verfahrenssupport@iqtig.org)

[www.iqtig.org](http://www.iqtig.org)

## 1.1 Grundlegende Konventionen

### 1.1.1 Datengrundlage

Generell wird bei der Berechnung davon ausgegangen, dass die jeweilige Datengrundlage als sogenannter *data frame* zur Verfügung steht. Einzelne Variablen beziehen sich somit immer auf eine komplette Spalte.

Beispiel:

ENTLGRUND	ENTLGRUND %==% "01"
"00"	FALSE
"01"	TRUE
"07"	FALSE

### 1.1.2 Mehrfachfelder

Felder, die in der Spezifikation mehrere Werte annehmen, wie z. B. OPSCHLUESSEL, werden innerhalb der Rechenregeln als eine Listenspalte dargestellt. So wird OPSCHLUESSEL\_1 = "A", OPSCHLUESSEL\_2 = "B", OPSCHLUESSEL\_3 = "C" zu einer einzelnen Spalte OPSCHLUESSEL mit dem Eintrag `list(c("A", "B", "C"))`.

Beispiel:

	FELDNAME	Beispiel INHALT
Richtige Abfrage	OPSCHLUESSEL	c("A", "B", "C")

### 1.1.3 Operatorpräzedenz

Um eine einheitliche Behandlung von fehlenden Werten zu gewährleisten, werden spezielle Vergleichsoperatoren verwendet. Diese unterscheiden sich einzig durch die Behandlung von fehlenden Werten (sog. NAs). Da diese eine höhere Präzedenz als die üblichen Vergleichsoperatoren haben, müssen bei Verwendung dieser Operatoren zusätzliche Klammern gesetzt werden.

Beispiel:

Ausdruck	Ergebnis
15 >= 10 + 8	FALSE (da „10+8=18“ und damit „Ist 15 größer als 18?“=Falsch )
15 %>=% 10 + 8	TRUE + 8 = 9

### 1.1.4 Vorfilter

Wenn im Zähler oder Nenner einer Kennzahl eine Formel in der folgenden Art steht, dann ist der Code nach dem „WENN“ als Vorfilter zu lesen.

Beispiel:

Ausdruck	Ergebnis
fn_funktionsname WENN fn_funktionsname %>% 0	Zuerst wird geprüft, ob die Bedingung nach dem „WENN“ erfüllt ist. Nur dann wird der Wert (im Beispiel: Ergebnis einer Funktion) vor dem „WENN“ übernommen, sonst nicht.

## 2 Variablen

### 2.1 VB

<b>VB</b>	<p>Eine Variable, über die Ergebnisse von Vorberechnungen abgefragt werden können.</p> <p>Dabei ist VB eine Variable, die in R als Liste oder data.frame angelegt werden kann. Sie enthält die Ergebnisse aller Vorberechnungen des jeweiligen QS-Verfahrens.</p>
Syntax	VB\$name
Parameter	name – der Name der Vorberechnung
Rückgabewert	Numeric Vektor in der Länge des Datensatzes
Beispiel	<p>to_year(OPDATUM) %==% VB\$Auswertungsjahr</p> <p>Erklärung: Prüfung, ob das Jahr aus dem OP-Datum dem Jahr aus der Vorberechnung mit dem Namen „Auswertungsjahr“ entspricht.</p>

### 2.2 LST

<b>LST</b>	<p>Eine Variable, über die alle Listen angesprochen werden können.</p> <p>Es handelt sich um eine Listenvariable, die alle Listen aus der QIDB als Character Vektoren enthält.</p>
Syntax	LST\$name
Parameter	name – der Name der Liste
Rückgabewert	Character Vektor, der die Elemente der Liste enthält
Beispiel	<p>Ansprechen der ICD-Liste „ICD_DekGrad_2“:</p> <p>LST\$ICD_DekGrad_2</p>

### 3 Operatoren

Zusätzlich zu Standard-R-Konstruktionen verwenden die Rechenregeln des IQTIG auch spezielle Funktionen und Operatoren (Benutzerdefinierte Operatoren). Diese Operatoren unterscheiden sich von den Standard-R-Operatoren dahingehend, dass NA-Werte anders behandelt werden.

#### 3.1 Gleich-Operator

%==%	Gleich-Operator								
Syntax	lhs %==% rhs								
Parameter	lhs – Ein Vektor rhs – Ein Vektor gleicher Länge wie <lhs>								
Rückgabewert	Logical Vektor mit der Länge gleich der Länge von <lhs>								
Besonderheiten	Dieser Operator unterscheidet sich von den Standard R Operatoren dahingehend, dass NA-Werte anders behandelt werden.								
Beispiele	<table border="1"> <tr> <td>2 %==% 2</td><td>TRUE</td></tr> <tr> <td>2 %==% 3</td><td>FALSE</td></tr> <tr> <td>2 %==% NA</td><td>FALSE</td></tr> <tr> <td>NA %==% NA</td><td>TRUE</td></tr> </table>	2 %==% 2	TRUE	2 %==% 3	FALSE	2 %==% NA	FALSE	NA %==% NA	TRUE
2 %==% 2	TRUE								
2 %==% 3	FALSE								
2 %==% NA	FALSE								
NA %==% NA	TRUE								

#### 3.2 Ungleich-Operator

%!=%	Ungleich-Operator								
Syntax	lhs %!=% rhs								
Parameter	lhs – Ein Vektor rhs – Ein Vektor gleicher Länge wie <lhs>								
Rückgabewert	Logical Vektor mit der Länge gleich der Länge von <lhs>								
Besonderheiten	Dieser Operator unterscheidet sich von den Standard R Operatoren dahingehend, dass NA-Werte anders behandelt werden.								
Beispiele	<table border="1"> <tr> <td>2 %!=% 1</td><td>TRUE</td></tr> <tr> <td>2 %!=% 2</td><td>FALSE</td></tr> <tr> <td>2 %!=% NA</td><td>TRUE</td></tr> <tr> <td>NA %!=% NA</td><td>FALSE</td></tr> </table>	2 %!=% 1	TRUE	2 %!=% 2	FALSE	2 %!=% NA	TRUE	NA %!=% NA	FALSE
2 %!=% 1	TRUE								
2 %!=% 2	FALSE								
2 %!=% NA	TRUE								
NA %!=% NA	FALSE								

### 3.3 Kleiner-Gleich-Operator

%<=%	Kleiner-Gleich-Operator								
Syntax	lhs %<=% rhs								
Parameter	lhs – Ein Vektor rhs – Ein Vektor gleicher Länge wie <lhs>								
Rückgabewert	Logical Vektor mit der Länge gleich der Länge von <lhs>								
Besonderheiten	Dieser Operator unterscheidet sich von den Standard R Operatoren dahingehend, dass NA-Werte anders behandelt werden.								
Beispiele	<table border="1"> <tr> <td>2 %&lt;=% 3</td><td>TRUE</td></tr> <tr> <td>2 %&lt;=% 1</td><td>FALSE</td></tr> <tr> <td>2 %&lt;=% NA</td><td>FALSE</td></tr> <tr> <td>NA %&lt;=% NA</td><td>TRUE</td></tr> </table>	2 %<=% 3	TRUE	2 %<=% 1	FALSE	2 %<=% NA	FALSE	NA %<=% NA	TRUE
2 %<=% 3	TRUE								
2 %<=% 1	FALSE								
2 %<=% NA	FALSE								
NA %<=% NA	TRUE								

