R language basics, part 3: factor

HUST Bioinformatics course series

Wei-Hua Chen (CC BY-NC 4.0)

18 September, 2024

section 1: TOC

前情提要

data frame and tibble

- declaration & usage
- manipulation （更多相关内容会在介绍 `dplyr` 时讲到）
- differences between data.frame and tibble
- advantages of using tibble (更多内容以后会介绍)
- `with`, `within`， `attach`, `detach` 等的用法

IO

- read from files of different formats
- write to files
- use GUI to read files (& get the corresponding code)

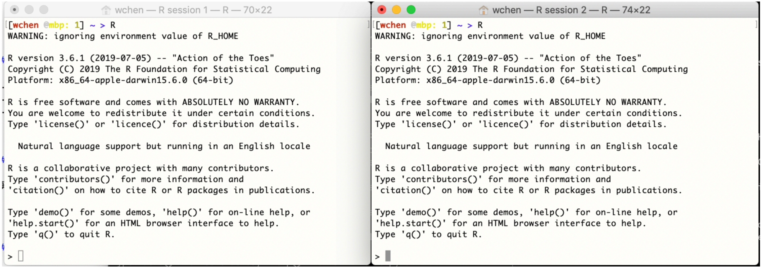
今次预报

1. IO, project management, working environment management
2. factors: R 中最重要的概念之一
3. exercises

section 2: IO and working enviroment management

R session 的概念

每个 R session 是一个单独的工作空间（work space），包含各自的数据、变量和操作历史。


wchen — R session 1 — R — 70×22
wchen — R session 2 — R — 74×22

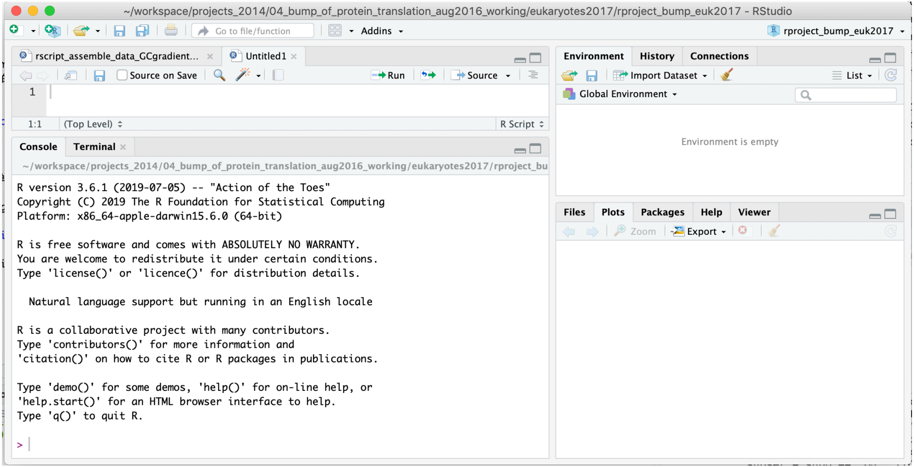
```
[wchen @mbp: 1] ~ > R
WARNING: ignoring environment value of R_HOME

R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

```
[wchen @mbp: 1] ~ > R
WARNING: ignoring environment value of R_HOME

R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

Figure 1: two R sessions

R session in RStudio

Each RStudio session is automatically associated with a R session


~/workspace/projects_2014/04_bump_of_protein_translation_aug2016_working/eukaryotes2017/rproject_bump_euk2017 - RStudio
Go to file/function
Addins
rproject_bump_euk2017
rscript_assemble_data_GCgradient...
Untitled1
Source on Save
Run
Source
1:1
(Top Level)
R Script
Console
Terminal
~/workspace/projects_2014/04_bump_of_protein_translation_aug2016_working/eukaryotes2017/rproject_bu
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Zoom
Export

