R

Лекция 6

Advanced dplyr

Задача - на каждой планете из датасета starwars выбрать представителя с наибольшим ростом

top_n

Выбрать из ВСЕЙ таблицы строку с наибольшим height

```
starwars %>% top_n(1, height)
```

```
## # A tibble: 1 x 13
##   name  height  mass hair_color skin_color
##   <chr>  <int> <dbl> <chr>      <chr>
## 1 Yara~    264    NA none       white
## # ... with 5 more variables: homeworld <chr
## #   vehicles <list>, starships <list>
```

top_n

Выбрать из ВСЕЙ таблицы строку с наименьшим height

```
starwars %>% top_n(-1, height)
```

```
## # A tibble: 1 x 13
##   name  height  mass hair_color skin_color e
##   <chr>  <int> <dbl> <chr>      <chr>       <
## 1 Yoda      66    17 white      green       l
## # ... with 5 more variables: homeworld <chr>
## #   vehicles <list>, starships <list>
```

group_by(variable)

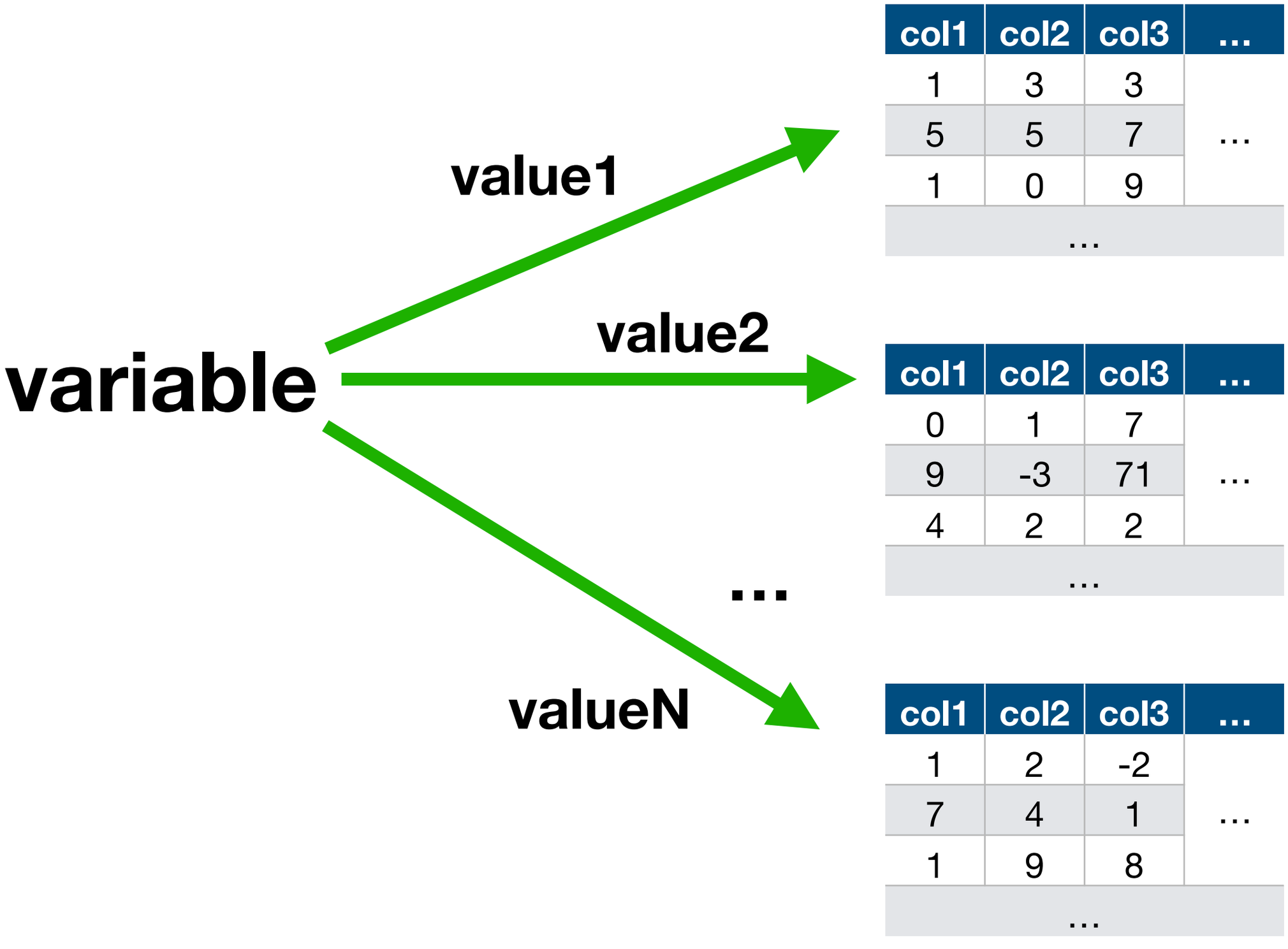
Разбивает таблицу на подтаблицы по значениям variable

```
starwars %>% group_by(homeworld)
```

```
## # A tibble: 87 x 13
## # Groups:   homeworld [49]
##    name  height  mass hair_color skin_color
##    <chr>  <int> <dbl> <chr>      <chr>
##  1 Luke~    172    77 blond      fair
##  2 C-3PO    167    75 <NA>       gold
##  3 R2-D2     96    32 <NA>       white, bl~
```

