

dplyr yordamida Ma'lumotlar Transformatsiyasi bo'yicha qo'llanma



dplyr funksiyalari trubka(pipe)lar bilan ishlaydi va toza ma'lumotni qabul qiladi.



Har bir **o'zgaruvchi** tegishli **ustunda**

Har bir **kuzatuv**, yoki **holat**, tegishli **satrda**



x %>% f(y) ga teng **f(x, y)**

Jamlash

Quyidagilar **jamlash funksiyalarini** ustunlarga qo'llab, yangi jadval yaratadi. Jamlash funksiyalari vektor qabul qiladi va bir qiymatni qaytaradi. (orqa sahifaga qarang).



summarise(.data, ...)
Xulosalar jadvalni hisoblash.
Shuningdek, **summarise_()**.
summarise(mtcars, avg = mean(mpg))

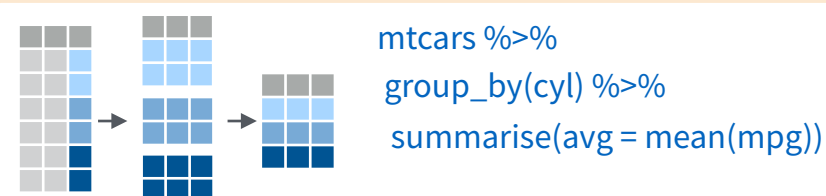
count(x, ..., wt = NULL, sort = FALSE)
Har bir guruhdagi satrlar sonini hisoblash.
Shuningdek, **tally()**.
count(iris, Species)

Turli shakllari

- **summarise_all()** - funksiyani har bir ustunga qo'llash.
- **summarise_at()** - funksiyani ma'lum bir ustunga qo'llash
- **summarise_if()** - funksiyani bir toifadagi ustunlarga qo'llash

Guruhlash

group_by() jadvalning guruhlangan nusxasini yaratishda qo'llaniladi. dplyr funksiyalari har bir guruh ustida alohida ishlab, so'ng natijalarni birlashtiradi.



group_by(.data, ..., add = FALSE)
... ga ko'ra guruhlangan jadval nusxasini qaytaradi.
g_iris <- group_by(iris, Species)

ungroup(x, ...)
Jadvalning guruhlanmagan nusxasini qaytaradi.
ungroup(g_iris)

Manipulyatsiyalash

Ajratish

Satr funksiyalari jadval ko'rinishidagi satrlar to'plamini qaytaradi. Nostandart kod uchun `_` bilan yakunlanuvchi variantidan foydalaning.

filter(.data, ...)
Shartni qanoatlantiruvchi satrlarni ajratish.
Shuningdek, **filter_()**.
filter(iris, Sepal.Length > 7)

distinct(.data, ..., .keep_all = FALSE)
Bir xil qiymatli satrlarni o'chirish.
Shuningdek, **distinct_()**.
distinct(iris, Species)

sample_frac(tbl, size = 1, replace = FALSE, weight = NULL, .env = parent.frame())
Berilgan nisbatda satrlarni tasodifiy ajratib olish.
sample_frac(iris, 0.5, replace = TRUE)

sample_n(tbl, size, replace = FALSE, weight = NULL, .env = parent.frame())
Satr o'lchamini tasodifiy tanlash.
sample_n(iris, 10, replace = TRUE)

slice(.data, ...)
Joylashuviga ko'ra satrni ajratib olish.
Shuningdek, **slice_()**. *slice(iris, 10:15)*

top_n(x, n, wt)
Top n ta yozuvni tanlash va tartiblash (agar guruhlangan bo'lsa, guruh bo'yicha).
top_n(iris, 5, Sepal.Width)

filter() bilan ishlatish mumkin bo'lgan mantiqiy operatorlar

<	<=	is.na()	%in%		xor()
>	>=	!is.na()	!	&	

Yordam uchun **?base::logic** va **?Comparison** ga qarang.

Tartibga keltirish

arrange(.data, ...)
Ustundagi qiymatlarga ko'ra satrlarni tartiblash (o'sish tartibida), kamayish tartibida esa **desc()** yordamida tartiblash mumkin.
arrange(mtcars, mpg)
arrange(mtcars, desc(mpg))

Qo'shish

add_row(.data, ..., .before = NULL, .after = NULL)
Jadvalga bir yoki bir nechta satrni qo'shish.
add_row(faithful, eruptions = 1, waiting = 1)

O'zgaruvchilar manipulyatsiyasi

O'zgaruvchilarni ajratish

Ustun funksiyalari ustunlar to'plamini yangi jadval shaklida taqdim qiladi. Nostandart kod uchun `_` bilan tugaydiganlaridan foydalaning.