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Figure 2: R session in RStudio

start a new RStudio session by creating a new project

1. 右上角的 Project 按钮，在弹出菜单里选 New Project …

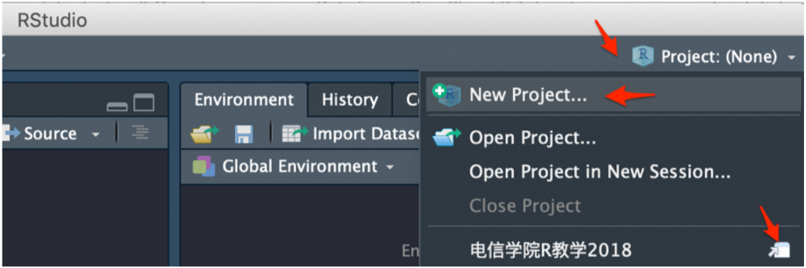
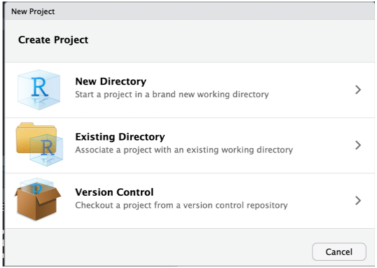


Figure 3: create new project, step 1

create a new project, cont.

2. Select: New directory -> New Project in the popup window



New Project

Create Project

New Directory
Start a project in a brand new working directory

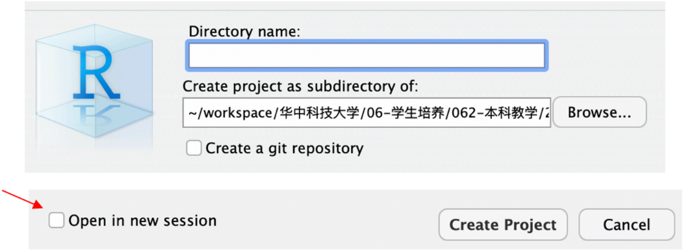
Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Figure 4: create new project, step 2

create a new project, cont.

3. Enter a new directory name, choose its mother directory ...



Directory name:

Create project as subdirectory of: ~/workspace/华中科技大学/06-学生培养/062-本科教学/

□ Create a git repository

□ Open in new session

Figure 5: create new project, step 3

现场演示

演示 ~~

working space

当前工作空间，包括所有已装入的数据、包和自制函数

可通过以下代码管理变量

```
ls();  ## 显示当前环境下所有变量
```

```
## [1] "color_block"
```

```
rm( x ); ## 删除一个变量
```

```
## Warning in rm(x): object 'x' not found
```

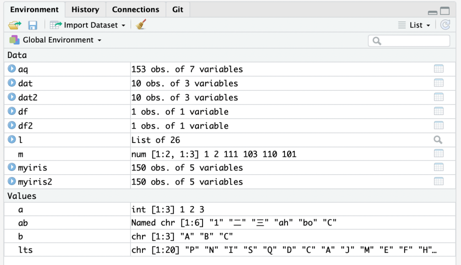
```
ls();
```

```
## [1] "color_block"
```

```
##rm(list=ls()); ## 删除当前环境下所有变量!!!
```

variables in working space in RStudio

在 RStudio 右上角的"Environment" 窗口显示了所有当前工作间的变量



Data	
aq	153 obs. of 7 variables
dat	10 obs. of 3 variables
dat2	10 obs. of 3 variables
df	1 obs. of 1 variable
df2	1 obs. of 1 variable
l	List of 26
m	num [1:2, 1:3] 1 2 111 103 110 101
myiris	150 obs. of 5 variables
myiris2	150 obs. of 5 variables
Values	
a	int [1:3] 1 2 3
ab	Named chr [1:6] "1" "二" "三" "ah" "bo" "C"
b	chr [1:3] "A" "B" "C"
lts	chr [1:20] "P" "N" "I" "S" "Q" "D" "C" "A" "J" "M" "E" "F" "H"…

Figure 6: RStudio enviroment window

save and restore work space

```r
## -- save all loaded variables into an external .RData file
save.image( file = "prj_r_for_bioinformatics_aug3_2019.RData" );

## -- restore ( load ) saved work space
load( file = "prj_r_for_bioinformatics_aug3_2019.RData" );
```

Notes

- existing variables will be kept, however, those will the same names will be replaced by loaded variables
- please consider using `rm( list=ls() )` to remove all existing variables to have a clean start
- you may need to reload all the packages