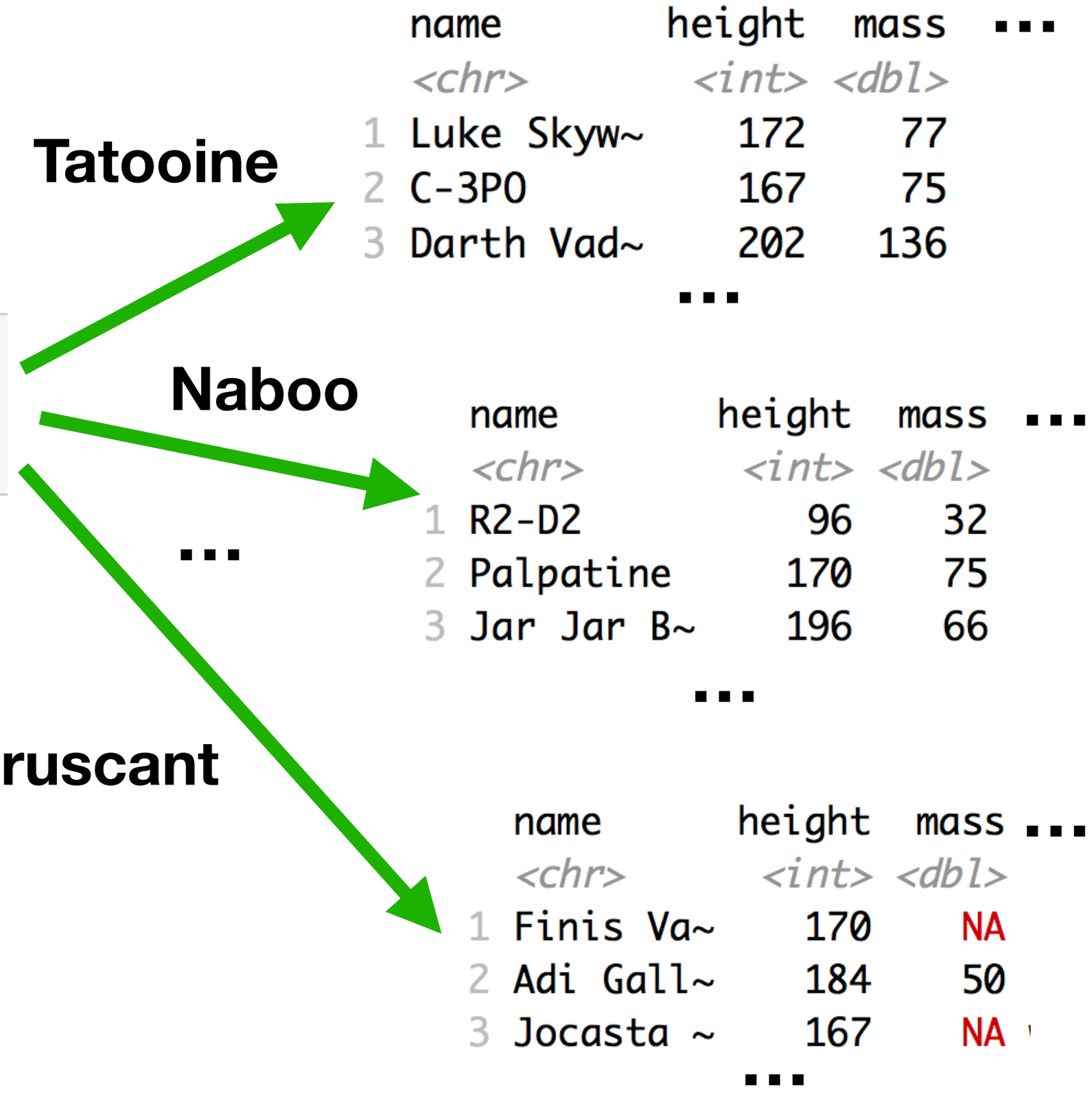
Специальный объект - tibble

Как выглядит groupby-объект



Как выглядит groupby-объект

```
starwars %>%
    group_by(homeworld)
```

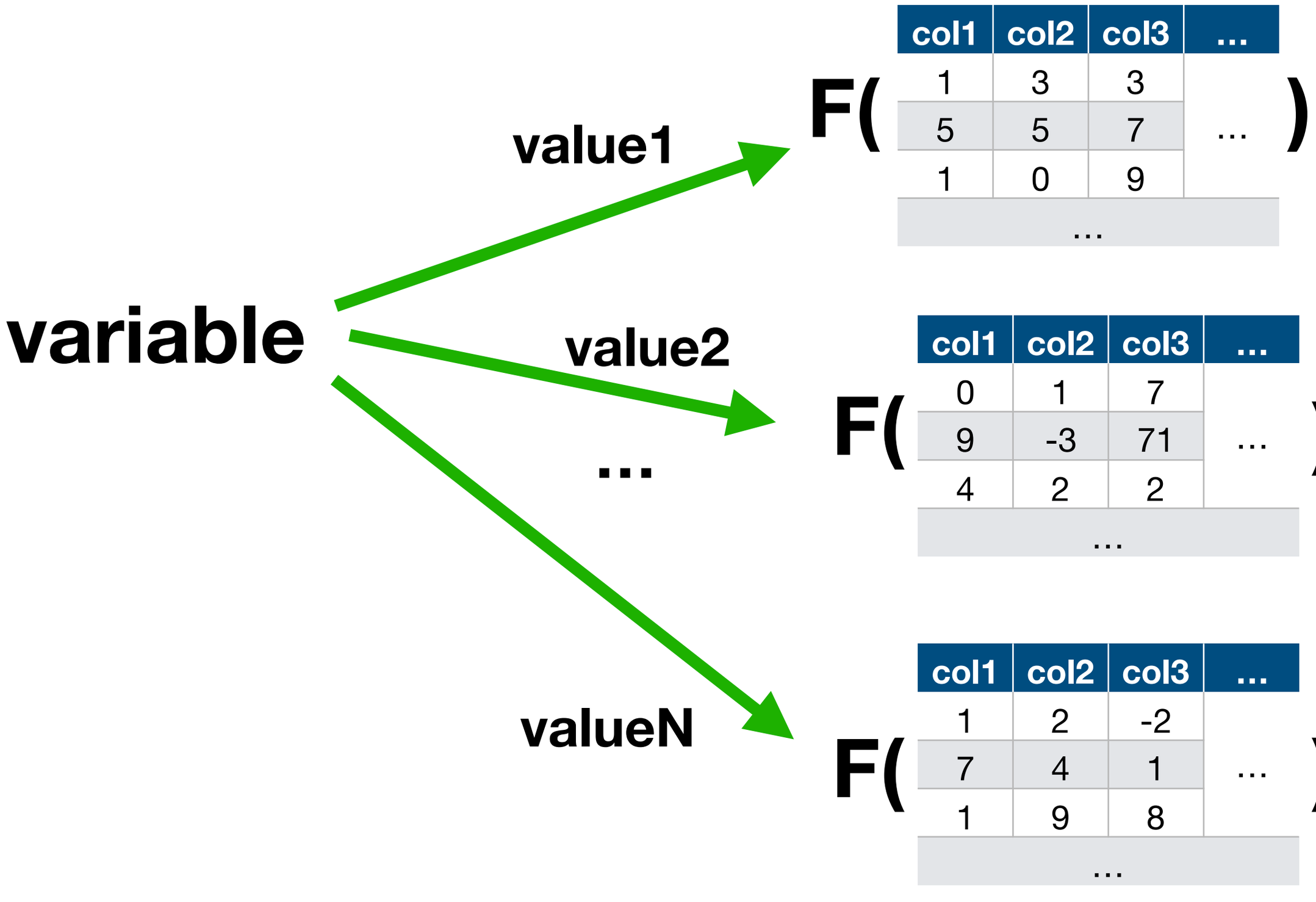


Решение нашей задачи?

```
starwars %>%
    group_by(homeworld) %>%
    top_n(1, height)
```

```
## # A tibble: 49 x 13
## # Groups:   homeworld [49]
##    name  height  mass hair_color skin_color
##    <chr>  <int> <dbl> <chr>      <chr>
##  1 Dart~    202   136 none       white
##  2 Obi-~    182    77 auburn, w~ fair
##  3 Wilh~    180    NA auburn, g~ fair
```

Как выполняется функция на groupby-объекте



Результат выполнения функции top_n(1) на starwars

```
starwars %>%
    group_by(homeworld) %>%
    top_n(1, height)
```

Tatooine

name	height	mass
<chr>	<int>	<dbl>
1 Darth Va~	202	136

Naboo

name	height	mass
<chr>	<int>	<dbl>
1 Roos Tarp~	224	82

...

Coruscant

name	height	mass
<chr>	<int>	<dbl>
1 Adi Gal~	184	50

Решение нашей задачи?

```
starwars %>%
    group_by(homeworld) %>%
    top_n(1, height)
```

```
## # A tibble: 49 x 13
## # Groups:   homeworld [49]
##    name  height  mass hair_color skin_color
##    <chr>  <int> <dbl> <chr>      <chr>
##  1 Dart~    202   136 none       white
##  2 Obi-~    182    77 auburn, w~ fair
##  3 Wilh~    180    NA auburn, g~ fair
```

Решение нашей задачи?

```
starwars %>%
    group_by(homeworld) %>%
    top_n(1, height)
```

```
## # A tibble: 49 x 13
## # Groups:   homeworld [49]
##      name  height  mass hair_color skin_color
##     <chr>  <int> <dbl> <chr>      <chr>
##   1 Dart~    202   136 none       white
##   2 Obi-~    182    77 auburn, w~ fair
##   3 Wilh~    180    NA auburn, g~ fair
```

Все еще не избавились от группировки

ungroup()

Снять разбиение на подгруппы

```
starwars %>%
    group_by(homeworld) %>% ungroup()
```

```
## # A tibble: 87 x 13
##    name  height  mass hair_color skin_color
##    <chr>  <int> <dbl> <chr>      <chr>
##  1 Luke~    172    77 blond      fair
##  2 C-3PO    167    75 <NA>       gold
##  3 R2-D2     96    32 <NA>       white, bl~
```

Решение нашей задачи

```
starwars %>%
    group_by(homeworld) %>%
    top_n(1, height) %>%
    ungroup()
```

```
## # A tibble: 49 x 13
##    name  height  mass hair_color skin_color
##    <chr>  <int> <dbl> <chr>      <chr>
##  1 Dart~    202   136 none       white
##  2 Obi-~    182    77 auburn, w~ fair
##  3 Wilh~    180    NA auburn, g~ fair
##  4 Han ~    180    80 brown      fair
##  5 Gree~    173    74 <NA>       green
```

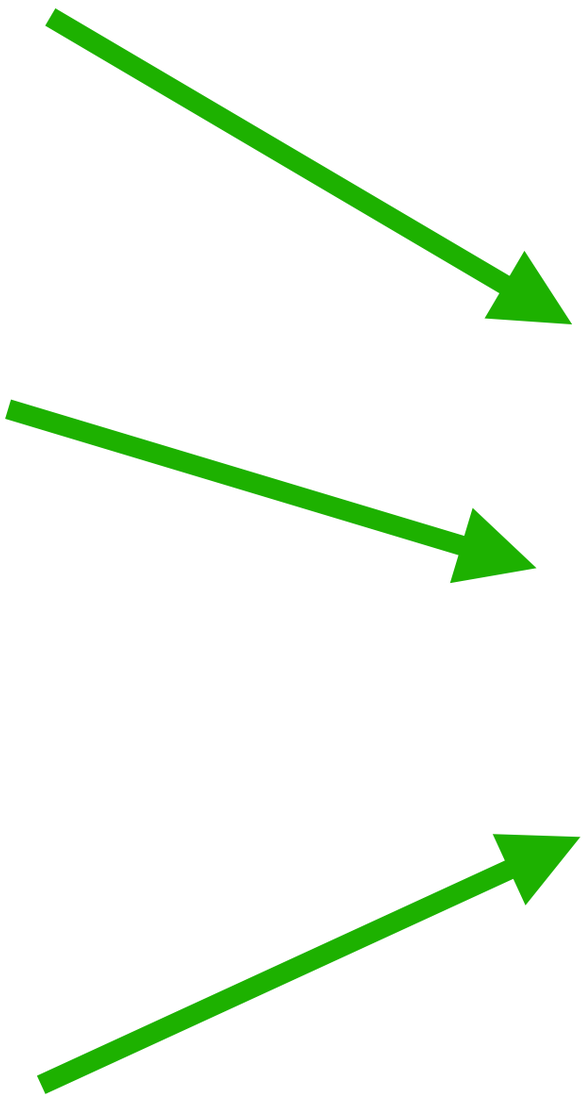
Как выполняется ungroup на groupby-объекте

col1	col2	col3	…
1	3	3	
5	5	7	…
1	0	9	
…			

…

col1	col2	col3	…
0	1	7	
9	-3	71	…
4	2	2	
…			

col1	col2	col3	…
1	2	-2	
7	4	1	…
1	9	8	
…			



col1	col2	col3	…
1	3	3	
5	5	7	
1	0	9	
2	3	3	
0	1	7	
9	-3	1	
4	2	2	…
0	4	3	
1	2	-2	
7	4	1	
1	9	8	
3	5	1	