### 3.4 Größer-Gleich-Operator

%>=%	Größer-Gleich-Operator								
Syntax	lhs %>=% rhs								
Parameter	lhs – Ein Vektor rhs – Ein Vektor gleicher Länge wie <lhs>								
Rückgabewert	Logical Vektor mit der Länge gleich der Länge von <lhs>								
Besonderheiten	Dieser Operator unterscheidet sich von den Standard R Operatoren dahingehend, dass NA-Werte anders behandelt werden.								
Beispiele	<table border="1"> <tr> <td>2 %&gt;=% 3</td><td>FALSE</td></tr> <tr> <td>2 %&gt;=% 1</td><td>TRUE</td></tr> <tr> <td>2 %&gt;=% NA</td><td>FALSE</td></tr> <tr> <td>NA %&gt;=% NA</td><td>TRUE</td></tr> </table>	2 %>=% 3	FALSE	2 %>=% 1	TRUE	2 %>=% NA	FALSE	NA %>=% NA	TRUE
2 %>=% 3	FALSE								
2 %>=% 1	TRUE								
2 %>=% NA	FALSE								
NA %>=% NA	TRUE								



### 3.5 Kleiner-Operator

%<%	Kleiner-Operator									
Syntax	lhs %<% rhs									
Parameter	lhs – Ein Vektor rhs – Ein Vektor gleicher Länge wie <lhs>									
Rückgabewert	Logical Vektor mit der Länge gleich der Länge von <lhs>									
Besonderheiten	Dieser Operator unterscheidet sich von den Standard R Operatoren dahingehend, dass NA-Werte anders behandelt werden.									
Beispiele	<table><tr><td>2 %&lt;% 3</td><td>TRUE</td></tr><tr><td>2 %&lt;% 1</td><td>FALSE</td></tr><tr><td>2 %&lt;% NA</td><td>FALSE</td></tr><tr><td>NA %&lt;% NA</td><td>FALSE</td></tr></table>		2 %<% 3	TRUE	2 %<% 1	FALSE	2 %<% NA	FALSE	NA %<% NA	FALSE
2 %<% 3	TRUE									
2 %<% 1	FALSE									
2 %<% NA	FALSE									
NA %<% NA	FALSE									

### 3.6 Größer-Operator

%>%	Größer-Operator								
Syntax	lhs %>% rhs								
Parameter	lhs – Ein Vektor rhs – Ein Vektor gleicher Länge wie <lhs>								
Rückgabewert	Logical Vektor mit der Länge gleich der Länge von <lhs>								
Besonderheiten	Dieser Operator unterscheidet sich von den Standard R Operatoren dahingehend, dass NA-Werte anders behandelt werden.								
Beispiele	<table><tr><td>2 %&gt;% 3</td><td>FALSE</td></tr><tr><td>2 %&gt;% 1</td><td>TRUE</td></tr><tr><td>2 %&gt;% NA</td><td>FALSE</td></tr><tr><td>NA %&gt;% NA</td><td>FALSE</td></tr></table>	2 %>% 3	FALSE	2 %>% 1	TRUE	2 %>% NA	FALSE	NA %>% NA	FALSE
2 %>% 3	FALSE								
2 %>% 1	TRUE								
2 %>% NA	FALSE								
NA %>% NA	FALSE								

### 3.7 %all\_in%

%all_in%	Prüft, ob alle Elemente der linken Seite in der Menge der rechten Seite vorhanden sind.									
Syntax	lhs %all_in% rhs									
Parameter	lhs – Ein Character Vektor oder eine Liste von Character Vektoren rhs – Ein Character Vektor									
Rückgabewert	Logical Vektor mit der Länge gleich der Länge von <lhs>									
Besonderheiten	Falls die linke Seite die Länge 0 hat (z. B. weil in einer Mehrfachauswahl kein Wert angegeben wurde), ist das Ergebnis immer TRUE.									
Beispiele	<pre>OPSCHLUESSEL_1 &lt;- "5-652.40" OPSCHLUESSEL_2 &lt;- "5-303.20" OPSCHLUESSEL_3 &lt;- NA OPS_GEB_VAG_OP &lt;- c("5-652.40", "5-303.20", "5-680.12")</pre> <table><tr><td>OPSCHLUESSEL %all_in% LST\$OPS_GEB_VAG_OP</td><td>TRUE</td></tr><tr><td>OPSCHLUESSEL %all_in% c("5-652.40", "5-303.20", "5-680.12")</td><td>TRUE</td></tr><tr><td>list(c("a", "b"), "c")) %all_in% c("a", "c")</td><td>c(FALSE, TRUE)</td></tr><tr><td>list(character(0)) %all_in% c("a", "b", "c")</td><td>TRUE</td></tr></table>		OPSCHLUESSEL %all_in% LST\$OPS_GEB_VAG_OP	TRUE	OPSCHLUESSEL %all_in% c("5-652.40", "5-303.20", "5-680.12")	TRUE	list(c("a", "b"), "c")) %all_in% c("a", "c")	c(FALSE, TRUE)	list(character(0)) %all_in% c("a", "b", "c")	TRUE
OPSCHLUESSEL %all_in% LST\$OPS_GEB_VAG_OP	TRUE									
OPSCHLUESSEL %all_in% c("5-652.40", "5-303.20", "5-680.12")	TRUE									
list(c("a", "b"), "c")) %all_in% c("a", "c")	c(FALSE, TRUE)									
list(character(0)) %all_in% c("a", "b", "c")	TRUE									

### 3.8 %all\_like%

<b>%all_like%</b>	Prüft, ob alle Elemente der linken Seite mit mindestens einem Element der rechten Seite „gematched“ werden können.				
Syntax	lhs %all_like% rhs				
Parameter	lhs – Ein Character Vektor oder eine Liste von Character Vektoren rhs – Ein Character Vektor. Die einzelnen Strings können Wildcard Operatoren "%" enthalten				
Rückgabewert	Logical Vektor mit der Länge gleich der Länge von <lhs>				
Besonderheiten	Es werden hier SQL-ähnliche reguläre Ausdrücke verwendet, die nur die Wildcard "%" kennen.  Falls die linke Seite die Länge 0 hat (z. B. weil in einer Mehrfachauswahl kein Wert angegeben wurde), ist das Ergebnis immer TRUE.				
Beispiele	<pre> OPSCHLUESSEL_1 &lt;- "5-12.32" OPSCHLUESSEL_2 &lt;- "5-33.20:L" OPSCHLUESSEL_3 &lt;- NA OPS_GEB_VAG_OP &lt;- c("5-12%", "5-33.2%", "5-652.40:R") </pre> <table border="1"> <tr> <td>OPSCHLUESSEL %all_like% LST\$OPS_GEB_VAG_OP</td><td>TRUE</td></tr> <tr> <td>list(character(0)) %all_like% c("a%", "b%", "c%")</td><td>TRUE</td></tr> </table>	OPSCHLUESSEL %all_like% LST\$OPS_GEB_VAG_OP	TRUE	list(character(0)) %all_like% c("a%", "b%", "c%")	TRUE
OPSCHLUESSEL %all_like% LST\$OPS_GEB_VAG_OP	TRUE				
list(character(0)) %all_like% c("a%", "b%", "c%")	TRUE				

3.9 %any\_in%

%any_in%	Prüft, ob mindestens ein Element der linken Seite in der Menge der rechten Seite vorhanden ist.	
Syntax	lhs %any_in% rhs	
Parameter	lhs – Ein Character Vektor oder eine Liste von Character Vektoren rhs – Ein Character Vektor	
Rückgabewert	Logical Vektor mit der Länge gleich der Länge von <lhs>	
Beispiele	OPSCHLUESSEL_1 <- "5-652.40:R" OPSCHLUESSEL_2 <- "5-744.5x" OPSCHLUESSEL_3 <- NA OPS_GEB_VAG_OP <- c("5-657.6x", "5-653.35", "5-652.40:R")	
	OPSCHLUESSEL %any_in% LST\$OPS_GEB_VAG_OP	TRUE
	OPSCHLUESSEL_1 <- "5-652.40" OPSCHLUESSEL_2 <- "5-744.5x" OPSCHLUESSEL_3 <- NA OPS_GEB_VAG_OP <- c("5-657.6x", "5-653.35", "5-652.40:R")	
	OPSCHLUESSEL %any_in% LST\$OPS_GEB_VAG_OP	FALSE
	list(c("a", "b", "c")) %any_in% c("a", "c")	c(TRUE, TRUE)