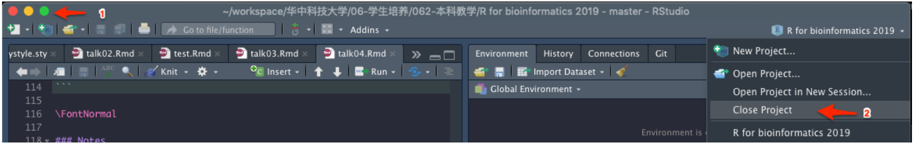
save selected variables

Sometimes you need to transfer processed data to a collaborator ...

```r
## save selected variables to external
save(city, country, file="1.RData"); ## you can specify directory name

## --
load( "1.RData" );
```

close and (re)open a project

close a project is easy:

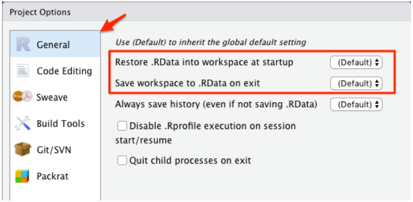


```
114  ```
115
116  \FontNormal
117
118  ### Notes
```

Figure 7: Two ways of closing a project

however ...

退出 projects 时的一些选项（RStudio）



Use (Default) to inherit the global default setting

Restore .RData into workspace at startup (Default)

Save workspace to .RData on exit (Default)

Always save history (even if not saving .RData) (Default)

□ Disable .Rprofile execution on session start/resume

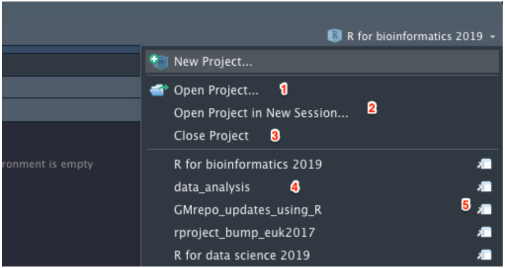
□ Quit child processes on exit

Figure 8: Project options

notes

- 退出时保存
- 打开时装入
- 但数据较大时，装入时间可能过长 ...

open a project


R for bioinformatics 2019

New Project...

Open Project... 1

Open Project in New Session... 2

Close Project 3

R for bioinformatics 2019

data_analysis 4

GMrepo_updates_using_R 5

rproject_bump_euk2017

R for data science 2019

ronment is empty

Figure 9: Open a project

演示项目的不同打开姿式（1-5）。

练习

- 创建一个项目
- 定义一些变量
- 从外部文件装入一些数据
- 保存 workspace 到.RData
- 退出 project
- 重新打开 project 并恢复 workspace

section 3: factors

什么是 factors？

Factor is a data structure used for fields that takes only predefined, finite number of values (categorical data).

factor 的作用

- 限制输入范围
- 有意义的排序
- 作图
- 其它
- factor 造成的不便

1. factor 用于限制输入范围

Factor 用于限制某个字段（列），只允许其接受某些值

```
x <- c("single", "married", "married", "single");
str(x);
```

```
##  chr [1:4] "single" "married" "married" "single"
```

```
## create factor as it is ...
x <- as.factor(x);

## please note the change in the displayed values ...
str(x);
```

```
##  Factor w/ 2 levels "married","single": 2 1 1 2
```

```
## create factor from scratch ...
x <- factor( c( "single", "married", "married", "single" ) );
str(x);
```

```
##  Factor w/ 2 levels "married","single": 2 1 1 2
```

** 注意 **： 用 as.factor 创建 factor 时，得到的 levels 按字母表排列；

输入不允许的值时，会产生 NA

```
str(x);
```

```
##  Factor w/ 2 levels "married","single": 2 1 1 2
```

```
x[ length(x) + 1 ] <- "widowed";
```

```
## Warning in `[<-.factor`(`*tmp*`, length(x) + 1, value = "widowed"): invalid
## factor level, NA generated
```

```
x;
```

```
## [1] single  married married single  <NA>
## Levels: married single
```

使用 levels() 函数调整 factor 的内容

```
levels(x) <- c(levels(x), "widowed");
x[ length(x) + 1 ] <- "widowed";
str(x);
```

```
##  Factor w/ 3 levels "married","single",..: 2 1 1 2 NA 3
```

注意：用 levels(y) 方式指定 levels 时，则按照指定的顺序；

levels 的顺序决定了排序的顺序

```
##
y <-  as.factor( c( "single", "married", "married", "single" ) );
levels(y);
```

```
## [1] "married" "single"
```

```
sort(y);
```

```
## [1] married married single  single
## Levels: married single
```

```
##
y2 <- y;
levels(y2) <- c("single", "married", "widowed");
sort(y2);
```

```
## [1] single  single  married married
## Levels: single married widowed
```