Результат выполнения функции ungroup

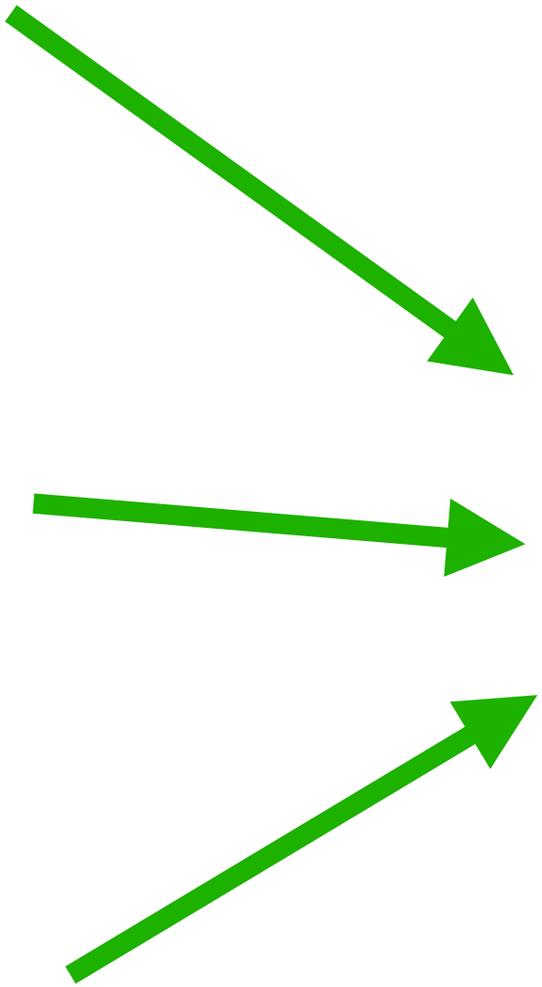
	name	height	mass
	<chr>	<int>	<dbl>
1	Darth Va~	202	136

...

	name	height	mass
	<chr>	<int>	<dbl>
1	Roos Tarp~	224	82

...

	name	height	mass
	<chr>	<int>	<dbl>
1	Adi Gal~	184	50



	name	height	mass
	<chr>	<int>	<dbl>
1	Darth V~	202	136
2	Obi-Wan~	182	77
3	Wilhuff~	180	NA
4	Han Solo	180	80

...

Задача - на каждой планете из датасета starwars выбрать 2 представителей с наибольшим ростом и посчитать средний для этих двух представителей

```
starwars %>%
    group_by(homeworld) %>%
    top_n(2, height) %>%
    summarise(mean=mean(height))
```

```
## # A tibble: 49 x 2
##    homeworld       mean
##    <chr>          <dbl>
##  1 Alderaan        190.
##  2 Aleen Minor      79
##  3 Bespin          175
##  4 Bestine IV      180
##  5 Cato Neimoidia  191
```

Summarise снимает одну группировку с таблицы

Почему summarise снимает группу

В общем случае у нас нет гарантии, что функция, примененная к подтаблице в группе, независимо от аргументов функции, вернет одну строку

variable

value1

value2

...

valueN

F(

col1	col2	col3	...
1	3	3	
5	5	7	...
1	0	9	
...			

)

F(

col1	col2	col3	...
0	1	7	
9	-3	71	...
4	2	2	
...			

)

В случае summarise такая гарантия есть.

F(

col1	col2	col3	...
1	2	-2	
7	4	1	...
1	9	8	
...			

)

Почему summarise снимает группу

summarise от таблицы/подтаблицы обязан возвращать строку.

```
starwars %>%
    summarise(mean=mean(height, na.rm=T))
```

```
## # A tibble: 1 x 1
##    mean
##   <dbl>
## 1  174.
```

В ней не обязательно один столбец, но это нам и не важно. Важно, что каждой группе соответствует только одна строка, потому группировку можно убрать

```
starwars %>%
    summarise(mean=mean(height, na.rm=T),
              sd=sd(height, na.rm=T))
```

```
## # A tibble: 1 x 2
##    mean    sd
##   <dbl> <dbl>
## 1  174.  34.8
```

Что мы не учли в решении задачи?

- А если есть несколько представителей с одинаковым и при этом минимальным ростом?
- А если нет даже 2 представителей данной планеты?

Что мы не учли в решении задачи?

- А если есть несколько представителей с одинаковым и при этом минимальным ростом? (документация top_n говорит, что в этом случае вернуться все такие представители)
- А если нет даже 2 представителей данной планеты? (документация top_n про это ничего не говорит, но можно проверить на игрушечном датасете, что вернется столько, сколько есть представителей вообще)

Важно курить манул

Задача - на каждой планете из датасета starwars найти два наименьших различных значения роста и посчитать средний рост для этих двух представителей

distinct

```
df <- data.frame(x = as.integer(c(10, 4, 1, 1, 1)),
                 y = as.integer(c(5, 5, 4, 1, 1)))
print(df)
```

```
##    x y
## 1 10 5
## 2  4 5
## 3  1 4
## 4  1 1
## 5  1 1
```

```
df %>% distinct()
```

Убрать повторяющиеся строки

```
##    x y
## 1 10 5
## 2  4 5
## 3  1 4
## 4  1 1
```

distinct

```
print(df)
```

```
##    x y
## 1 10 5
## 2  4 5
## 3  1 4
## 4  1 1
## 5  1 1
```

```
df %>% distinct(x)
```

```
##    x
## 1 10
## 2  4
## 3  1
```

Убрать повторяющиеся значения в колонках

Как не выбрасывать другие колонки?

Задача - на каждой планете из датасета starwars найти два наименьших различных значения роста и посчитать среднее этих двух значений

```
starwars %>%
    group_by(homeworld) %>%
    distinct(height) %>%
    top_n(-2, height) %>%
    summarise(mean=mean(height))
```

```
## # A tibble: 49 x 2
##    homeworld       mean
##    <chr>          <dbl>
##  1 Alderaan         169
##  2 Aleen Minor       79
##  3 Bespin           175
##  4 Bestine IV       180
```

distinct

```
print(df)
```

```
##    x y
## 1 10 5
## 2  4 5
## 3  1 4
## 4  1 1
## 5  1 1
```

```
df %>% distinct(x, .keep_all=T)
```

```
##    x y
## 1 10 5
## 2  4 5
## 3  1 4
```

Как определяется то, какую из повторяющихся строк оставит?

distinct

```
print(df)
```

```
##    x y
## 1 10 5
## 2  4 5
## 3  1 4
## 4  1 1
## 5  1 1
```

```
df %>% distinct(x, .keep_all=T)
```

```
##    x y
## 1 10 5
## 2  4 5
## 3  1 4
```

Как определяется то, какую из повторяющихся строк оставит?