### 3.10 %any\_like%

<b>%any_like%</b>	Prüft, ob mindestens ein Element der linken Seite mit mindestens einem Element der rechten Seite „gematched“ werden kann.	
Syntax	lhs %any_like% rhs	
Parameter	lhs – Ein Character Vektor oder eine Liste von Character Vektoren rhs – Ein Character Vektor. Die einzelnen Strings können Wildcard Operatoren "%" enthalten	
Rückgabewert	Logical Vektor mit der Länge gleich der Länge von <lhs>	
Besonderheiten	Es werden hier SQL-ähnliche reguläre Ausdrücke verwendet, die nur die Wildcard "%" kennen.	
Beispiel	OPSCHLUESSEL_1 <- "5-652.40:R" OPSCHLUESSEL_2 <- "5-744.5x" OPSCHLUESSEL_3 <- NA OPS_GEB_VAG_OP <- c("5-657.6x", "5-653.35", "5-652.40%")	
	<div> <div>OPSCHLUESSEL %any_like% LST\$OPS_GEB_VAG_OP</div> <div>TRUE</div> </div>	

### 3.11 %between%

<b>%between%</b>	Prüft, ob Werte der linken Seite zwischen den zwei Werten der rechten Seite liegen.				
Syntax	lhs %between% rhs				
Parameter	lhs – Ein Vektor beliebigen Typs rhs – Ein Vektor der Länge 2 mit gleichem Typ wie <lhs>				
Rückgabewert	Logical Vektor mit der Länge gleich der Länge von <lhs>				
Besonderheiten	Äquivalent zu <code>lhs &gt;= rhs[1] &amp; lhs &lt;= rhs[2]</code> . D. h. der Wert in lhs kann auch auf den Grenzen des Intervalls liegen.  Dieser Operator liefert immer TRUE oder FALSE zurück, auch wenn lhs ein NA-Wert ist.				
Beispiele	<p>alter &lt;- 39</p> <table border="1"> <tr> <td>alter %between% c(40, 50)</td><td>FALSE</td></tr> <tr> <td>5 %between% c(1, 10)</td><td>TRUE</td></tr> </table>	alter %between% c(40, 50)	FALSE	5 %between% c(1, 10)	TRUE
alter %between% c(40, 50)	FALSE				
5 %between% c(1, 10)	TRUE				

### 3.12 %group\_by%

<b>%group_by%</b>	Evaluation einer Funktion, welche nach einer Menge von Feldern gruppiert wurde.
Syntax	lhs %group_by% rhs
Parameter	lhs – Code, der für jede Gruppe evaluiert wird. Er sollte typstabil sein und entweder einen Wert der Länge 1 oder einen Vektor in Länge der Anzahl Zeilen der jeweiligen Gruppe zurückgeben. rhs – Ein Name oder ein Vektor von Namen, die Spalten des globalen Datensatzes entsprechen.
Rückgabewert	Ein Vektor, abhängig vom Typ von <lhs>. Die Länge des Vektors ist immer gleich der Anzahl Zeilen des kompletten Datensatzes.
Besonderheiten	%group_by% bezieht sich immer implizit auf einen globalen Datensatz.
Beispiele	<p><code>minimum(OPDATUM) %group_by% TDS_B</code>  Erklärung: Ermittlung des kleinsten OP-Datums aller dem Basisbogen untergeordneter Bögen.</p> <p><code>minimum(OPDATUM[fn_IstErsteTxInAufenthalt]) %group_by% TDS_P</code>  Erklärung: Ermittlung des kleinsten OP-Datums der dem Patienten untergeordneten Bögen, für die die Funktion <code>fn_IstErsteTxInAufenthalt</code> erfüllt ist.</p> <p><code>minimum(OPDATUM) %group_by% c(TDS_P, bf_EntlassungsJahr)</code></p>

## 4 Funktionen

### 4.1 maximum

maximum	Ermittelt das Maximum eines Vektors.							
Syntax	maximum( <b>x</b> )							
Parameter	<b>x</b> – Ein Vektor vom Typ Integer, Character, Date, POSIXct, POSIXlt, oder Logical							
Rückgabewert	Vektor der Länge 1 vom Typ des Eingabe-Parameters							
Besonderheiten	<p>Unterschied zu base R:</p> <ul style="list-style-type: none"><li>- NA-Werte werden ignoriert</li><li>- Maximum eines leeren Vektors ist immer NA</li></ul> <p>Es wird versucht, typischer NAs zurück zu geben.</p> <p>Bei DateTime Typen wird die Zeitzone aus dem Attribut "tzone" auf einen möglichen NA-Rückgabewert angewendet.</p>							
Beispiele	<table><tr><td>maximum(c(1, 2, 3))</td><td>3</td></tr><tr><td>maximum(NA)</td><td>NA</td></tr><tr><td>maximum(character(0L))</td><td>NA_character</td></tr></table>		maximum(c(1, 2, 3))	3	maximum(NA)	NA	maximum(character(0L))	NA_character
maximum(c(1, 2, 3))	3							
maximum(NA)	NA							
maximum(character(0L))	NA_character							

### 4.2 minimum

<b>minimum</b>	Ermittelt das Minimum eines Vektors.							
Syntax	minimum( <b>x</b> )							
Parameter	<b>x</b> – Ein Vektor vom Typ Integer, Character, Date, POSIXct, POSIXlt, oder Logical							
Rückgabewert	Vektor der Länge 1 vom Typ den Eingabe-Parameters							
Besonderheiten	<p>Unterschied zu base R:</p> <ul style="list-style-type: none"><li>- NA-Werte werden ignoriert</li><li>- Maximum eines leeren Vektors ist immer NA</li></ul> <p>Es wird versucht, typischer NAs zurück zu geben.</p> <p>Bei DateTime Typen wird die Zeitzone aus dem Attribut "tzone" auf einen möglichen NA-Rückgabewert angewendet.</p>							
Beispiele	<table><tr><td>minimum(c(1, 2, 3))</td><td>1</td></tr><tr><td>minimum(NA)</td><td>NA</td></tr><tr><td>minimum(character(0L))</td><td>NA_character</td></tr></table>		minimum(c(1, 2, 3))	1	minimum(NA)	NA	minimum(character(0L))	NA_character
minimum(c(1, 2, 3))	1							
minimum(NA)	NA							
minimum(character(0L))	NA_character							

### 4.3 replace\_na

<b>replace_na</b>	Ersetzt NA-Werte durch einen anderen Wert.					
Syntax	replace_na( <b>x</b> , <b>by</b> )					
Parameter	<b>x</b> – Ein Vektor beliebiger Länge <b>by</b> – Ein Wert der Länge 1 mit gleichem Typ wie <x>					
Rückgabewert	Ein Vektor mit gleicher Länge und gleichem Typ wie <x>. Alle NA-Werte sind durch <by> ersetzt.					
Beispiele	<table><tr><td>replace_na(alter, 20)</td><td>Wenn Feld „Alter“ leer ist, wird es durch 20 ersetzt</td></tr><tr><td>replace_na(c(20, NA, 10), 15)</td><td>c(20, 15, 10)</td></tr></table>		replace_na(alter, 20)	Wenn Feld „Alter“ leer ist, wird es durch 20 ersetzt	replace_na(c(20, NA, 10), 15)	c(20, 15, 10)
replace_na(alter, 20)	Wenn Feld „Alter“ leer ist, wird es durch 20 ersetzt					
replace_na(c(20, NA, 10), 15)	c(20, 15, 10)					