select(.data, ...)
Nomiga ko'ra ustunni ajratib olish.
Shuningdek, **select_if()**.
select(iris, Sepal.Length, Species)

select() bilan quyidagi yordamchi funksiyalardan foydalaning,
Masalan: *select(iris, starts_with("Sepal"))*

contains(match)	num_range(prefix, range)	•, e.g. mpg:cyl
ends_with(match)	one_of(...)	-, e.g. -Species
matches(match)	starts_with(match)	everything()

Yangi o'zgaruvchilarni yaratish

Quyidagilar, **vektorli funksiyalarni** ustunlarga tatbiq qiladi. Vektorli funksiyalar vektorni qabul qiladi va bir xil o'lchamdagi vektorni qaytaradi.



mutate(.data, ...)
Yangi ustun(lar)ni hisoblash.
mutate(mtcars, gpm = 1/mpg)

transmute(.data, ...)
Yangi ustun(lar)ni hisoblash, qolganlarini tashlab yuborish.
transmute(mtcars, gpm = 1/mpg)

mutate_all(.tbl, .funs, ...)
Funksiyalarni har bir ustunga tatbiq qilish.
funs() bilan ishlatish lozim.
mutate_all(faithful, funs(log(.), log2(.)))

mutate_at(.tbl, .cols, .funs, ...)
Funksiyalarni faqat ayrim ustunlarga tatbiq qilish. **funs()**, **vars()** va **select()** uchun yordamchi funksiyalardan foydalanish lozim.
mutate_at(iris, vars(-Species), funs(log(.)))

mutate_if(.tbl, .predicate, .funs, ...)
Funksiyalarni bir xil toifadagi ustunlarga qo'llash. **funs()** bilan qo'llang.
mutate_if(iris, is.numeric, funs(log(.)))

add_column(.data, ..., .before = NULL, .after = NULL)
Yangi ustun(lar)ni qo'shish.
add_column(mtcars, new = 1:32)

rename(.data, ...)
Ustunlarni qayta nomlash.
rename(iris, Length = Sepal.Length)

Vektorli funksiyalar

mutate() bilan qo'llash uchun

mutate() va **transmute()** vektorli funksiyalarni ustunlarqa qo'llab, yangi ustun hosil qiladi. Vektorli funksiyalar vektorlarni qabul qilib, shu o'lchamli vektorni qaytaradi.



Siljitish

dplyr::lag() - Elementlarni 1 ta oldinga siljitish.
dplyr::lead() - Elementlarni 1 ta orqaga siljitish.

Yig'uvchi funksiyalar

dplyr::cumall() - yig'uvchi all()
dplyr::cumany() - yig'uvchi any()
cummax() - yig'uvchi max()
dplyr::cummean() - yig'uvchi mean()
cummin() - yig'uvchi min()
cumprod() - yig'uvchi prod()
cumsum() - yig'uvchi sum()

Reytinglash

dplyr::cume_dist() - barcha qiymatlar nisbati <=
dplyr::dense_rank() - ties = min ga ko'ra reytinglash, bo'sh joylarsiz.
dplyr::min_rank() - ties = min ga ko'ra reytinglash
dplyr::ntile() - bins into n bins
dplyr::percent_rank() - [0,1] oralig'iga keltirilgan min_rank
dplyr::row_number() - ties = "first" ga ko'ra reytinglash

Matematika

+, **-**, *****, **/**, **^**, **%/%**, **%%** - arifmetik amallar
log(), **log2()**, **log10()** - logarifmlar
<, **<=**, **>**, **>=**, **!=**, **==** - mantiqiy taqqoslash

Boshqalar

dplyr::between() - $x \geq \text{left} \ \& \ x \leq \text{right}$
dplyr::case_when() - ko'p holatli if_else()
dplyr::coalesce() - vektorlar to'plami elementlarining, birinchi mavjud qiymatlari
dplyr::if_else() - elementma-element if() + else()
dplyr::na_if() - ma'lum qiymatlarni NAga almashtirish
pmax() - elementma-element max()
pmin() - elementma-element min()
dplyr::recode() - vektorli switch()
dplyr::recode_factor() - faktorlar uchun vektorli switch()

Jamlash funksiyalari

summarise() bilan qo'llash uchun

summarise() jamlash funksiyalarini ustunlarqa qo'llab yangi jadval yaratadi. Jamlash funksiyalari, vektorlarni qabul qilib, yagona qiymat qaytaradi.