Надо посмотреть в мануал - оставим первую строку в таблице

RTFM

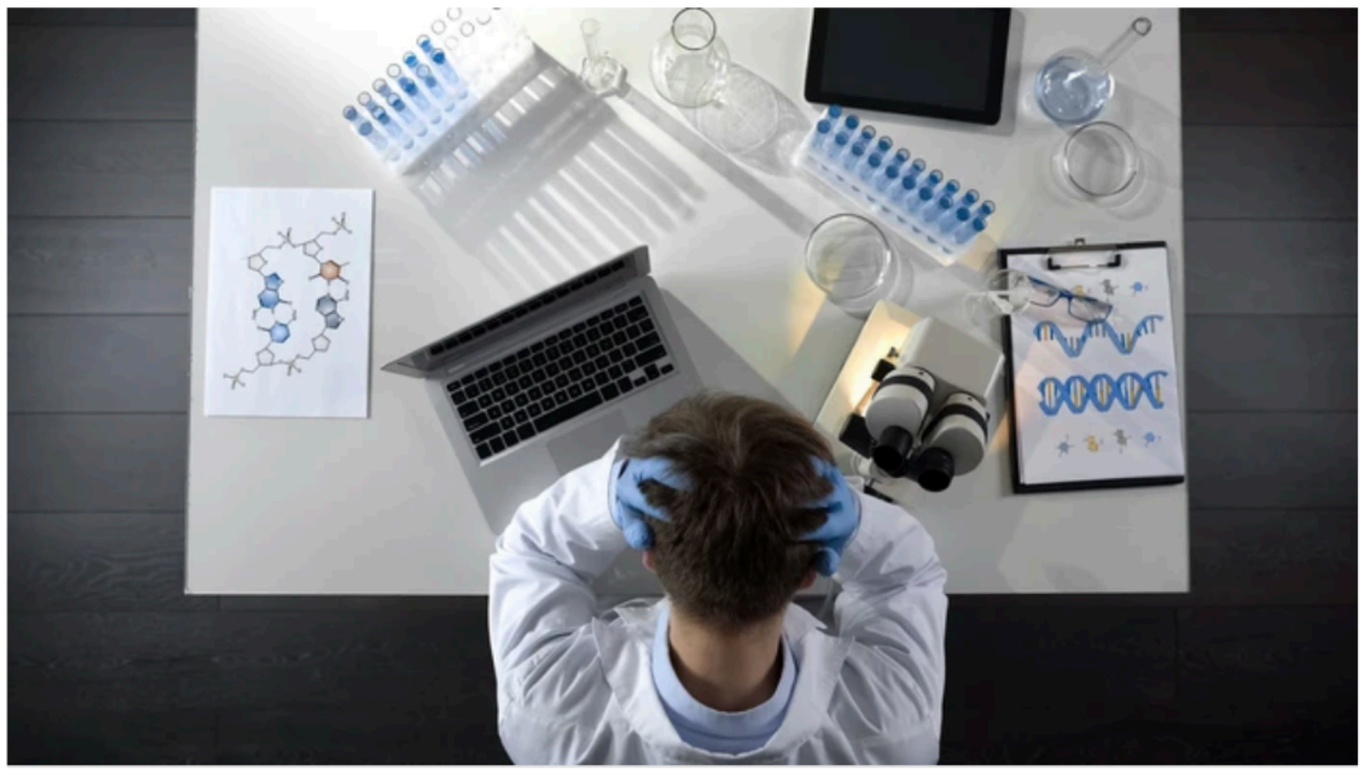
Если не RTFM, то

A Code Glitch May Have Caused Errors In More Than 100 Published Studies

The discovery is a reminder that science is collaborative and ideally self-correcting, but that nothing can be taken for granted.

By **Maddie Bender**

Oct 10 2019, 4:00pm Share Tweet



Люди “просто” не прочитали документацию питоновской функции glob.glob и никак не проверили, что их понимание того, как она работает, совпадает с реальностью

Если не RTFM,
да еще и сам M плохо написан

Misunderstood parameter of NCBI BLAST impacts the correctness of bioinformatics workflows

Nidhi Shah, Michael G Nute, Tandy Warnow, Mihai Pop ✉

Bioinformatics, Volume 35, Issue 9, 1 May 2019, Pages 1613–1614,
https://doi.org/10.1093/bioinformatics/bty833
Published: 24 September 2018 **Article history** ▾

Issue Section: SEQUENCE ANALYSIS

Associate Editor: John Hancock

distinct

```
df %>% distinct(x, y)
```

```
##    x y
## 1 10 5
## 2  4 5
## 3  1 4
## 4  1 1
```

Задача - представители какого цвета глаз оказываются с самым высоким ростом среди всех на планете чаще всего

```
starwars %>%
    group_by( homeworld) %>%
    top_n(1, height) %>%
    count(eye_color) %>%
    top_n(1, n)
```

```
## # A tibble: 49 x 3
## # Groups:   homeworld [49]
##    homeworld      eye_color      n
##    <chr>          <chr>      <int>
##  1 Alderaan       brown          1
##  2 Aleen Minor    unknown        1
##  3 Bespin         blue           1
##  4 Bestine IV     blue           1
```

Что неверно?

Задача - представители какого цвета глаз оказываются с самым высоким ростом среди всех на планете чаще всего

```
starwars %>%
    group_by( homeworld) %>%
    top_n(1, height) %>%
    ungroup() %>%
    count(eye_color) %>%
    top_n(1, n)
```

```
## # A tibble: 1 x 2
##   eye_color     n
##   <chr>     <int>
## 1 yellow        9
```

Задача - подсчитайте максимальный по всем планетам средний по планете рост для каждого цвета глаз

```
starwars %>%
    select(homeworld, eye_color, height)
```

```
## # A tibble: 87 x 3
##    homeworld eye_color height
##    <chr>     <chr>       <int>
##  1 Tatooine  blue          172
##  2 Tatooine  yellow        167
##  3 Naboo     red            96
##  4 Tatooine  yellow        202
##  5 Alderaan  brown         150
```

Выбираем нужные для решения строки, далее нужно сгруппировать по нескольким переменным

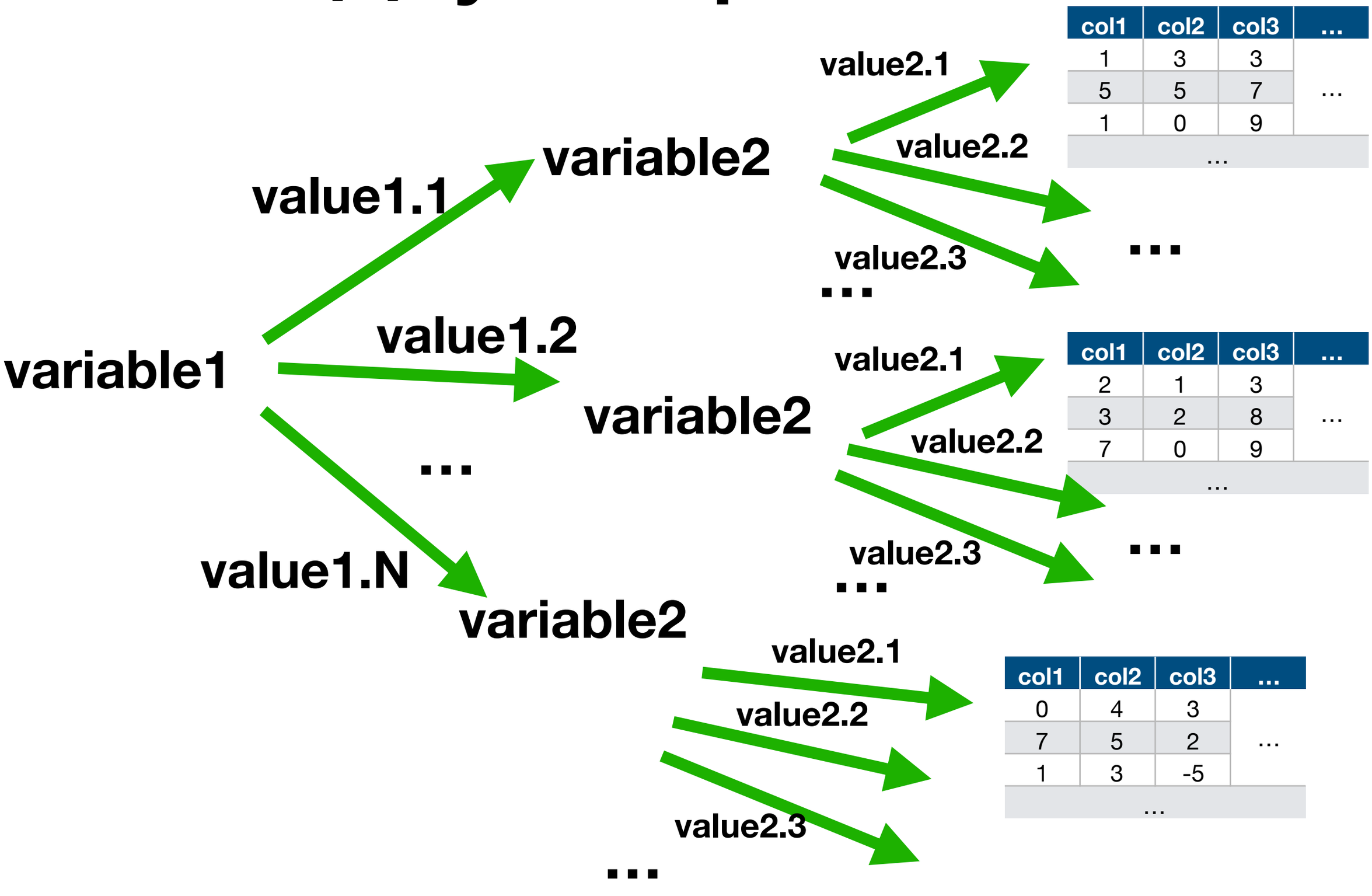
Задача - подсчитайте максимальный по всем планетам средний по планете рост для каждого цвета глаз

```
starwars %>%
    select(homeworld, eye_color, height) %>%
    group_by( homeworld, eye_color)
```

