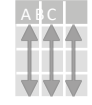


dplyr ile Veri Düzenleme : Referans Kağıdı



dplyr, "pipe" zincir operatörünü kullanır. dplyr fonksiyonları zincir operatörleriyle çalışır ve düzenli veri ister.



Her değişken kendi sütunundadır.

&



Her gözlem veya durum kendi satırındadır.



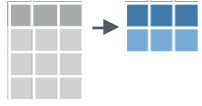
Zincir (pipes)

$x \%>\% f(y)$ ifadesi $f(x, y)$ demektir.

Durumları Özetle

Bu fonksiyonlar, bir özet istatistik tablosu oluşturmak için özet fonksiyonları kullanır. Özet fonksiyonları, vektörleri girdi olarak alır ve sonucu bir değerdir (diğer sayfalara bakınız).

özet fonksiyon



`summarise(.data, ...)` Özet bir tablo oluşturur. Örneğin: `summarise(mtcars, avg = mean(mpg))`



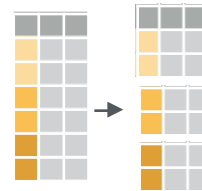
`count(x, ..., wt=NULL, sort=FALSE)` Değişkenler tarafından oluşturulan her gruptaki satır sayısını verir. Bakınız: `tally()`. Örneğin: `count(iris, Species)`

VARYASYONLAR

`summarise_all()` - Fonksiyonları tüm sütuna uygular.
`summarise_at()` - Fonksiyonları belirtilen sütuna uygular.
`summarise_if()` - Aynı tipteki bütün sütunlara bu fonksiyonları uygular.

Durumları Gruplama

Bir tablonun "gruplanmış" bir kopyasını oluşturmak için `group_by()` fonksiyonu kullanılır. dplyr fonksiyonları her "grubu" aynı işleri ve ardından sonuçları birleştirir. Örneğin:



`mtcars %>%`
`group_by(cyl) %>%`
`summarise(avg = mean(mpg))`

`group_by(.data, ..., add=FALSE)`
Aynı tablonun belirlenen değişkenlerle kopyasını oluşturur. Örneğin: `g_iris <- group_by(iris, Species)`

`ungroup(x, ...)`
Tabloyu gruptan çıkarır.
Örneğin: `ungroup(g_iris)`

Durumları Düzenleme

DURUMLARI SEÇME

Satır fonksiyonları, yeni bir tablo ortaya çıkarır:



`filter(.data, ...)` Mantıksal ölçütlere uyan satırları getirir. Örneğin: `filter(iris, Sepal.Length > 7)`



`distinct(.data, ..., .keep_all = FALSE)` Yinelenen değerlere sahip satırları teke indirip, tek satır olarak getirir. Örneğin: `distinct(iris, Species)`
`sample_frac(tbl, size = 1, replace = FALSE, weight = NULL, .env = parent.frame())`



Satırların belirli bir yüzdesini seçip, getirir. Örneğin: `sample_frac(iris, 0.5, replace = TRUE)`
`sample_n(tbl, size, replace = FALSE, weight = NULL, .env = parent.frame())` Belirlenen sayıda rastgele satır seçip getirir. Örneğin: `sample_n(iris, 10, replace = TRUE)`



`slice(.data, ...)` Sıra numarasına göre ilgili satır seçip, getirir. Örneğin: `slice(iris, 10:15)`
`top_n(x, n, wt)` En üst n satır (gruplandırılmış veri ise grubuna göre) seçip, sıralar. Örneğin: `top_n(iris, 5, Sepal.Width)`

`filter()` ile kullanılacak mantıksal ve matematiksel operatörler

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

Konsola `?base::Logic` ve `? yazılarak yardım sayfaları açılabilir.`

DURUMLARI SIRALA



`arrange(.data, ...)` Satırlar bir sütun veya sütunların değerlerine göre (büyükten küçüğe) sıralanır, büyükten küçüğe doğru sıralamak için `desc()` kullanılır. Örneğin: `arrange(mtcars, mpg)`
`arrange(mtcars, desc(mpg))`

DURUM EKLEME



`add_row(.data, ..., .before=NULL, .after=NULL)` Tabloya bir veya daha fazla satır ekler. Örneğin: `add_row(faithful, eruptions = 1, waiting = 1)`

Değişkenleri Düzenleme

DEĞİŞKEN(SÜTUN) SEÇME

Sütun fonksiyonları, sütunların bir alt kümesini seçip, yeni bir tablo oluşturur.