### 4.4 row\_sums

<b>row_sums</b>	Zeilenweise Summe mehrerer Vektoren					
Syntax	row_sums(Vektor1, Vektor2, ..., VektorN)					
Parameter	... – Vektoren gleicher Länge					
Rückgabewert	Numeric Vektor					
Besonderheiten	NA-Werte werden entfernt. Äquivalent zu <code>rowSums(cbind(...), na.rm = TRUE)</code> Es können auch Logical Vektoren verarbeitet werden. TRUE wird dann bei der Summenbildung als 1 gezählt und FALSE als 0					
Beispiele	<div>GEHTRAINING = 1 HILFSMITTEL = 3 MEDIKATION = NA</div> <table><tr><td>row_sums(GEHTRAINING, HILFSMITTEL, MEDIKATION)</td><td>4</td></tr></table> <div>bf_Reizschwelle_VO = TRUE bf_Reizschwelle_VE1 = FALSE</div> <table><tr><td>row_sums(bf_Reizschwelle_VO, bf_Reizschwelle_VE1)</td><td>1</td></tr></table>		row_sums(GEHTRAINING, HILFSMITTEL, MEDIKATION)	4	row_sums(bf_Reizschwelle_VO, bf_Reizschwelle_VE1)	1
row_sums(GEHTRAINING, HILFSMITTEL, MEDIKATION)	4					
row_sums(bf_Reizschwelle_VO, bf_Reizschwelle_VE1)	1					



## 4.5 to\_year

<b>to_year</b>	Extrahiert die Jahreszahl aus einem Character Vektor							
Syntax	to_year( <b>x</b> )							
Parameter	<b>x</b> – Ein Vektor vom Typ Character, Date oder POSIXt (Uhrzeit)							
Rückgabewert	Ein Integer Vektor von Jahreszahlen oder NA_integer_, falls x NA war							
Besonderheiten	<p>Es werden drei Muster unterstützt:</p> <ul style="list-style-type: none"><li>▪ mm.yyyy</li><li>▪ [1-4]/yyyy</li><li>▪ yyyy</li></ul> <p>Sollten nicht alle Elemente des Vektors eines dieser Muster erfüllen, so wird angenommen, dass es sich bei &lt;x&gt; um ein <i>Date</i> handelt.</p>							
Beispiele	<table><tr><td>to_year("12.2015")</td><td>2015L</td></tr><tr><td>to_year("1.2014")</td><td>2014L</td></tr><tr><td>to_year(c(NA_character_, "2014"))</td><td>c(NA_integer_, 2014L)</td></tr></table>		to_year("12.2015")	2015L	to_year("1.2014")	2014L	to_year(c(NA_character_, "2014"))	c(NA_integer_, 2014L)
to_year("12.2015")	2015L							
to_year("1.2014")	2014L							
to_year(c(NA_character_, "2014"))	c(NA_integer_, 2014L)							

## 4.6 sdat\_get

<b>sdat_get</b>	Durchsucht die zu einem QS-Bogen zugeordneten Sozialdaten.	
Syntax	<code>sdat_get(x, from, filter = NULL)</code>	
Parameter	<ul style="list-style-type: none"> <li>▪ <b>x</b> – Der Name des Datenfeldes aus den Sozialdaten, das zurückgegeben werden soll</li> <li>▪ <b>from</b> – Ein Character Vektor mit Listennamen oder der Name einer konkreten Liste als Symbol. Dieser Parameter gibt an, welche Sozialdatenlisten durchsucht werden sollen</li> <li>▪ <b>filter</b> – Eine Filterbedingung, die die durchsuchten Sozialdaten-Informationen einschränkt</li> </ul>	
Rückgabewert	Eine Liste mit den Werten von x für die Fälle, die die Filterbedingungen erfüllen.	
Besonderheiten	Zu jedem QS-Bogen sind i. d. R. alle Sozialdaten-Informationen der betreffenden Patientin bzw. des betreffenden Patienten zugeordnet.	
Beispiele	<code>sdat_get("fall_id", sdat_301_ops, code %any_like% c("5.12%", "5.13%"))</code>	Gibt die IDs sämtlicher Fälle der Patientin bzw. des Patienten zurück, die einen OPS-Kode enthalten, der mit "5.12" oder "5.13" beginnt
	<code>lengths(sdat_get("fall_id", sdat_301_ops, code %any_like% LST\$OPS_CHE_Reintervention)) &gt; 0L</code>	Wurde mindestens ein OPS-Kode aus der Liste LST\$OPS_CHE_Reintervention in den Sozialdaten dokumentiert?
	<code>sdat_get("fall_id", sdat_301_ops, (datum &lt;= OPDATUM + 90) &amp; (datum &gt; OPDATUM) &amp; (code %any_like% LST\$OPS_CHE_Reintervention))</code>	Gibt die IDs sämtlicher Fälle zurück, bei denen ein OPS-Kode aus der Liste LST\$OPS_CHE_Reintervention im Zeitraum von 90 Tagen nach OPDATUM dokumentiert wurde.
	<code>sdat_get("code", "301_icd", (datum &gt;= ENTLDATUM) &amp; (datum &lt;= OPDATUM + 7))</code>	Eine Liste von ICD-Kodes nach §301 SGB V, welche ein Diagnose-datum zwischen (jeweils einschließlich) dem Entlassdatum und OP-Datum + 7 Tage haben. Äquivalent zu: <code>sdat_list("301_icd", ENTLDATUM, OPDATUM + 7)</code>

## 4.7 sdat\_list

<b>sdat_list</b>	Schränkt Sozialdatenkelisten der zu einem QS-Bogen zugeordneten Sozialdaten nach Datums- und sonstigen Filtern ein.	
Syntax	<code>sdat_list(list_names, date_from, date_end, filter = NULL)</code>	
Parameter	<ul style="list-style-type: none"> <li>• <b>list_names</b> – Ein Character Vektor mit Listennamen oder der Name einer konkreten Liste als Symbol. Dieser Parameter gibt an, welche Sozialdatenlisten eingeschränkt werden sollen.</li> <li>• <b>date_from</b> – Jeder Sozialdatenkode hat ein Datum. Dieses Datum muss gleich date_from oder später sein.</li> <li>• <b>date_end</b> – Jedes Datum muss früher oder gleich date_end sein.</li> <li>• <b>filter</b> – Neben dem Datum können Sozialdaten noch weitere Eigenschaften wie z. B. die Diagnoseart besitzen. Die interessierenden Kodes können über ein optionales Filterargument eingeschränkt werden.</li> </ul>	
Rückgabewert	Eine Liste mit Kodes, die die Bedingungen (Datumseinschränkungen und optionaler Filter) erfüllen. Diese Kodelisten werden dann in der Regel mit ICD/OPS Listen abgeglichen.	
Besonderheiten	Zu jedem QS-Bogen sind i. d. R. alle Sozialdaten-Informationen der betreffenden Patientin bzw. des betreffenden Patienten zugeordnet.	
Beispiele	<code>sdat_list("301_icd", ENTL DATUM, OPDATUM + 7)</code>	Eine Liste von ICD-Kodes nach §301 SGB V, welche ein Diagnose-datum zwischen (jeweils einschließlich) dem Entlassdatum und OP-Datum + 7 Tage haben.
	<code>sdat_list(sdat_301_icd, ENTL DATUM, OPDATUM + 7)</code>	Äquivalent zum Ausdruck oben. Die Verwendung der Variable als Symbol hat in der Regel technische Gründe.
	<code>sdat_list("301_ops", OPDATUM, OPDATUM + 7) %any_like% LST\$OPS_PCI_Erneut</code>	Jeder OPS-Kode, der die Datumseinschränkung erfüllt, wird mit einer Liste von OPS-Kodes verglichen. Ist der Kode Teil dieser Liste, so wird TRUE zurückgegeben, ansonsten FALSE.
	<code>sdat_list("301_icd", OPDATUM, OPDATUM + 7, diagnoseart %==% 1 ) %any_like% LST\$ICD_PCI_-AneurysmaSpurium</code>	Ähnlich zum Beispiel oben, allerdings müssen die ICD-Kodes noch die Einschränkung "diagnoseart %==% 1" erfüllen, bevor sie mit der entsprechenden ICD-Liste abgeglichen werden.