Soni

dplyr::n() - qiymatlar/satrlar soni
dplyr::n_distinct() - unikal satrlar soni
sum(!is.na()) - qiymati mavjud satrlar soni

Joylashuvi

mean() - o'rtacha qiymat, **mean(!is.na())**
median() - mediana

Mantiqiy qiymatlar

mean() - ROST qiymatlar nisbati
sum() - ROST (TRUE) qiymatlari soni

Joylashuv/tartib

dplyr::first() - birinchi qiymat
dplyr::last() - oxirgi qiymat
dplyr::nth() - vektorning n-qiymati

Rang

quantile() - n-kvartil
min() - minimum qiymat
max() - maksimum qiymat

Yoyilishi

IQR() - Inter-kvartil oralig'i
mad() - chetlanishning o'rtacha absolyut qiymati
sd() - o'rtacha kvadratik chetlanish
var() - dispersiya

Satr nomlari

Toza ma'lumotlar ustunlardan alohida saqlanuvchi satr nomlaridan foydalanmaydi. Ular bilan ishlash uchun avval ustunga ko'chirish lozim.

rownames_to_column()

A	B	C
1	a	t
2	b	u
3	c	v

 →

A	B	C
1	a	t
2	b	u
3	c	v

Satr nomlarini ustunga ko'chirish.
`a <- rownames_to_column(iris, var = "C")`

column_to_rownames()

A	B	C
a	t	1
b	u	2
c	v	3

 →

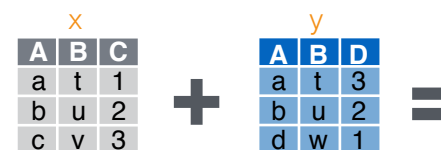
A	B
1	a
2	b
3	c

Ustunni satr nomlariga ko'chirish.
`column_to_rownames(a, var = "C")`

Hamda **has_rownames()**, **remove_rownames()**

Jadvallarni birlashtirish

O'zgaruvchilarni birlashtirish



bind_cols() dan jadvallarni qanday bo'lsa shundayligicha bir-biriga yonma-yon qo'yishda foydalaning.

bind_cols(...)
Jadvallarni yonma-yon joylashtirib, yagona jadval shaklida qaytaradi.
SATRLAR MOS KELISHI LOZIM.

"O'zgartirib birlashtirish" funksiyalaridan bir jadvalga boshqasining ustunlaridan foydalanib mos satrlarga birlashtirishda foydalanish mumkin. Har bir funksiya jadvaldan turli qiymatlar jadvalini saqlab qoladi.

left_join(x, y, by = NULL, copy=FALSE, suffix=c("x","y"),...)
Mos qiymatlarni y dan x ga joylash.

right_join(x, y, by = NULL, copy = FALSE, suffix=c("x","y"),...)
Mos qiymatlarni x dan y ga joylash.

inner_join(x, y, by = NULL, copy = FALSE, suffix=c("x","y"),...)
Ma'lumotlarni birlashtirish. Faqat mos keluvchi satrlar saqlanib qolinadi.

full_join(x, y, by = NULL, copy=FALSE, suffix=c("x","y"),...)
Ma'lumotlarni birlashtirish. Barcha qiymatlar va satrlar saqlanib qolinadi.

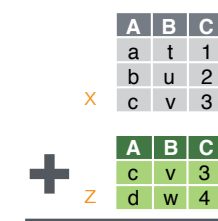
by = c("col1", "col2") dan mos keltirish uchun ustun(lar)ni ko'rsatishda foydalaning.

left_join(x, y, by = "A")

by = c("col1" = "col2") dan turli nomdagi ustunlarni mos keltirishda foydalaning.
left_join(x, y, by = c("C" = "D"))

suffix dan bir xil ustun nomlari uchun nom berishda foydalaning.
left_join(x, y, by = c("C" = "D"), suffix = c("1", "2"))

Birlashtirish



bind_rows() dan jadvallarni qanday bo'lsa shundayligicha bir-birining ostiga joylashtirishda foydalaning.

bind_rows(..., .id = NULL)
Yagona jadval sifatida jadvallarni birining ostiga joylashtirib qaytaradi.

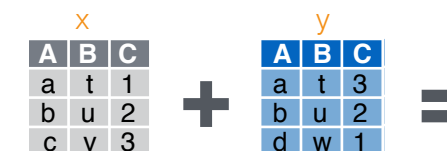
intersect(x, y, ...)
x va z da mavjud satrlarni qaytaradi.

setdiff(x, y, ...)
Faqat x da mavjud satrlar

union(x, y, ...)
x yoki z da mavjud satrlar. (Bir xil satrlar o'chiriladi). **union_all()** bir xil satrlar saqlanib qolinadi.

Jadvallar bir xil satrlarga ega yoki ega emasligi tekshiriladi (tartibidan qat'iy nazar).

Satrlarni saralab olish



"Saralab birlashtirish" funksiyalaridan bir jadvalni, 2-sining satrlariga ko'ra saralab olishda foydalanish mumkin.

semi_join(x, y, by = NULL, ...)
x ning y bilan mos keluvchi satrlarini qaytaradi. NIMALAR BIRLASHTIRILISHINI KO'RISH MAQSADGA MUVOFIQ.

anti_join(x, y, by = NULL, ...)
x ning y bilan mos kelmaydigan satrlarini qaytaradi.
NIMALAR BIRLASHTIRILMASLIGINI KO'RISH MAQSADGA MUVOFIQ.