2. 用 factor 做有意义的排序

无意义的排序

```
## Month
x1 <- c("Dec", "Apr", "Jan", "Mar");
sort(x1);
```

```
## [1] "Apr" "Dec" "Jan" "Mar"
```

有意义的排序

```
month_levels <- c(
  "Jan", "Feb", "Mar", "Apr", "May", "Jun",
  "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"
)

y1 <- factor(x1, levels = month_levels)
sort(y1);
```

```
## [1] Jan Mar Apr Dec
## Levels: Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
```

3. factor 用于作图（真正精髓）

举例：对一项调查结果进行统计并作图

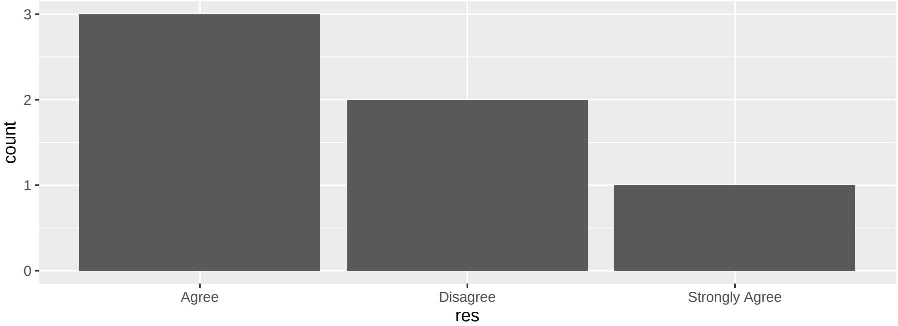
是否增加更多作业???

```r
## 同学们的回答:
( responses <- factor( c("Agree", "Agree", "Strongly Agree", "Disagree",
                         "Disagree", "Agree") ) );
```

```
## [1] Agree          Agree          Strongly Agree Disagree       Disagree
## [6] Agree
## Levels: Agree Disagree Strongly Agree
```

```r
## -- plot the results --
library(ggplot2);
barplot <-
  ggplot( data = data.frame( res = responses ), aes( x = res )) +
  geom_bar();
```

作图结果



默认情况下，factor 按字母表排序： Agree -> Disagree -> Strong Agree 。ggplot2 也会按 factor 的排序作图

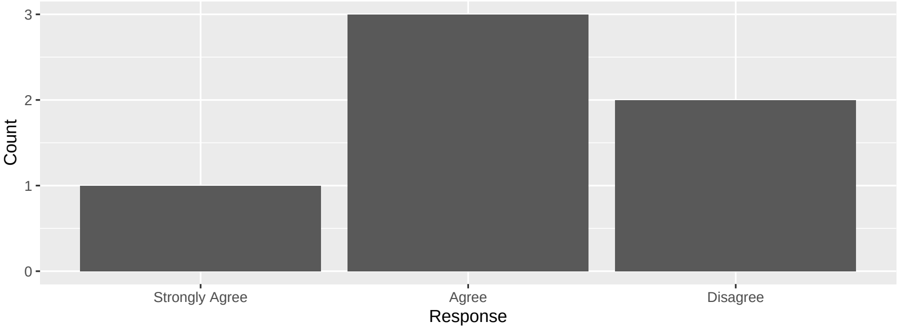
调整 factor 使统计结果更合理

```
res <- data.frame( res=responses );
## -- 按照同意程度从强-> 弱 排序
res$res <- factor( res$res, levels = c( "Strongly Agree", "Agree",  "Disagree" ), ordered = T )
str(res);
```

```
## 'data.frame':    6 obs. of  1 variable:
##  $ res: Ord.factor w/ 3 levels "Strongly Agree"<..: 2 2 1 3 3 2
```

```
plot2 <-
  ggplot( data = res, aes( x = res )) +
  geom_bar() +
  xlab( "Response" ) + ylab("Count");
```