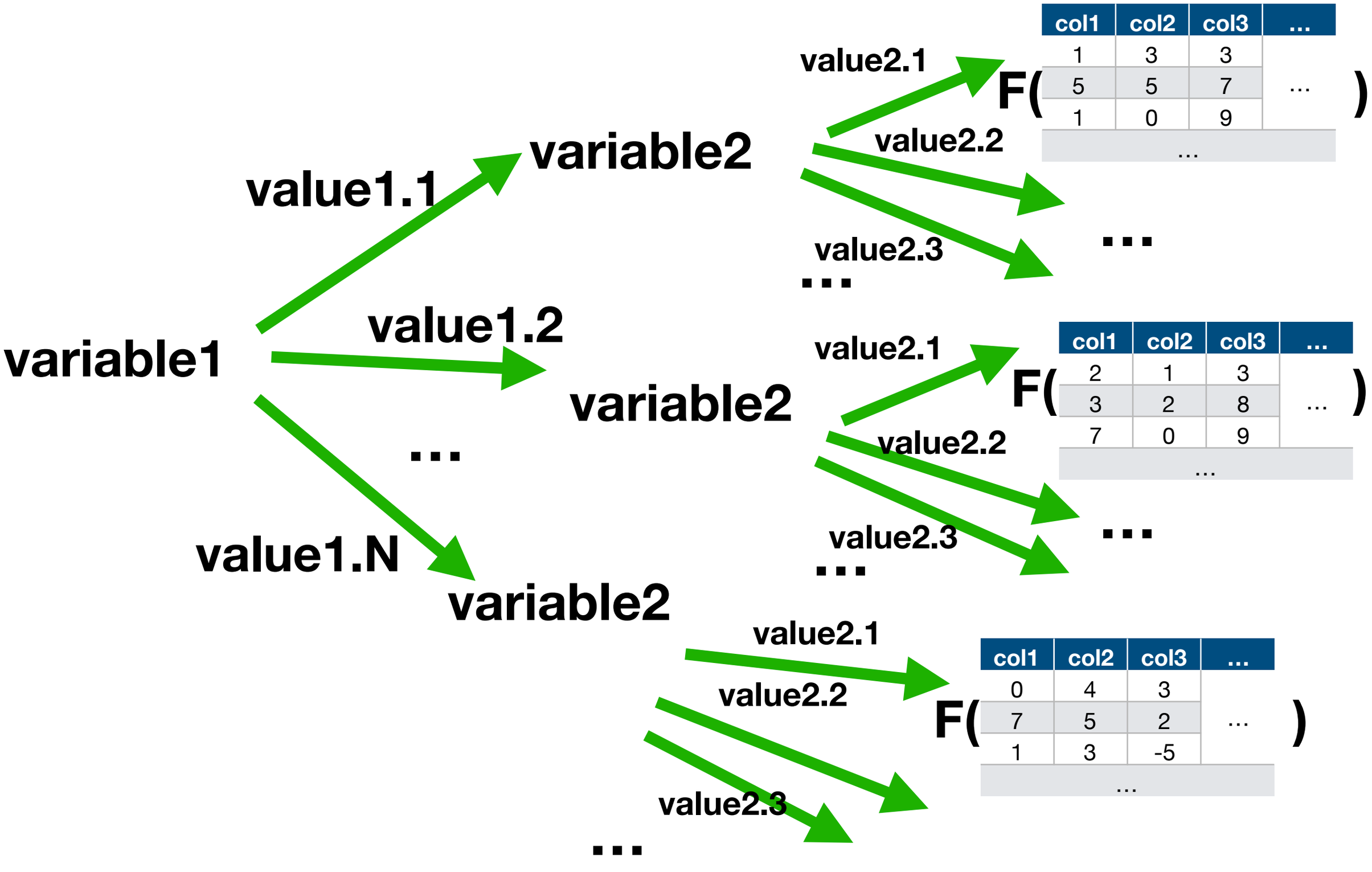
```
## # A tibble: 87 x 3
## # Groups:   homeworld, eye_color [66]
##    homeworld eye_color height
##    <chr>     <chr>      <int>
##  1 Tatooine  blue         172
##  2 Tatooine  yellow       167
##  3 Naboo     red           96
##  4 Tatooine  yellow       202
##  5 Alderaan  brown        150
```

Сгруппировали

Как выглядит groupby-объект по двум переменным



Как выполняется функция на groupby-объекте с двумя переменным



Задача - подсчитайте максимальный по всем планетам средний по планете рост для каждого цвета глаз

```
starwars %>%
    select(homeworld, eye_color, height) %>%
    group_by( homeworld, eye_color) %>%
    summarise(mean=mean(height))
```

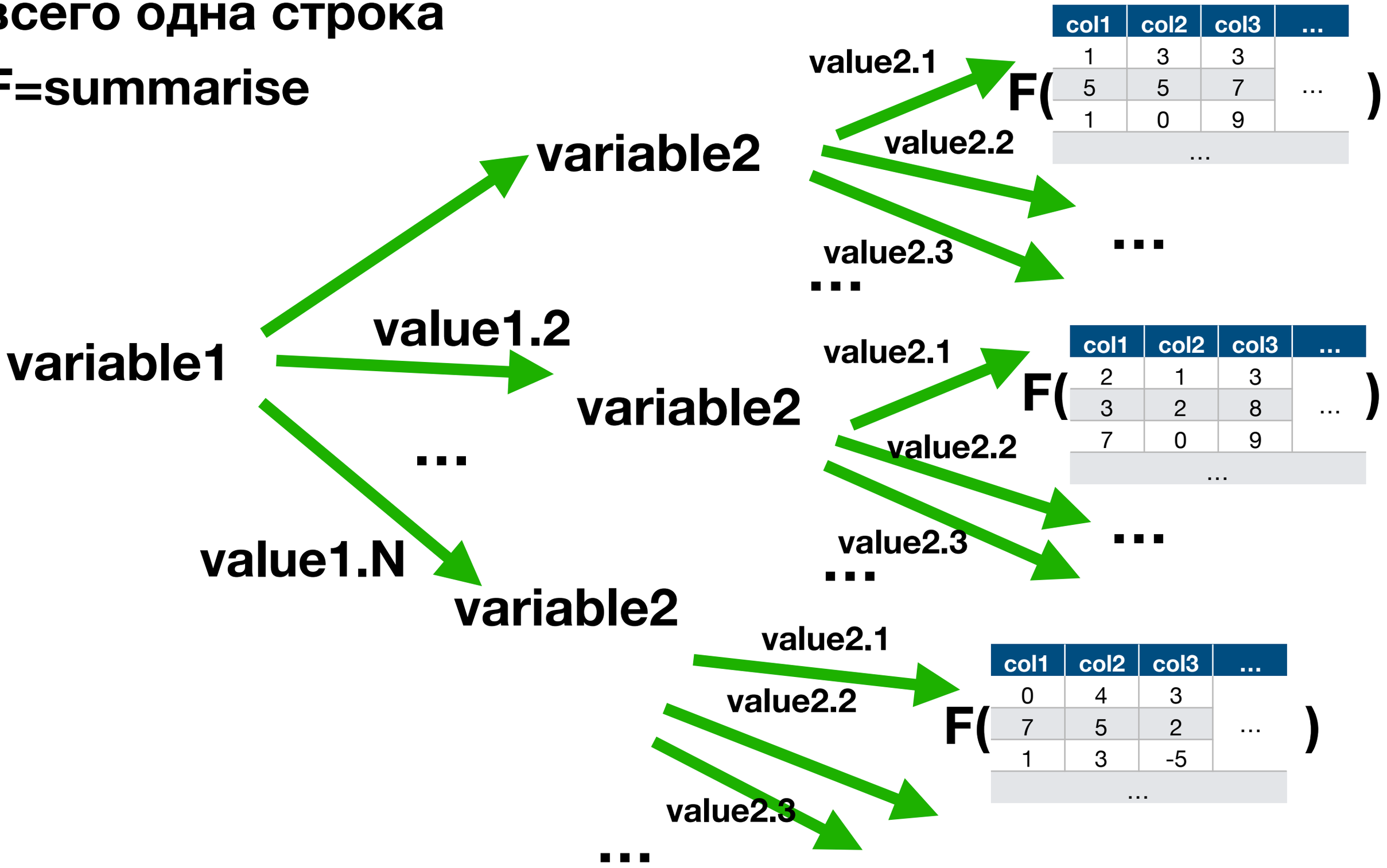
```
## # A tibble: 66 x 3
## # Groups:   homeworld [49]
##    homeworld      eye_color  mean
##    <chr>          <chr>     <dbl>
##  1 Alderaan       brown      176.
##  2 Aleen Minor    unknown     79
##  3 Bespin         blue       175
##  4 Bestine IV     blue       180
```