`pull(.data, var=1)` Sütun değerlerine göre seçim yapar. Sütun ismine veya dizine göre seçim yapılır. Örneğin: `pull(iris, Sepal.Length)`



`select(.data, ...)` Sütun ismine göre seçip, tabloyu yapar. Bakınız: `select_if()`. Örneğin: `select(iris, Sepal.Length, Species)`

`select()` için aşağıdaki yardımcıları kullanabilirsiniz. Örneğin: `select(iris, starts_with("Sepal"))`

<code>contains (match)</code>	<code>num_range (prefix, range)</code>	;e.g. <code>mpg:cyl</code>
<code>ends_with (match)</code>	<code>one_of(...)</code>	;e.g. <code>Tür eşleşmeleri</code>
<code>matches (match)</code>	<code>start_with (match)</code>	

YENİ DEĞİŞKENLER OLUŞTURMA

Bunlar, sütunlara vektörleştirilmiş fonksiyonlar uygular. Vektörleştirilmiş fonksiyonlar, girdi olarak vektörleri alır ve çıktı olarak aynı uzunlukta vektörler getirir (diğer sayfalara bakınız).

vektörleştirilmiş fonksiyon



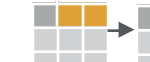
`mutate(.data, ...)` Yeni sütun(lar) hesaplanır. Örneğin: `mutate(mtcars, gpm = 1/mpg)`



`transmute(.data, ...)` Yeni sütun(lar) hesaplanır, diğerlerini bırakır. Örneğin: `transmute(mtcars, gpm = 1/mpg)`



`mutate_all(.tbl, .funs, ...)` Fonksiyonlar tüm sütunlara uygulanır. Bakınız: `mutate_if()`. Örneğin: `mutate_all(faithful, funs(log(.), log2(.)))` `mutate_if(iris, is.numeric, funs(log(.)))`



`mutate_at(.tbl, .cols, .funs, ...)` Belirli sütunlara fonksiyonlar uygulanır. `funs()`, `var()` ve `select()` için yardımcı fonksiyonlar kullanılır. Örneğin: `mutate_at(iris, vars(-Species), funs(log(.)))`



`add_column(.data, ..., .before=NULL, .after=NULL)` Yeni sütun(lar) eklenir. Bakınız: `add_count()`, `add_tally()`. Örneğin: `add_column(mtcars, new = 1:32)`



`rename(.data, ...)` Sütunlar yeniden adlandırılır. Örneğin: `rename(iris, Length = Sepal.Length)`



Vektör Fonksiyonları

MUTATE() İLE KULLANIM

Mutate() ve transmute(), yeni sütunlar oluşturmak için sütunlara vektörleştirilmiş fonksiyonlar uygular. Vektörleştirilmiş fonksiyonlar vektörleri girdi olarak alır ve aynı uzunluktaki vektörleri çıktı olarak getirir.

vektörleştirilmiş fonksiyon

KAYDIRMA

dplyr::lag(): Değerleri 1 satır kaydırır.
dplyr::lead(): Değerleri -1 satır kaydırır.

KÜMÜLATİF TOPLAMLAR

dplyr::cumall() - Tümü kümülatif
dplyr::cumany() - Herhangi biri kümülatif
cummax() - Kümülatif maksimum
dplyr::cummean() - Kümülatif ortalama
cummin() - Kümülatif minimum
cumprod() - Kümülatif çarpım
cumsum() - Kümülatif toplam

SIRALAMALAR

dplyr::cume_dist() - Tüm değerlerin oranı <= dplyr::dense_rank() - sıralama, beraberlikte = min, boşluksuz dplyr::min_rank() - beraberlikte = min dplyr::ntile() - n'e kadar grubu gruplar. dplyr::percent_rank() - min_siralama [0,1] olarak ölçeklendirilir.
dplyr::row_number() - sıralama, beraberlikte= "ilk"

MATEMATİKSEL İFADELER

+, -, *, /, ^, %/%, %% - aritmetik işlemler
log(), log2(), log10() - logaritmik
<, <=, >, >=, !=, == - mantıksal karşılaştırmalar

dplyr::between() - x >= sol & x <= sağ dplyr::near() - safe==gerçek sayılar

DiĞER

dplyr::case_when() - çoklu if_else()
iris %>% mutate(Species = case_when(
Species == "versicolor" ~ "versi",
Species == "virginica" ~ "virgi",
TRUE ~

Species)) dplyr::coalesce() - bir vektör kümesinde ilk NA olmayan değerleri getirir.
dplyr::if_else() - eleman-eleman if() + else()
dplyr::na_if() - belirli değerleri NA değeri ile değiştirir.
pmax() - eleman-eleman maksimum()
pmin() - eleman-eleman minimum()
dplyr::recode() - Vektörleştirilmiş switch()
dplyr::recode_factor() - Vektörleştirilmiş switch() faktörler için.