## 4.8 **sdat\_code, sdat\_aufndatum, sdat\_entlgrund,...**

<b>sdat_code, etc</b>	Schränkt Informationen aus den zu einem QS-Bogen zugeordneten Sozialdaten nach einer Filterbedingung ein. Dabei gibt sdat_code die Codes zurück, sdat_aufndatum die Aufnahmezeiten usw.
Syntax	<pre>sdat_code(<b>from</b>, <b>filter</b> = NULL) sdat_aufndatum(<b>from</b>, <b>filter</b> = NULL) sdat_entldatum(<b>from</b>, <b>filter</b> = NULL) ...</pre>
Parameter	<ul style="list-style-type: none"> <li>• <b>from</b> – Der Name der Sozialdatenliste als Symbol, zum Beispiel sdat_301_icd. Mehrere Listen können mit einem Ausdruck der Form c(sdat_301_icd, sdat_301_icd_sek) angegeben werden.</li> <li>• <b>filter</b> – Eine Filterbedingung, die die durchsuchten Sozialdaten-Informationen einschränkt. Dabei wird NA wie FALSE gewertet.</li> </ul>
Rückgabewert	Eine Liste mit denjenigen Werten aus den Sozialdaten (Codes, Aufnahmezeiten, ...) aus <b>from</b> , die die Filterbedingung <b>filter</b> erfüllen.
Besonderheiten	Zu jedem QS-Bogen sind i. d. R. alle Sozialdaten-Informationen der betreffenden Patientin bzw. des betreffenden Patienten zugeordnet.

Beispiele	<pre>sdat_code(sdat_301_icd, aufndatum %&gt;=% OPDATUM &amp; aufndatum %&lt;=% (OPDATUM + 7))</pre>	<p>Eine Liste aller ICD-Kodes (Primär-diagnosen) nach §301 SGB V, die in Fällen mit Aufnahmedatum zwischen (jeweils einschließlich) OP-Datum und OP-Datum + 7 Tage haben. Dabei ist bei Nutzung von "%&lt;=%" statt "&lt;=" unbedingt die Klammerung um "OPDATUM + 7" zu beachten.</p>
	<pre>sdat_code(c(sdat_301_amb_icd, sdat_301_amb_icd_sek), aufndatum %&gt;=% OPDATUM &amp; aufndatum %&lt;=% (OPDATUM + 7))</pre>	<p>Ähnlich oben, aber die ICD-Kodes stammen aus ambulanten Behandlungen im Krankenhaus und es werden sowohl Primär- als auch Sekundärdiagnosen ausgegeben.</p>
	<pre>sdat_code(sdat_301_ops, datum %&gt;=% OPDATUM &amp; datum %&lt;=% (OPDATUM + 7)) %any_like% LST\$OPS_PCI_Erneut</pre>	<p>Ähnlich oben, diesmal mit OPS-Kodes. Typische Verwendung gemeinsam mit dem Abgleich, ob Kodes aus einer bestimmten Liste vorlagen. Ist einer der gefundenen Kodes Teil der Liste OPS_PCI_Erneut, wird TRUE zurückgegeben, ansonsten FALSE.</p>
	<pre>sdat_entlgrund(sdat_301_icd, ent- ldatum %&gt;% ENTLDATUM &amp; entlda- tum %&lt;=% (fn_IndexEingrDatum + 90))</pre>	<p>Eine Liste der Entlassungsgründe zu den in der Liste sdat_301_icd enthaltenen Aufenthalten mit Entlassungsdatum nach dem Indexaufenthalt und innerhalb von 90 Tagen nach dem Indexeingriff.</p>
	<pre>sdat_fall_id(sdat_295_ops, code %any_like% c("5.12%", "5.13%"))</pre>	<p>Gibt die IDs sämtlicher Fälle der Patientin bzw. des Patienten zurück, die einen OPS-Kode nach §295 enthalten, der mit "5.12" oder "5.13" beginnt.</p>

## 5 Darstellung der Ergebnisse für Vergleichsoperatoren

Die folgenden Tabellen zeigen beispielhaft, wie aktuell bei der Berechnung von Qualitätsindikatoren mit NA-Werten umgegangen wird. Es gilt für alle Tabellen

- a und b sind vom Typ *Numeric* und
- „a ist kleiner b“

%==%	a	b	NA
a	TRUE	FALSE	FALSE
b	FALSE	TRUE	FALSE
NA	FALSE	FALSE	TRUE

%!=%	a	b	NA
a	FALSE	TRUE	TRUE
b	TRUE	FALSE	TRUE
NA	TRUE	TRUE	FALSE

%<=%	a	b	NA
a	TRUE	TRUE	FALSE
b	FALSE	TRUE	FALSE
NA	FALSE	FALSE	TRUE

%>=%	a	b	NA
a	TRUE	FALSE	FALSE
b	TRUE	TRUE	FALSE
NA	FALSE	FALSE	TRUE

<b>%&lt;%</b>	<b>a</b>	<b>b</b>	<b>NA</b>
<b>a</b>	FALSE	TRUE	FALSE
<b>b</b>	FALSE	FALSE	FALSE
<b>NA</b>	FALSE	FALSE	FALSE

<b>%&gt;%</b>	<b>a</b>	<b>b</b>	<b>NA</b>
<b>a</b>	FALSE	FALSE	FALSE
<b>b</b>	TRUE	FALSE	FALSE
<b>NA</b>	FALSE	FALSE	FALSE

## 6 Import und Kombination von Ergebnissen (über mehrere Leistungsbereiche hinweg)

Mit den folgenden Funktionen können QIs und Kennzahlen definiert werden, die Ergebnisse anderer Kennzahlen direkt weiterverwenden.

### 6.1 Import von Ergebnissen anderer QIs und Kennzahlen

<b>import_indicator</b>	Importiert das Ergebnis eines anderen QIs bzw. einer anderen Kennzahl.
Syntax	<code>import_indicator(module, id)</code>
Parameter	<b>module</b> – Name des Leistungsbereiches, aus dem der QI bzw. die Kennzahl importiert werden soll (Character). <b>id</b> – ID des zu importierenden QIs bzw. der zu importierenden Kennzahl (Character).
Rückgabewert	Das Ergebnis des importierten QIs bzw. der importierten Kennzahl als <code>data.frame</code> .
Beispiel	Risikostatistik: Anzahl aller Fälle: <code>import_indicator(module = "RST", id = "24851")</code>

### 6.2 Summen von Ergebnissen anderer QIs und Kennzahlen

<b>sum_indicator</b>	Summiert die Ergebnisse anderer QIs bzw. anderer Kennzahlen auf.
Syntax	<code>sum_indicator(list(module, id), list(module, id), ...)</code>
Parameter	<b>list(module, id)</b> – Eine beliebige Anzahl von Referenzen auf andere QIs bzw. Kennzahlen. Die Parameter <code>module</code> und <code>id</code> sind dabei analog zu den Parametern von <code>import_indicator</code> definiert.
Rückgabewert	Die Summe der Ergebnisse der angegebenen QIs bzw. Kennzahlen. Dabei werden Zähler und Nenner getrennt aufsummiert. Das Ergebnis wird dann als Quotient von Zähler und Nenner berechnet. Alle Leistungserbringer, die mindestens für ein Element ein Ergebnis aufweisen, werden Teil des Ergebnissdatensatzes.
Beispiel	ID 52307 (09/2) – Index: <code>sum_indicator(   list(module = "09/1", id = "52312"),   list(module = "09/2", id = "52313"),   list(module = "09/3", id = "52314") )</code>