Считаем среднее в каждой подтаблице, одна строка

Поведеие summarise

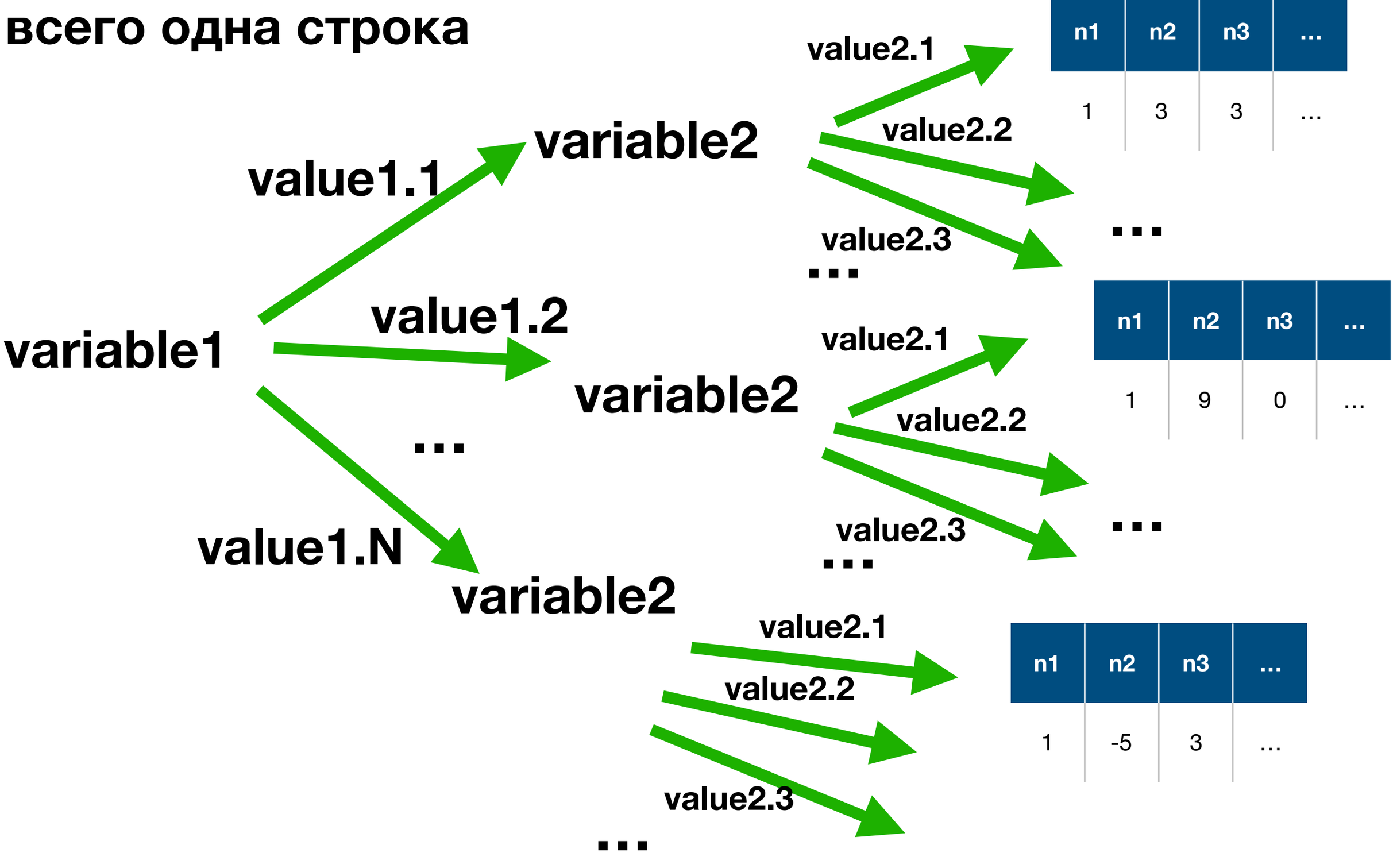
summarise сняло группировку, но только по eye_color, так как далее нет гарантий, что в получившихся группах всего одна строка

F=summarise



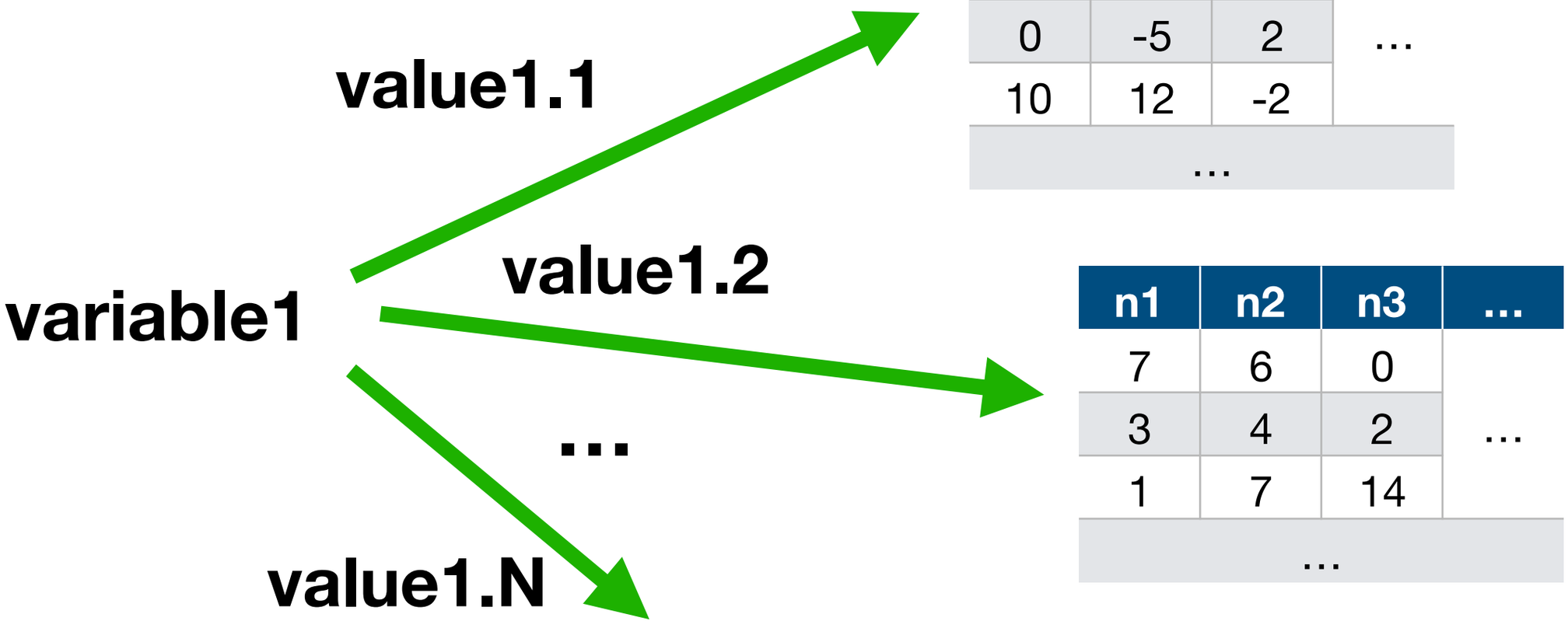
Поведеие summarise

summarise сняло группировку, но только по eye_color, так как далее нет гарантий, что в получившихся группах всего одна строка



Поведеие summarise

summarise сняло группировку, но только по eye_color, так как далее нет гарантий, что в получившихся группах всего одна строка



Задача - подсчитайте максимальный по всем планетам средний по планете рост для каждого цвета глаз

```
starwars %>%
    select(homeworld, eye_color, height) %>%
    group_by( homeworld, eye_color) %>%
    summarise(mean=mean(height)) %>%
    ungroup()
```

```
## # A tibble: 66 x 3
##    homeworld      eye_color  mean
##    <chr>          <chr>     <dbl>
##  1 Alderaan       brown      176.
##  2 Aleen Minor    unknown     79
##  3 Bespin         blue       175
##  4 Bestine IV     blue       180
##  5 Cato Neimoidia red        191
```

