Vectorized Functions

to use with mutate()

mutate()) transmute()++1 /*-$
!0)/$)./**0()./*- /) 2*0().u
/*-$0)/$)./& /*-..$+0/)
- /0-)1 /*-.*!/#(')''/#*0/+0/u

O sets

lag() ä
lead() ä

Cumulative Aggregates

cumall()
cumany()
cummax()
cummean()
cummin()
cumprod()
cumsum()

Rankings

cume_dist()
dense_rank() ä
min_rank() ä
ntile()
percent_rank()
row_number() ä

Math

+v-v*v/v^v%/%v%% ä
log()vlog2()vlog10()
<v<=v>v>=v!=v==

Misc

between()
case_when()
coalesce()
'( )/-*.. /*1*.
if_else() ä( )/
na_if() ä'+
pmax() ä( )/
pmin() ä( )/
recode()ä*$
recode_factor() ä*$
!/*.

Summary Functions

to use with summarise()

summarise()++$.0((-0)/$)./*
0()./- /) 2/'u0((-4
!0)/$)./& /*-..$+0/)- /0-).$''
10 ..*0/+0/u

Counts

n()
n_distinct()
sum(!is.na())

Location

mean()
median()

Logicals

mean()
sum()

Position/Order

first()
last()
nth()

Rank

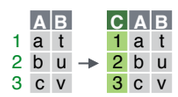
quantile()
min()
max()

Spread

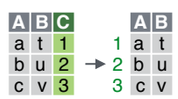
IQR()
mad()
sd()
var()

/- n*(-$

.)/0. -*2)(.v2*-
1-$0/.$!/#0().u*2*-2
/#*2)(.v!$/(*1 /(*0**0()u



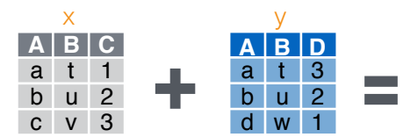
rownames_to_column()
1(.$/t
a <- rownames_to_column(iris,
var = "C")



column_to_rownames()
1*)(.u
column_to_rownames(a,
var = "C")

has_rownames()vremove_rownames()

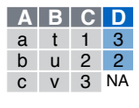
-01#/



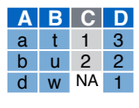
. bind_cols()/*+./ /'. .$#/#
./#4u

bind_cols(y)
.$.$''/u
u

. Mutating Join/*%*$*) /'/**0().
!-*()*/#v(/$*'10 .2$#*2./#
/#4-- .+*)/*u%*$> /$.$$)
$0(

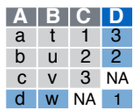


le_join(
)
$$'10!(

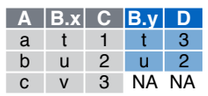


right_join(
v .
)
$$'10!(

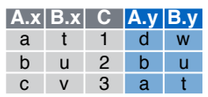
inner_join(
v .
)
$.$2.2$
(#u



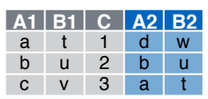
full_join(
)
$.$10v'
2.u



by = c("col1", "col2") /*
.+ $#0()d*(/#
*)u
left_join(x, y, by = "A")

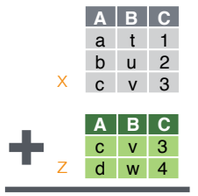


.)(1 /*-vby =
c("col1" = "col2")v/*(/#)
*0().2
#
left_join(x, y, by = c("C" = "D"))



. suffix/*.+ $40$$
left_join(
suffix = c("1", "2"))

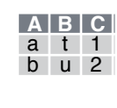
Combine Cases



. bind_rows()/*+./ /'. *2#/#.
/#4u

bind_rows(yvu$
..$''/u
)($#$$
'/)(.

intersect(y)

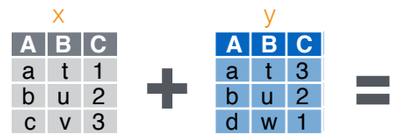


setdi(y)

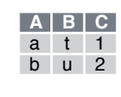
union(y)
-(1 union_all() - /$
$.u

. setequal()/*/ ./2##/2*/. /.*)/$
/#3(-*2.$)4- -

/-



. Filtering Join/*!$-*) /'"$//#
-*2.*!)*/#u



semi_join(y)
u

anti_join(y)
u