Özet Fonksiyonları

SUMMARISE() İLE KULLANMAK İÇİN

summarise() yeni bir tablo oluşturmak için sütunlara özet fonksiyonlar uygulanır. Özet fonksiyonlar, vektörleri girdi olarak alır ve çıktı olarak tek değer getirir.

özet fonksiyon

SAYMA

dplyr::n() -değer/satır sayısı
dplyr::n_distinct() - tekil sayısı
sum(lis.na()) - NA olmayan sayısı

KONUM

mean() -ortalama
aynca mean(lis.na())
median() - medyan

MANTIKSAL İŞLEMLER

mean() - Doğruların oranı
sum() - Doğru sayısı

POZİSYON/SIRALAMA

dplyr::first() -
ilk değer dplyr::last() - son değer
dplyr::nth() - Vektörün n'inci konumdaki değeri

SIRA

quantile() - n'inci kuantil min() - minimum değer max() - maksimum değer

DAĞILIM

IQR() -çeyrekler arası açıklık
mad() - medyan mutlak sapma
sd() - standart sapma
var() -varyans

Satır Adları

Düzenli veriler, sütunların dışında satır adı kullanmaz. Satır adlarıyla çalışmak için önce bunlar bir sütuna taşınır.

rownames_to_column()
Satır isimleri sütuna taşınır.
a <- rownames_to_column(iris, var = "C")

column_to_rownames()
Sütun isimleri satıra taşınır.
column_to_rownames(a, var = "C")

Ayrıca has_rownames(), remove_rownames()

Tabloları Birleştirme

DEĞİŞKENLERİ (SÜTUNLARI) BİRLEŞTİRME

x + y =

bind_cols() yan yana olan tabloları yapıştırmak için kullanılır.

bind_cols(...) yan yana yerleştirilmiş tabloları tek bir tablo olarak getirir.
SATIR SAYISININ AYNI OLDUĞUNDAN EMİN OLUN.

Bir tabloyu, diğer tablodaki sütunlara birleştirmek için "Mutasyon Birleştirme" kullanılır, değerlere karşılık gelen satırlarla eşleştirilir. Her birleştirme, tablolardan farklı değerlerin bir kombinasyonunu gösterir.

left_join(x, y, by=NULL, copy=FALSE, suffix=c("x", "y"),...) y' deki eşleşen değerler x'e birleştirir.

right_join(x, y, by=NULL, copy=FALSE, suffix=c("x", "y"),...) x' deki eşleşen değerleri y ile birleştirir.

inner_join(x, y, by=NULL, copy=FALSE, suffix=c("x", "y"),...) Verilere birleştirir. Sadece uyuşan satırları getirir.

full_join(x, y, by=NULL, copy=FALSE, suffix=c("x", "y"),...) Verileri birleştirir. Tüm değerleri, tüm satırları getirir.

by = c("col1", "col2", ...) Bir veya daha fazla ortak sütun belirtmek için kullanılır.
left_join(x, y, by = "A")

İsimlendirilmiş bir vektör kullanılır, by = c("col1" = "col2"), her tabloda farklı isimlere sahip sütunlar tek bir sütun olarak birleştirilmek için kullanılır. left_join(x, y, by = c("C" = "D"))

Her iki tabloda da aynı isimli sütunlara verilecek değerleri belirtmek için vektör kullanılır. left_join(x, y, by = c("C" = "D"), suffix = c("1", "2"))

DURUMLARI BİRLEŞTİRME

x + y =

Bind_row(), alt taraftaki tabloyu olduğu gibi yapıştırmak için kullanılır.

bind_rows(..., .id=NULL)
Tabloları tek bir tablo olarak diğerinin üstüne getirir. Orjinal tablo adlarını bir sütuna eklemek için .id bir sütun ismi olarak atanır.

intersect(x, y, ...) Hem x hem de y tablosunda görünen satırları alır.

setdiff(x, y, ...) x'de görünen ancak y tablosundaki olmayan satırları alır.

union(x, y, ...) x veya y'de görünen satırları (tekrarlanan satırları hariç). union_all() tekrarları tutar.

İki veri kümesinin aynı satırları (herhangi bir sırada) içerip içermediğini test etmek için setequal() kullanılır.

SATIRLARI AYIKLA

x + y =

Bir tabloyu diğerinin satırlarına göre filtrelemek için "Ortak Filtreleme" kullanılır.

semi_join(x, y, by=NULL, ...) y ile uyuşan x satırlarını getirir. NEYİN BİRLEŞTİRİLECEĞİNİ GÖRMEK İÇİN KULLANIŞLIDIR.

anti_join(x, y, by=NULL, ...) y ile uyuşmayan x satırlarını getirir. NEYİN BİRLEŞTİRİLMEMEYECİĞİNİ GÖRMEK İÇİN KULLANIŞLIDIR.