Убираем группу

Задача - подсчитайте максимальный по всем планетам средний по планете рост для каждого цвета глаз

```
starwars %>%
    select(homeworld, eye_color, height) %>%
    group_by( homeworld, eye_color) %>%
    summarise(mean=mean(height)) %>%
    ungroup() %>%
    group_by(eye_color) %>%
    top_n(1, mean)
```

```
## # A tibble: 14 x 3
## # Groups:   eye_color [14]
##    homeworld  eye_color       mean
##    <chr>      <chr>          <dbl>
##  1 Kalee      green, yellow   216
##  2 Kamino     black           221
##  3 Kashyyyk   blue            231
##  4 Muunilinst gold            191
##  5 Naboo      orange          209.
```

Еще раз группируем и выбираем top-1 mean для каждого цвета глаз

Задача - подсчитайте максимальный по всем планетам средний по планете рост для каждого цвета глаз

```
starwars %>%
    select(homeworld, eye_color, height) %>%
    group_by( homeworld, eye_color) %>%
    summarise(mean=mean(height)) %>%
    ungroup() %>%
    group_by(eye_color) %>%
    top_n(1, mean) %>% ungroup()
```

Ответ

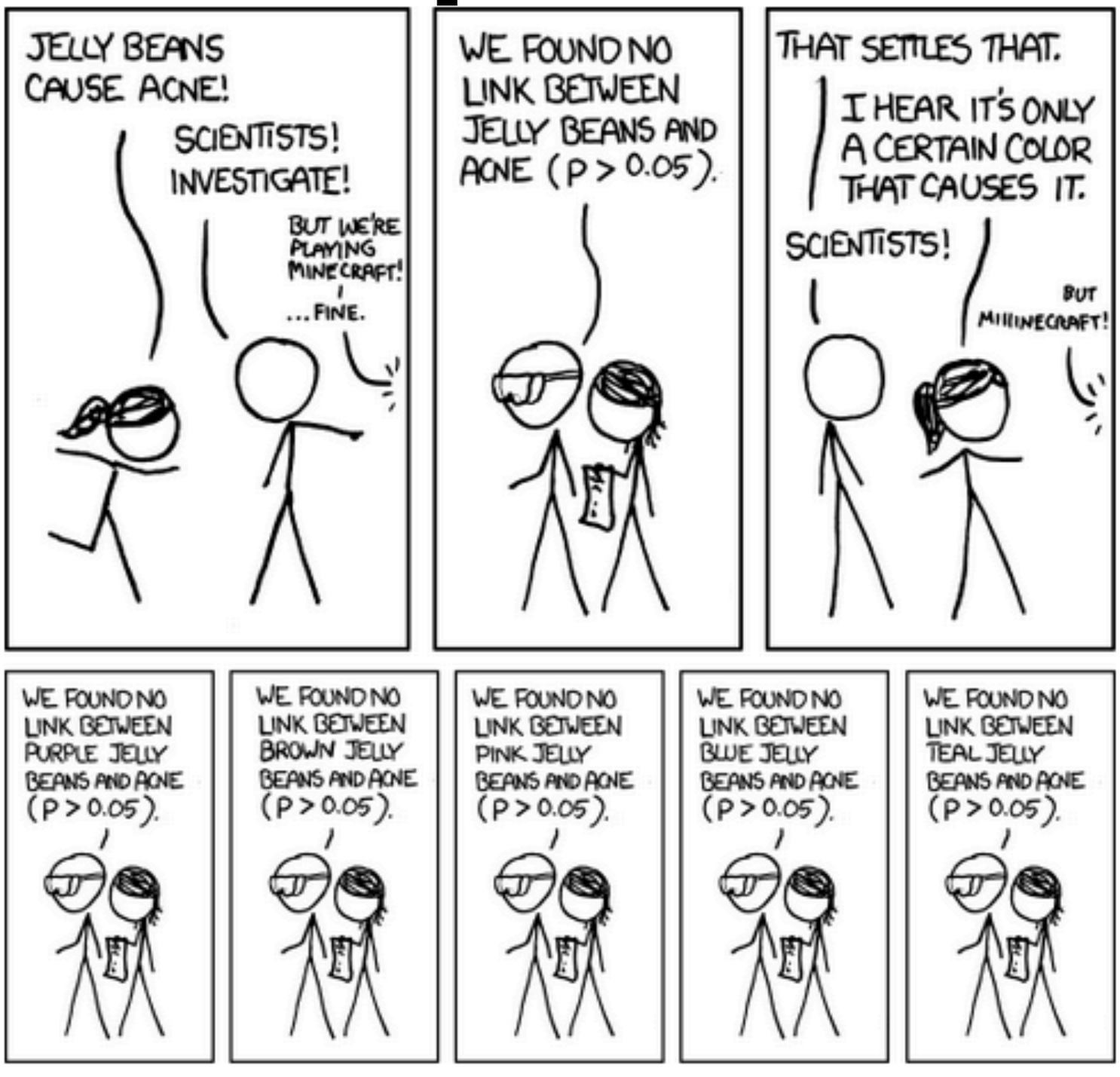
```
## # A tibble: 14 x 3
##    homeworld  eye_color       mean
##    <chr>      <chr>          <dbl>
##  1 Kalee      green, yellow   216
##  2 Kamino     black           221
##  3 Kashyyyk   blue            231
##  4 Muunilinst gold            191
```

Множественное тестирование

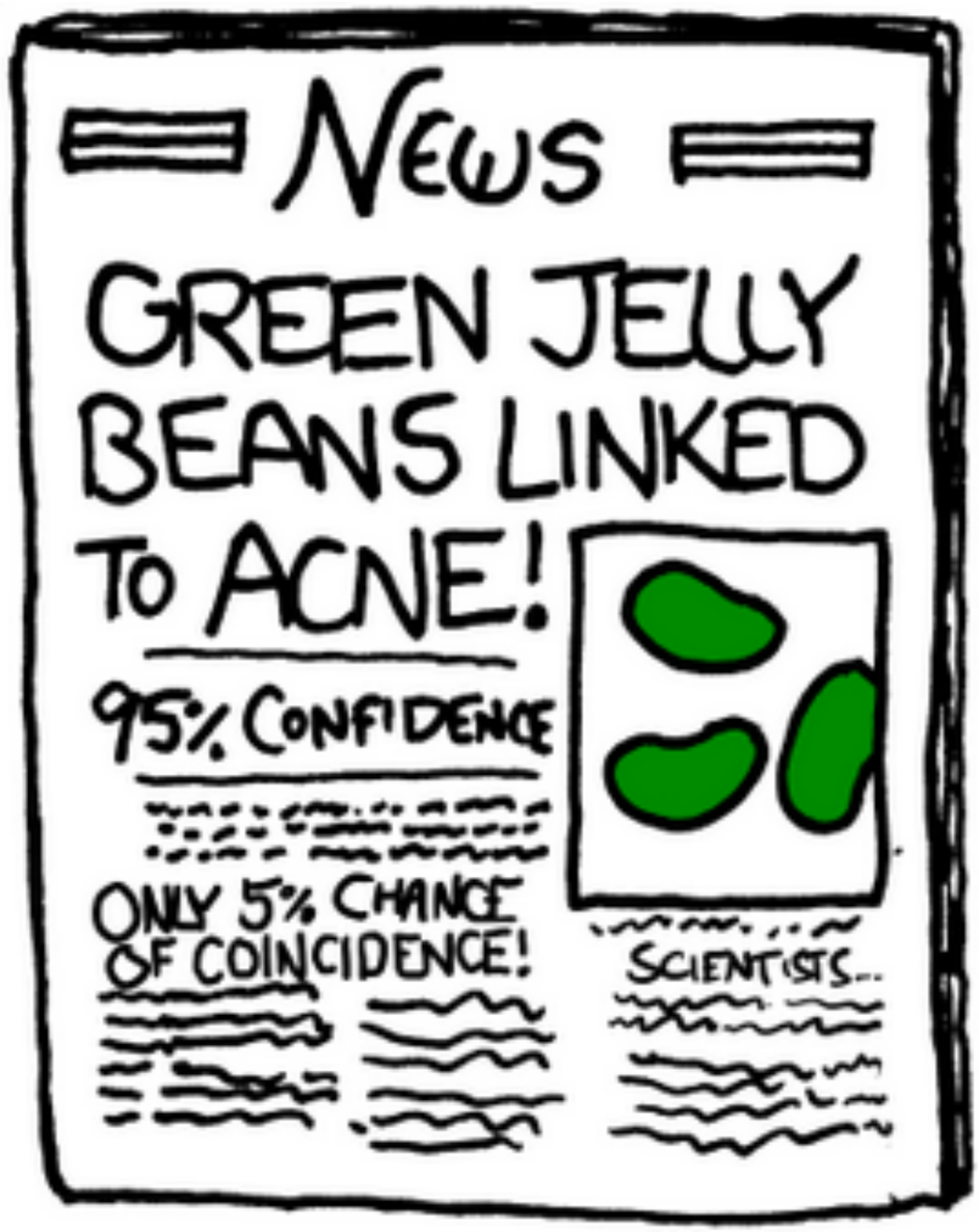
Не ищите того, чего нет. А то в конечном итоге - найдете. А это будет ошибка первого рода

@Игорь

Множественное тестирование



Множественное тестирование



Множественное тестирование

Рассмотрим датасет с 30000 генов, в котором нет ни одного дифференциально экспрессирующегося гена

Проведем t-test для каждого гена. Будем считать ген дифференциально экспрессируемым если $p < 0.05$.