### 6.3 Quoten-Indikatoren – Zähler und Nenner aus unterschiedlichen Quellen

<b>quotient_indicator</b>	Quoten-Indikator, dessen Zähler und Nenner aus unterschiedlichen Berechnungen und aus unterschiedlichen Datensätzen stammen.										
Syntax	<code>quotient_indicator(numerator, denominator, expected = NULL, units_from = "union")</code>										
Parameter	<p><b>numerator</b> – Zähler-Ergebnisse auf Leistungserbringer-Ebene in einem data.frame. Die weiter unten aufgeführten Hilfsfunktionen stehen hierfür zur Verfügung.</p> <p><b>denominator</b> – Nenner-Ergebnisse auf Leistungserbringer-Ebene in einem data.frame. Die weiter unten aufgeführten Hilfsfunktionen stehen hierfür zur Verfügung.</p> <p><b>expected</b> – Ergebnisse für die erwartete Anzahl interessierender Ereignisse (E) in einem data.frame (optional). Die weiter unten aufgeführten Hilfsfunktionen stehen hierfür zur Verfügung.</p> <p><b>units_from</b> – Gibt an, für welche Leistungserbringer der Quoten-Indikator Ergebnisse liefert (Character).</p> <p>Möglich sind:</p> <table border="1"> <thead> <tr> <th>Wert</th><th>Bedeutung</th></tr> </thead> <tbody> <tr> <td>"union"</td><td>Default. Alle Leistungserbringer, für die entweder in denominator ein Ergebnis für den Nenner oder in numerator ein Ergebnis für den Zähler vorliegt (entspricht full join).</td></tr> <tr> <td>"intersection"</td><td>Alle Leistungserbringer, für die sowohl in denominator ein Ergebnis für den Nenner als auch in numerator ein Ergebnis für den Zähler vorliegt (entspricht inner join).</td></tr> <tr> <td>"denominator"</td><td>Alle Leistungserbringer, für die in denominator ein Ergebnis für den Nenner vorliegt (entspricht left bzw. right join).</td></tr> <tr> <td>"numerator"</td><td>Alle Leistungserbringer, für die in numerator ein Ergebnis für den Zähler vorliegt (entspricht right bzw. left join).</td></tr> </tbody> </table>	Wert	Bedeutung	"union"	Default. Alle Leistungserbringer, für die entweder in denominator ein Ergebnis für den Nenner oder in numerator ein Ergebnis für den Zähler vorliegt (entspricht full join).	"intersection"	Alle Leistungserbringer, für die sowohl in denominator ein Ergebnis für den Nenner als auch in numerator ein Ergebnis für den Zähler vorliegt (entspricht inner join).	"denominator"	Alle Leistungserbringer, für die in denominator ein Ergebnis für den Nenner vorliegt (entspricht left bzw. right join).	"numerator"	Alle Leistungserbringer, für die in numerator ein Ergebnis für den Zähler vorliegt (entspricht right bzw. left join).
Wert	Bedeutung										
"union"	Default. Alle Leistungserbringer, für die entweder in denominator ein Ergebnis für den Nenner oder in numerator ein Ergebnis für den Zähler vorliegt (entspricht full join).										
"intersection"	Alle Leistungserbringer, für die sowohl in denominator ein Ergebnis für den Nenner als auch in numerator ein Ergebnis für den Zähler vorliegt (entspricht inner join).										
"denominator"	Alle Leistungserbringer, für die in denominator ein Ergebnis für den Nenner vorliegt (entspricht left bzw. right join).										
"numerator"	Alle Leistungserbringer, für die in numerator ein Ergebnis für den Zähler vorliegt (entspricht right bzw. left join).										
Rückgabewert	Das Ergebnis des Quoten-Indikators data.frame.										

## Hilfsfunktionen

Die folgenden Hilfsfunktionen können verwendet werden, um die Parameter denominator und numerator zu bestimmen. Sie unterscheiden sich von den obigen Funktionen technisch dahingehend, dass sie stets ein data.frame auf Leistungserbringer-Ebene (d. h. eine Zeile pro Leistungserbringer) zurückgeben. Die Ermittlung eines Gesamtergebnisses durch Zusammenfügen von Zähler und Nenner und Summation findet erst nachgelagert in der Funktion quotient\_indicator statt. Je nachdem, ob das Ergebnis in quotient\_indicator als Zähler, Nenner oder E eingebunden wird, findet auch nur dieser Teil des Ergebnisses dort Verwendung.

<b>evaluate</b>	Wertet eine Rechenregel auf dem Datensatz des aktuellen Moduls mit dem Teildatensatzbezug des aktuellen Indikators aus.
Syntax	evaluate( <b>code</b> )
Parameter	<b>code</b> – Die zu evaluierende Rechenregel
Rückgabewert	Das Ergebnis der Ausführung der Rechenregel auf Leistungserbringer-Ebene (Anzahl Teildatensätze, für die die Rechenregel auf TRUE evaluiert).
Beispiel	<p><b>ID 52009 (DEK) – Stationär erworbener Dekubitalulcus (ohne Dekubitalulcera Stadium/Kategorie 1):</b></p> <pre> numerator &lt;- evaluate(   fn_DEKDatensatzPlausibel &amp; fn_DEKStadium_2bis4 &amp;   fn_DEKnichtPOA)  denominator &lt;- import_results(module = "RST",   id = "24851_52009")  expected &lt;- import_results(module = "RST",   id = "23014_52009")  quotient_indicator(numerator = numerator,   denominator = denominator,   expected = expected,   units_from = "denominator" ) </pre>

<b>import_results</b>	Importiert das Ergebnis eines anderen QIs bzw. einer anderen Kennzahl auf Leistungserbringer-Ebene.
Syntax	<code>import_results(module, id)</code>
Parameter	<p><b>module</b> – Name des Leistungsbereiches, aus dem der QI bzw. die Kennzahl importiert werden soll (Character).</p> <p><b>id</b> – ID des zu importierenden QIs bzw. der zu importierenden Kennzahl (Character).</p>
Rückgabewert	Das Ergebnis des importierten QIs bzw. der importierten Kennzahl als <code>data.frame</code> auf Leistungserbringer-Ebene.
Beispiel	<p><b>ID 52010 (DEK) – Stationär erworbener Dekubitalulcus Stadium/Kategorie 4:</b></p> <pre> numerator &lt;- evaluate(   fn_DEKDatensatzPlausibel &amp; fn_DEKStadium_4 &amp;   fn_DEKnichtPOA)  denominator &lt;- import_results(module = "RST",   id = "24852_52010")  quotient_indicator(numerator = numerator,   denominator = denominator,   units_from = "union" ) </pre>

<b>sum_results</b>	Summiert Ergebnisse auf Leistungserbringer-Ebene auf.
Syntax	sum_results(...)
Parameter	... – Eine beliebige Anzahl an Leistungserbringer-Ergebnissen als data.frames. Diese können beispielsweise mit import_results erzeugt werden.
Rückgabewert	Die Summe der Ergebnisse der angegebenen QIs bzw. Kennzahlen als data.frame auf Leistungserbringer-Ebene. Dabei werden Zähler und Nenner getrennt aufsummiert.
Beispiel	<p><b>ID 851805 (DEK) – Relative Differenz zwischen den Angaben in der QS-Dokumentation und der Risikostatistik: mehr Dekubitalulcera in der QS-Dokumentation als in der Risikostatistik:</b></p> <pre> dekubitus_qs &lt;- evaluate(   fn_DEKDatensatzPlausibel &amp; fn_DEKStadium_2bis4)  dekubitus_rst_negativ &lt;- import_results(module = "RST",   id = "321800_851805")  quotient_indicator(   numerator = sum_results(dekubitus_qs, dekubitus_rst_negativ),   denominator = dekubitus_qs,   units_from = "denominator" ) </pre>

## 7 Nutzung weiterer Datenquellen

Hilfsfunktion, um Tabellen aus der Datenbank zu laden. Ermöglicht das Einbeziehen weiterer Datenquellen zusätzlich zur Standarddatentabelle des jeweiligen QS-Verfahrens.