调整结果



ordered factor

通过 ordered 参数，让用户知道 factors 是经过精心排序的

```
( responses <- factor( c("Agree", "Agree", "Strongly Agree", "Disagree", "Disagree", "Agree"),
                       levels = c( "Disagree", "Agree", "Strongly Agree"), ordered = T ) );
```

```
## [1] Agree          Agree          Strongly Agree Disagree       Disagree
## [6] Agree
## Levels: Disagree < Agree < Strongly Agree
```

```
is.ordered( responses );
```

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

4. factor 的其它应用

通过 factor 改变值

使用 dplyr 包的 recode() 函数改变 value

```
( x <- factor( c( "alpha", "beta", "gamma", "theta", "beta", "alpha" ) ) );
```

```
## [1] alpha beta  gamma theta beta  alpha
## Levels: alpha beta gamma theta
```

```
## --
library( dplyr );
x <- recode( x, "alpha" = "one", "beta" = "two" );
str(x);
```

```
##  Factor w/ 4 levels "one","two","gamma",..: 1 2 3 4 2 1
```

use factor to clean data

```r
library(forcats); ## just to make sure the codes will run smoothly ...

## 假设我有一组性别数据，其写法非常不规整；
gender <- c("f", "m ", "male ","male", "female", "FEMALE", "Male", "f", "m");

## 要求：都改为 Female, Male
gender <- as_factor( gender );
fct_count( gender );
```

```
## # A tibble: 8 x 2
##   f             n
##   <fct>     <int>
## 1 "f"           2
## 2 "m "          1
## 3 "male "       1
## 4 "male"        1
## 5 "female"      1
## 6 "FEMALE"      1
## 7 "Male"        1
## 8 "m"           1
```

```r
gender <- fct_collapse(
  gender,
  Female = c("f", "female", "FEMALE"),
  Male   = c("m ", "m", "male ", "male", "Male")
)
```

调整的结果

```
fct_count(gender);
```

```
## # A tibble: 2 x 2
##   f          n
##   <fct>  <int>
## 1 Female     4
## 2 Male       5
```

or use fct_relabel

```r
gender <- c("f", "m ", "male ","male", "female", "FEMALE", "Male", "f", "m")
gender <- as_factor(gender)
gender <- fct_relabel(gender, ~ ifelse(tolower(substring(., 1, 1)) == "f", "Female", "Male"))

fct_count(gender)
```

```
## # A tibble: 2 x 2
##   f          n
##   <fct>  <int>
## 1 Female     4
## 2 Male       5
```

5. factor 造成的不便

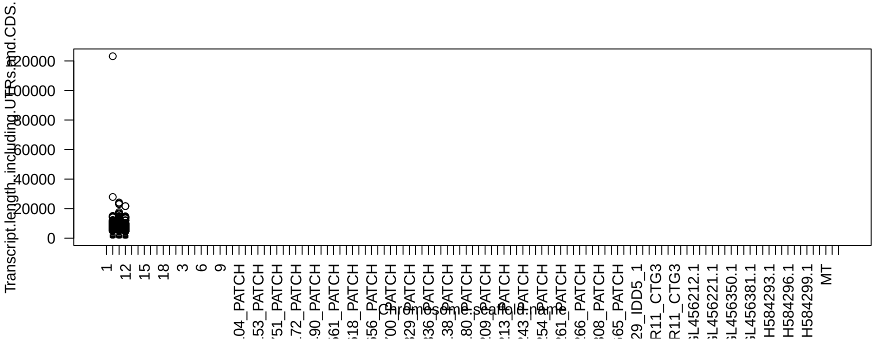
先读取一个文件，注意 stringsAsFactors 参数（默认为 F）；

```
mouse.genes <- read.delim( file = "data/talk04/mouse_genes_biomart_sep2018.txt",
                           sep = "\t", header = T, stringsAsFactors = T );

str(mouse.genes);
```

```
## 'data.frame':    138532 obs. of  6 variables:
##  $ Gene.stable.ID                                : Factor w/ 55029 levels "ENSMUSG00000000001",.
##  $ Transcript.stable.ID                          : Factor w/ 138532 levels "ENSMUST00000000001",
##  $ Protein.stable.ID                             : Factor w/ 65897 levels "","ENSMUSP00000000001
##  $ Transcript.length..including.UTRs.and.CDS.: int  67 67 1144 69 519 1824 71 59 67 1378 ...
##  $ Transcript.type                               : Factor w/ 48 levels "3prime_overlapping_ncRNA
##  $ Chromosome.scaffold.name                      : Factor w/ 117 levels "1","10","11",..: 115 11
```

从 mouse.genes 中取部分数据

```
mouse.chr_10_12 <- subset( mouse.genes,  Chromosome.scaffold.name %in% c( "10", "11", "12" ) );
## plot length distribution --

boxplot( Transcript.length..including.UTRs.and.CDS. ~ Chromosome.scaffold.name,
         data = mouse.chr_10_12, las = 2 );
```



问题!!: subset() 无法去除不用的 factors ...