Какова вероятность, что ни один ген не будет помечен как дифференциально экспрессируемый?

Сколько в среднем генов будет помечено как дифференциально экспрессируемые?

Поправки

- FWER (Family-Wise Error Rate) - вероятность, что среди отобранных генов хотя бы один ложно-положительный ген меньше заданного порога (0.05, например)

- FDR (False Discovery Rate) - процент ложно-положительных генов среди отобранных не больше, например, 20%

Смысл alpha разный для двух подходов

FWER

test	p-value
test1	p-value1
test2	p-value2
...	...
testN	p-valueN

Наша изначальная таблица

test	k	p-value
test1’	1	p-value1’
test2’	2	p-value2’
...	...	...
testM’	M	p-valueN’

Тесты, для которых мы отвергаем H0.

Гарантируем, что вероятность того, что во всей отобранной таблице встретится хотя бы один тест, для которого мы ошибочно отвергли H0 - alpha

FWER

One-step procedures:

1) Sidak correction
2) Bonferonni correction

Step-down procedures:

1) Holm-Sidak correction
2) Holm-Bonferonni correction

Step-up procedures:

Hochberg correction **Не рассматриваем**

FDR

test	p-value
test1	p-value1
test2	p-value2
…	…
testN	p-valueN

Наша изначальная таблица

test	k	p-value
test1’	1	p-value1’
test2’	2	p-value2’
…	…	…
testM’	M	p-valueN’

Тесты, для которых мы отвергаем H0. Гарантируем, что доля генов, для которых мы ошибочно отвергли H0 - alpha

p.adjust {stats}

Adjust P-values for Multiple Comparisons

Description

Given a set of p-values, returns p-values adjusted using one of several methods.

Usage

```
p.adjust(p, method = p.adjust.methods, n = length(p))

p.adjust.methods
# c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY",
#   "fdr", "none")
```

Возвращает скорректированные (adjusted) p-value
Это p-value, при сравнении которых с вашим alpha, меньше alpha окажутся те p-value, которые были бы отобраны соответствующим методом

Adjusted p-value

p-value, при сравнении которых с вашим alpha, меньше alpha окажутся те p-value, которые были бы отобраны соответствующим методом

Пример: сколько p-value из списка 0.01, 0.05, 0.04, 0.03, 0.001, 0.015, 0.20 останутся значимыми после поправки Холма-Бонферонни на уровне значимости alpha=0.05

```
alpha <- 0.05
pvals <-c(0.01, 0.05, 0.04, 0.03, 0.001, 0.015, 0.20)
adj_pvals <- p.adjust(pvals, method = 'holm')
sum(adj_pvals < 0.05)
```

```
## [1] 1
```

Adjusted p-value
One-step procedure

$$p < \frac{alpha}{N} = thres$$

На примере Бонферонни, мы можем записать следующее условие иначе

$$adjust\_p = p \cdot N < \alpha$$

Adjusted p-value

Step-down procedure

Неверный подход

$$p_k < \frac{\alpha}{N-k+1} = thres(k)$$

На примере Холма Бонферонни, мы можем записать следующее условие иначе

$$adjust\_p_k = p_k \cdot (N-k+1) < \alpha$$

Adjusted p-value

Step-down procedure

Неверный подход

$$p_k < \frac{\alpha}{N - k + 1} = thres(k)$$

На примере Холма Бонферонни, мы можем записать следующее условие иначе

$$adjust\_p_k = p_k \cdot (N - k + 1) < \alpha$$

Мы должны гарантировать, что все p-value, больше того, которое не прошло порог alpha (включая это p-value), будут больше alpha

Adjusted p-value

Step-down procedure

Верный подход

$$p_k < \frac{\alpha}{N - k + 1} = thres(k)$$

На примере Холма Бонферонни, мы можем записать следующее условие иначе

$$adjust\_p_1 = p_1 \cdot N < \alpha$$

$$adjust\_p_k = max(p_k \cdot (N - k + 1), p_{k-1} \cdot (N - k + 2)) < \alpha$$

Adjusted p-value

Step-down procedure.

Пример, Холм-Бонферонни

test	p-value	Порядок в сортировке (k)	adjusted p-value
test1	0.01	2	
test2	0.05	6	
test3	0.04	5	
test4	0.03	4	
test5	0.001	1	
test6	0.015	3	
test7	0.20	7	

Adjusted p-value

Step-down procedure.

Пример, Холм-Бонферонни

test	p-value	Порядок в сортировке (k)	adjusted p-value
test1	0.01	2	
test2	0.05	6	
test3	0.04	5	
test4	0.03	4	
test5	0.001	1	0.001 * 7 = 0.007
test6	0.015	3	
test7	0.20	7	

Adjusted p-value

Step-down procedure.

Пример, Холм-Бонферонни

test	p-value	Порядок в сортировке (k)	adjusted p-value
test1	0.01	2	min(0.007, 0.01 * 6)=0.06
test2	0.05	6	
test3	0.04	5	
test4	0.03	4	
test5	0.001	1	0.001 * 7 = 0.007
test6	0.015	3	
test7	0.20	7	

Adjusted p-value

Step-down procedure.

Пример, Холм-Бонферонни

test	p-value	Порядок в сортировке (k)	adjusted p-value
test1	0.01	2	0.06
test2	0.05	6	
test3	0.04	5	
test4	0.03	4	
test5	0.001	1	0.007
test6	0.015	3	min(0.06, 0.015 * 5)=0.075
test7	0.20	7	

Adjusted p-value

Step-down procedure.

Пример, Холм-Бонферонни

test	p-value	Порядок в сортировке (k)	adjusted p-value
test1	0.01	2	0.06
test2	0.05	6	0.120
test3	0.04	5	0.120
test4	0.03	4	0.120
test5	0.001	1	0.007
test6	0.015	3	0.075
test7	0.20	7	0.200

Adjusted p-value

Step-up procedure

Верный подход

$$p_k < \frac{k}{N}\alpha = thres(k)$$

На примере поправки Бенджамини-Хохберга, мы можем записать следующее условие иначе

$$adjust\_p_N = p_N < \alpha$$

$$adjust\_p_k = min(\frac{p_k \cdot N}{k}, \frac{p_{k+1} \cdot N}{k+1}) < \alpha$$

Adjusted p-value

Step-up procedure. Пример, Бенджамини-Хохберга

test	p-value	Порядок в сортировке (k)	adjusted p-value
test1	0.01	2	
test2	0.05	6	
test3	0.04	5	
test4	0.03	4	
test5	0.001	1	
test6	0.015	3	
test7	0.20	7	

Adjusted p-value

Step-up procedure. Пример, Бенджамини-Хохберга

test	p-value	Порядок в сортировке (k)	adjusted p-value
test1	0.01	2	
test2	0.05	6	
test3	0.04	5	
test4	0.03	4	
test5	0.001	1	
test6	0.015	3	
test7	0.20	7	0.20

Adjusted p-value

Step-up procedure. Пример, Бенджамини-Хохберга

test	p-value	Порядок в сортировке (k)	adjusted p-value
test1	0.01	2	
test2	0.05	6	min(0.05 * 7 / 6, 0.20) = 0.058
test3	0.04	5	
test4	0.03	4	
test5	0.001	1	
test6	0.015	3	
test7	0.20	7	0.20

Adjusted p-value

Step-up procedure. Пример, Бенджамини-Хохберга

test	p-value	Порядок в сортировке (k)	adjusted p-value
test1	0.01	2	
test2	0.05	6	0.058
test3	0.04	5	min(0.04 * 7 / 5, 0.058) = 0.056
test4	0.03	4	
test5	0.001	1	
test6	0.015	3	
test7	0.20	7	0.20

Adjusted p-value

Step-up procedure. Пример, Бенджамини-Хохберга

test	p-value	Порядок в сортировке (k)	adjusted p-value
test1	0.01	2	0.035
test2	0.05	6	0.058
test3	0.04	5	0.056
test4	0.03	4	0.053
test5	0.001	1	0.007
test6	0.015	3	0.035
test7	0.20	7	0.20