<b>get_dataset_by_name</b>	Lädt eine Tabelle aus dem zur Berechnung genutzten Datenpool.
Syntax	<code>get_dataset_by_name(<b>table_name</b>)</code>
Parameter	<b>table_name</b> – Name der Tabelle
Rückgabewert	Die gewählte Tabelle als data.frame.

## 8 Follow-Up-Kennzahlen

Mit den folgenden Funktionen werden die Follow-Up-Kennzahlen aus den Verfahren HSMDEF-HSM-IMPL, HSMDEF-DEFI-IMPL, HGV-HEP und KEP abgebildet. Details zur Implementierung der statistischen Methodik sind in: Das IQTIG - Empfehlungen - Biometrische Grundlagen - Biometrische Methodik bei Follow-up-Kennzahlen[Ereigniszeitanalyse-Methodik für die Follow-up-Indikatoren nach QSKH-RL. Stand: 06.04.2017. Berlin: IQTIG. URL: [https://iqtig.org/dateien/berichte/2017/IQTIG\\_Ereigniszeitanalyse-Methodik-fuer-Follow-up-Indikatoren-nach-QSKH-RL\\_2017-04-06.pdf](https://iqtig.org/dateien/berichte/2017/IQTIG_Ereigniszeitanalyse-Methodik-fuer-Follow-up-Indikatoren-nach-QSKH-RL_2017-04-06.pdf) ] veröffentlicht.

### 8.1 Follow-Up Typ "Rate"

Die Berechnung von Follow-Up-Raten unterscheidet sich von "normalen" Raten dadurch, dass unvollständige Beobachtungen des Follow-Up-Zeitraums (genannt Rechtszensierung) berücksichtigt werden müssen. Dazu wird der Kaplan-Meier Schätzer verwendet. Eine Follow-Up-Ratenberechnung mittels Kaplan-Meier-Schätzer wird durch die Funktion `follow_up_rate` umgesetzt.

Neben Rechtszensierung sind einige Follow-Up-Ratenkennzahlen auch von Linkstrunkierung betroffen. Dies geschieht bei der Periodensichtweise, wenn Patienten, die vor Beginn der Periode ihr Indexereignis und noch kein Folgeereignis hatten, in die Berechnung eingeschlossen werden. Zum korrekten Umgang mit Linkstrunkierung wird die Funktion `calc_follow_up_rate` verwendet. `calc_follow_up_rate` kann überall angewandt werden, wo `follow_up_rate` verwendet wird. Die Funktion `follow_up_rate` wurde vor allem in der Vergangenheit verwendet und wird aus Reproduzierbarkeitsgründen beibehalten.

<b>follow_up_rate</b>	Gibt die Ergebnisse eines Follow-Up-Indikators vom Typ "Rate" zurück.
Syntax	<code>follow_up_rate(dataset, denominator, numerator, method = "Kohortensichtweise")</code>
Parameter	<b>dataset</b> – Der der Follow-Up-Berechnung zu Grunde liegende Datensatz. <b>denominator</b> – Die Nenner-Rechenregel als Code. <b>numerator</b> – Die Zähler-Rechenregel als Code. Darin muss ein Term der Form Beobachtungszeit %<=% ... enthalten sein, der den Follow-Up-Zeitraum spezifiziert. <b>method</b> – Die Art der Follow-up-Auswertung: "Kohortensichtweise" oder "Periodensichtweise".
Rückgabewert	Das Ergebnis des Follow-Up-QIs als <code>data.frame</code> .
Beispiel	<b>ID 54128 (KEP) – Knieendoprothesen-Erstimplantation ohne Wechsel bzw. Komponentenwechsel im Verlauf:</b> <code>AJ &lt;- VB\$Auswertungsjahr[[1]]</code>

	<pre>follow_up_rate( dataset = get_dataset_by_name("FUMKEP"), denominator = OPDATUM %&gt;=% (as.Date(paste0(AJ, '-01-01')) - 90), numerator = Beobachtungszeit %&lt;=% 90 &amp; to_year(FU_OPDATUM) %==% AJ, method = "Periodensichtweise")</pre>
--	---

<b>calc_follow_up_rate</b>	Gibt die Ergebnisse eines Follow-Up-Indikators vom Typ "Rate" zurück.
Syntax	<pre>calc_follow_up_rate(dataset, numerator, denominator,                     left_truncation_date, right_censoring_date,                     fu_period, fu_date = 'FU_OPDATUM',                     index_date = 'OPDATUM',                     count_non_events = FALSE)</pre>
Parameter	<p><b>dataset</b> – Datensatz</p> <p><b>numerator</b> – Inhaltliche Definition des Folgeereignisses.</p> <p><b>denominator</b> – Inhaltliche Nennerdefinition.</p> <p><b>left_truncation_date</b> – Zeitpunkt, ab dem Folgeereignisse gezählt werden.</p> <p><b>right_censoring_date</b> – Zeitpunkt, bis wann Folgeereignisse gezählt werden.</p> <p><b>fu_period</b> – Maximale Länge der Beobachtungszeit für einen einzelnen Patienten.</p> <p><b>fu_date</b> – Spaltenname für die Datumsangabe des Folgeereignisses.</p> <p><b>index_date</b> – Spaltenname für die Datumsangabe des Index-Ereignisses.</p> <p><b>count_non_events</b> – TRUE, wenn gezählt werden soll, wie viele Patienten keine Folgeereignisse bekommen haben, sonst FALSE.</p>
Rückgabewert	Das Ergebnis des Follow-Up-QIs als data.frame.
Beispiel	<p><b>ID 54128 (KEP) – Knieendoprothesen-Erstimplantation ohne Wechsel bzw. Komponentenwechsel innerhalb von 90 Tagen:</b></p> <pre>EJ &lt;- VB\$Erfassungsjahr[[1]]  calc_follow_up_rate( dataset = get_dataset_by_name("FUMKEP"), denominator = VERSORGPOLY %!=% 1 &amp; FU_VERSORGPOLY %!=% 1, numerator = !is.na(FU_OPDATUM), left_truncation_date = paste0(EJ, '-01-01'), right_censoring_date = paste0(EJ, '-12-31'), fu_period = 90, count_non_events = TRUE )</pre>

## 8.2 Follow-Up Typ "Hazardverhältnis"

Die beiden Funktionen `follow_up_oe` und `calc_hazard_ratio` berechnen das Hazardverhältnis einer Population im Vergleich zu einer Referenzpopulation, optional unter Berücksichtigung relevanter Risikofaktoren. Die ältere Funktion `follow_up_oe` wird aus Gründen der Reproduzierbarkeit weiterhin dokumentiert, jedoch nicht mehr für neue Auswertungen eingesetzt. Ihr Funktionsname ist irreführend, da sie nicht das Verhältnis von beobachteten und erwarteten Ereignissen berechnet, sondern auch ein Hazardverhältnis. Für aktuelle Analysen wird daher die Funktion `calc_hazard_ratio` verwendet. Im Unterschied zur älteren Variante berechnet diese die erforderlichen Beobachtungszeiten intern, sodass diese nicht mehr als separate Spalten im Datensatz bereitgestellt werden müssen.