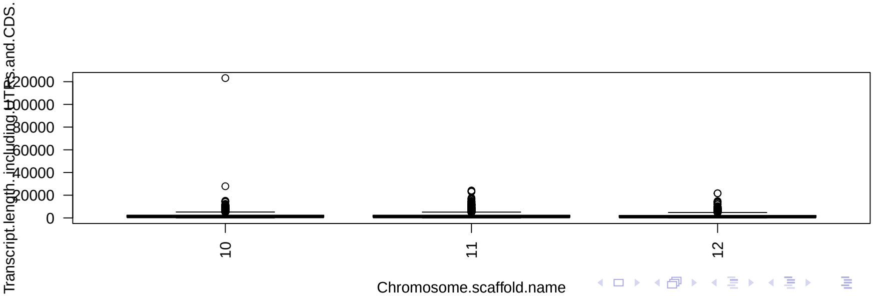
如何去除不用的 levels？

使用 droplevels 函数；

```
mouse.chr_10_12$Chromosome.scaffold.name <-
  droplevels( mouse.chr_10_12$Chromosome.scaffold.name );
levels( mouse.chr_10_12$Chromosome.scaffold.name );
```

```
## [1] "10" "11" "12"
```

```
## 再次 plot ...
boxplot( Transcript.length..including.UTRs.and.CDS. ~ Chromosome.scaffold.name,
         data = mouse.chr_10_12, las = 2 );
```



也可以使用 tibble ，完全不用担心 factor 的问题 ...

```
library( readr );
mouse.tibble <- read_delim( file = "data/talk04/mouse_genes_biomart_sep2018.txt",
                            delim = "\t", quote = "" )
```

```
## Rows: 138532 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (5): Gene stable ID, Transcript stable ID, Protein stable ID, Transcript...
## dbl (1): Transcript length (including UTRs and CDS)
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

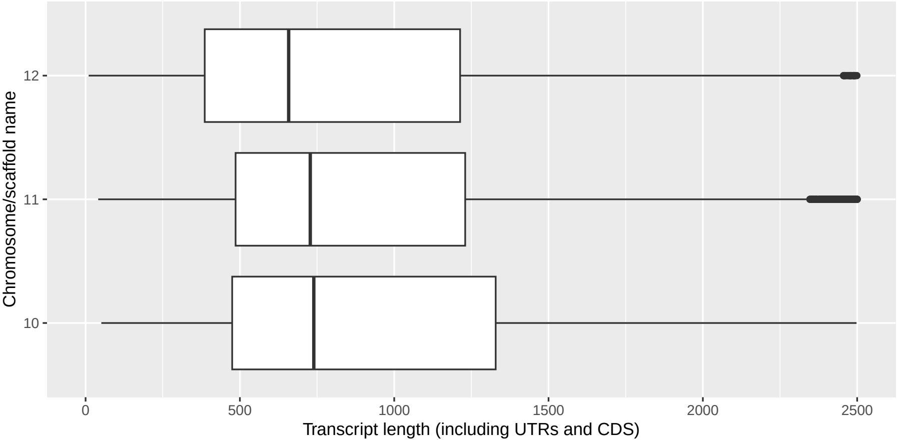
```
mouse.tibble.chr10_12 <-
  mouse.tibble %>% filter( `Chromosome/scaffold name` %in%  c( "10", "11", "12" ) );

plot3 <-
  ggplot( data = mouse.tibble.chr10_12,
        aes( x = `Chromosome/scaffold name`,
             y = `Transcript length (including UTRs and CDS)` ) ) +
  geom_boxplot() +
  coord_flip() +
  ylim( 0, 2500 ) ;
```

另一种不显示 outlier 的方法（更科学）

```r
p1 <-
  ggplot( data = mouse.tibble.chr10_12,
        aes( x = `Chromosome/scaffold name`,
             y = `Transcript length (including UTRs and CDS)` ) ) +
  geom_boxplot() +
  coord_flip(ylim = c(0, 5000), clip = "on");
```

用 tibble 解决 factor 的问题，cont.