<b>follow_up_oe</b>	Gibt die Ergebnisse eines Follow-Up-Indikators vom Typ "O/E" zurück.
Syntax	<code>follow_up_oe(dataset, denominator, numerator, expected_events, method = "Kohortensichtweise")</code>
Parameter	<p><b>dataset</b> – Der der Follow-Up-Berechnung zu Grunde liegende Datensatz. Der Datensatz muss eine Spalte <code>expected_events</code> enthalten oder der Parameter <code>"expected_events"</code> im Funktionsaufruf muss eine Spalte für die Risikoadjustierung definieren.</p> <p><b>denominator</b> – Die Nenner-Rechenregel als Code.</p> <p><b>numerator</b> – Die Zähler-Rechenregel als Code. Darin muss ein Term der Form <code>Beobachtungszeit %&lt;=% ...</code> enthalten sein, der den Follow-Up-Zeitraum spezifiziert.</p> <p><b>expected_events</b> – Der Name der für die Risikoadjustierung definierten Spalte als Character.</p> <p><b>method</b> – Die Art der Follow-up-Auswertung: <code>"Kohortensichtweise"</code> oder <code>"Periodensichtweise"</code>.</p>
Rückgabewert	Das Ergebnis des Follow-Up-QIs als <code>data.frame</code> .
Beispiel	<p><b>ID 10271 (HGV-HEP) – Hüftendoprothesen-Wechsel bzw. -Komponentenwechsel im Verlauf:</b></p> <pre>AJ &lt;- VB\$Auswertungsjahr[[1]]  follow_up_oe(   dataset = get_dataset_by_name("FUMHEP"),   denominator = OPDATUM %&gt;= %     (as.Date(paste0(AJ, '-01-01')) - 90),   numerator = Beobachtungszeit %&lt;=% 90 &amp;     to_year(FU_OPDATUM) %==% AJ,   expected_events = "expected_events_10271",   method = "Periodensichtweise" )</pre>



<b>calc_hazard_ratio</b>	Gibt die Ergebnisse eines Follow-Up-Indikators zurück, der das Hazardverhältnis ggf. unter Berücksichtigung von Risikofaktoren berechnet.
Syntax	<code>calc_hazard_ratio(dataset, denominator, numerator,  score,  left_truncation_date, right_censoring_date,  fu_period, fu_date = 'FU_OPDATUM',  index_date = 'OPDATUM')</code>
Parameter	<p><b>dataset</b> – Datensatz</p> <p><b>denominator</b> – Inhaltliche Nennerdefinition.</p> <p><b>numerator</b> – Inhaltliche Definition des Folgeereignisses.</p> <p><b>score</b> – Spaltenname für die Scores.</p> <p><b>left_truncation_date</b> – Zeitpunkt, ab dem Folgeereignisse gezählt werden.</p> <p><b>right_censoring_date</b> – Zeitpunkt, bis wann Folgeereignisse gezählt werden.</p> <p><b>fu_period</b> – Maximale Länge der Beobachtungszeit für einen einzelnen Patienten.</p> <p><b>fu_date</b> – Spaltenname für die Datumsangabe des Folgeereignisses.</p> <p><b>index_date</b> – Spaltenname für die Datumsangabe des Index-Ereignisses.</p>
Rückgabewert	Das Ergebnis des Follow-Up-QIs als data.frame.
Beispiel	<p><b>ID 10271 (HGV-HEP) – Hüftendoprothesen-Wechsel bzw. -Komponentenwechsel im Verlauf:</b></p> <p>EJ &lt;- VB\$Erfassungsjahr[[1]]</p> <pre>calc_hazard_ratio( dataset = get_main_module_data(), denominator = INITIALACETAB %!=% 1 &amp; VERSORGPOLY %!=% 1 &amp; FU_VERSORGPOLY %!=% 1, numerator = TRUE, score = "fn_HEPScore_10271", left_truncation_date = paste0(EJ, '-01-01'), right_censoring_date = paste0(EJ, '-12-31'), fu_period = 90, fu_date = "FU_OPDATUM", index_date = "OPDATUM" )</pre>

### 8.3 Follow-Up Typ "Sentinel Event"

<b>follow_up_sentinel</b>	Gibt die Ergebnisse eines Follow-Up-Indikators vom Typ "Sentinel-Event" zurück.
Syntax	<code>follow_up_sentinel(dataset, denominator, numerator)</code>
Parameter	<p><b>dataset</b> – Der der Follow-Up-Berechnung zu Grunde liegende Datensatz.</p> <p><b>denominator</b> – Die Nenner-Rechenregel als Code.</p> <p><b>numerator</b> – Die Zähler-Rechenregel als Code. Darin muss ein Term der Form Beobachtungszeit %&lt;=% ... enthalten sein, der den Follow-Up-Zeitraum spezifiziert.</p>
Rückgabewert	Das Ergebnis des Follow-Up-QIs als data.frame.
Beispiel	<p><b>ID 2190 (HSMDEF-HSM-IMPL) – Laufzeit des alten Herzschrittmacher-Aggregats unter 4 Jahren bei Ein- und Zweikammersystemen:</b></p> <pre>AJ &lt;- VB\$Auswertungsjahr[[1]]  follow_up_sentinel( dataset = get_dataset_by_name("FUM09N1"), denominator = ENTLGRUND %!="% 07" &amp; ASMSYSTEMPO %in% c(1,2,3,4,7) &amp; Beobachtungszeit %&gt;% 0, numerator = to_year(FU_OPDATUM) %==% AJ &amp; (FU_AGGWECHSIND %==% 1   FU_AGGREGATPROBLEM %==% 1) &amp; FU_ORTLETZTEOP %==% 1 &amp; Beobachtungszeit %&lt;=% 1460 )</pre>

## 9 Hilfsfunktionen für O/E-Indikatoren

Hilfsfunktionen, um aus Ergebnissen von O/E-Indikatoren die Ergebnisse der zugehörigen kalkulatorischen Kennzahlen zu berechnen.

### 9.1 Ergebnis O-Kennzahl

<b>as_o_indicator_result</b>	Berechnet aus den Ergebnisinformationen eines O/E-Qualitätsindikators das Ergebnis der zugehörigen O-Kennzahl
Syntax	<code>as_o_indicator_result(<b>result</b>)</code>
Parameter	<b>result</b> – das Ergebnis eines O/E-Indikators
Rückgabewert	Das Ergebnis der zugehörigen O-Kennzahl (Zähler: O, Nenner: Anzahl Grundgesamtheit).
Beispiel	<b>ID O_52009 (DEK) – O-Kennzahl:</b> <pre>result &lt;- import_indicator(module = "DEK", id = "52009") as_o_indicator_result(result)</pre>

### 9.2 Ergebnis E-Kennzahl

<b>as_e_indicator_result</b>	Berechnet aus den Ergebnisinformationen eines O/E-Qualitätsindikators das Ergebnis der zugehörigen E-Kennzahl
Syntax	<code>as_e_indicator_result(<b>result</b>)</code>
Parameter	<b>result</b> – das Ergebnis eines O/E-Indikators
Rückgabewert	Das Ergebnis der zugehörigen E-Kennzahl (Zähler: E, Nenner: Anzahl Grundgesamtheit).
Beispiel	<b>ID E_52009 (DEK) – E-Kennzahl:</b> <pre>result &lt;- import_indicator(module = "DEK", id = "52009") as_e_indicator_result(result)</pre>

# Impressum

## HERAUSGEBER

---

IQTIG – Institut für Qualitätssicherung  
und Transparenz im Gesundheitswesen  
Katharina-Heinroth-Ufer 1  
10787 Berlin

Telefon: (030) 58 58 26-0

[info@iqtig.org](mailto:info@iqtig.org)

[iqtig.org](http://iqtig.org)