```
## Warning: Removed 4770 rows containing non-finite outside th
## (`stat_boxplot()`).
```

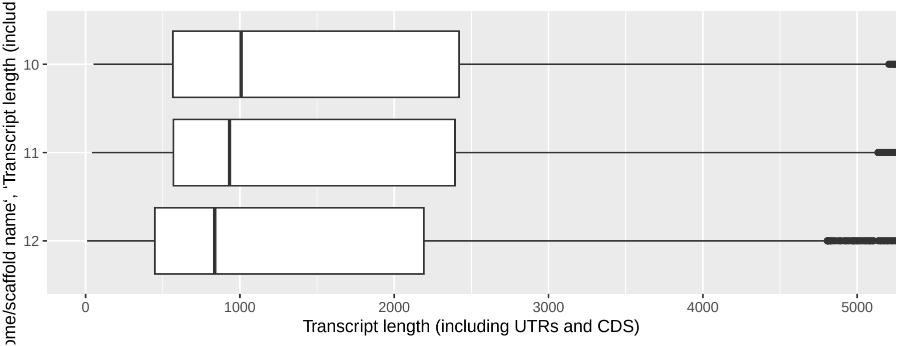


按基因长度中值从大 -> 小排序

```
plot4 <-
  ggplot( data = mouse.tibble.chr10_12,
        aes( x = reorder( `Chromosome/scaffold name`,
                          `Transcript length (including UTRs and CDS)`,
                          median ),
             y = `Transcript length (including UTRs and CDS)` ) ) +
  geom_boxplot() +
  coord_flip(ylim = c(0, 5000), clip = "on")
```

`reorder( vector_with_factor, numeric_value , FUN = mean  )` 的用法

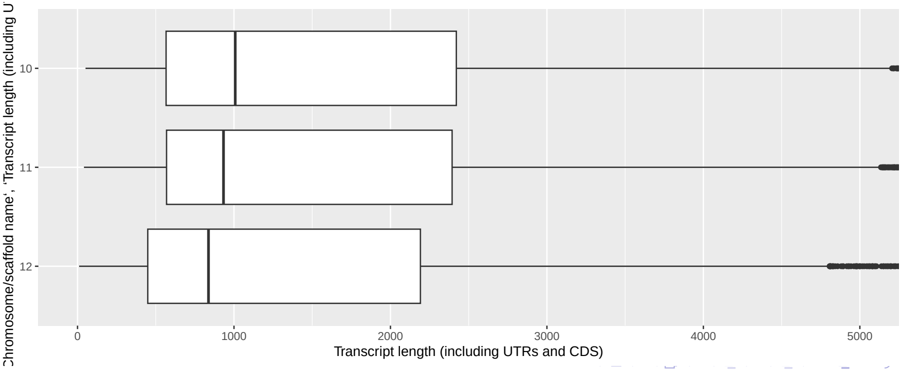
按基因长度中值从大 -> 小排序，cont.



** 注意 ** reorder(`Chromosome/scaffold name`, - `Transcript length (including UTRs and CDS)`, median) 的作用

use forcats::fct_reorder to reorder factors

```
ggplot( data = mouse.tibble.chr10_12,
      aes( x = fct_reorder( `Chromosome/scaffold name`,
                        `Transcript length (including UTRs and CDS)`,
                        median ),
           y = `Transcript length (including UTRs and CDS)` ) ) +
geom_boxplot() +
coord_flip(ylim = c(0, 5000), clip = "on");
```



section 4: 练习 & 作业

小结

今次提要

- IO, project management, working environment management
- factor ： R 另一个超级重要的概念
  - 定义
  - 操作
  - 使用
- 基础和进阶绘图（配合 factor 讲解）

下次预告

- data-wrangler: `dplyr`

important

- all codes are available at Github: `https://github.com/evolgeniusteam/R-for-bioinformatics`

练习 & 作业

- `Exercises and homework` 目录下 `talk04-homework.Rmd` 文件;
- 完成时间： 见钉群的要求