

PART-1. *R* 설치 및 기초 문법



1. R 설치 및 개요
2. 데이터 유형과 구조
3. 데이터 입출력
4. 제어문과 함수



1. R 설치 및 개요

Chap01_Basic 수업내용

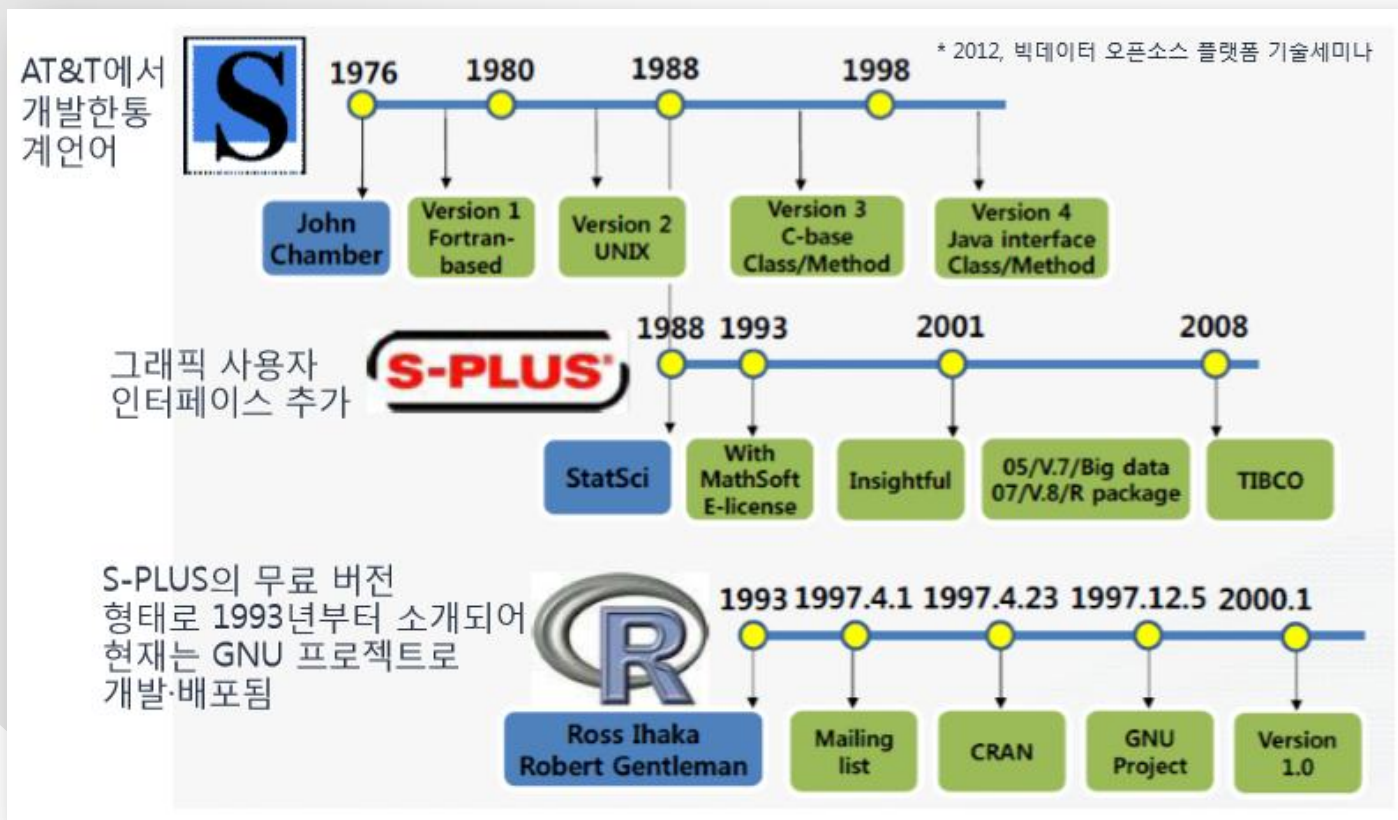
1. R 설치 및 개요(Overview)
2. 변수와 데이터 유형
3. 패키지 사용
 - 패키지 개념, 기본 설치 패키지
 - 패키지 설치 확인, 사용/제거
4. 기본 함수 사용
5. 작업공간
6. 한글 저장 및 깨짐 현상 처리 방법 : UTF-8



1. R 설치 및 개요

- R 프로그램 탄생과 발전과정

- ✓ 1993년 뉴질랜드 오클랜드 대학의 통계학과 교수2명(Ross Ihaka, Robert Gentleman) 개발





1. R 설치 및 개요

● R 프로그램 특징

I. In-Memory Computing

- ✓ 빠른 처리 속도(H/W 메모리 크기에 영향 받음)

II. Object-oriented programming

- ✓ 데이터, 함수, 차트 등 모든 것이 object로 관리

III. Package

- ✓ 최신의 알고리즘 및 방법론이 Package로 제공됨

IV. Visualization

- ✓ 분석에 통찰을 부여할 수 있는 그래픽에 대한 강력한 지원



1. R 설치 및 개요

<http://www.r-project.org/>

✓ CRAN Site에 패키지 0,000개 등록: `dim(available.packages())`

The screenshot shows two browser windows. The main window displays the R Project homepage at <http://www.r-project.org/>. The page features the R logo, a 'Download' button (highlighted with a red box) linking to the CRAN site, and a list of links for 'About R', 'Contributors', 'What's New?', 'Mailing Lists', 'Bug Tracking', 'Conferences', and 'Search'. Below these are links for the 'R Foundation' and 'Documentation'. The 'Getting Started' section explains that R is a free software environment for statistical computing and data manipulation, available on various UNIX platforms. It also mentions that users can choose their preferred CRAN mirror. The 'News' section lists recent updates, including R 3.2.1 (World-Famous Astronaut), R version 3.2.0 (Full of Ingredients), R version 3.1.3 (Smooth Sidewalk), and The R Journal Volume 6/2. A secondary window shows the CRAN mirrors page at <http://cran.r-project.org/mirrors>, listing mirrors for Ireland, Italy, Japan, Korea (highlighted with a red box), Lebanon, and Mexico. The Korea section lists three mirrors: <http://cran.nexr.com/>, <http://healthstat.snu.ac.kr/CRAN/>, and <http://cran.biodisk.org/>.



1. R 설치 및 개요

R console 화면

```
R Console

R version 3.1.1 (2014-07-10) -- "Sock it to Me"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R은 자유 소프트웨어이며, 어떠한 형태의 보증없이 배포됩니다.
또한, 일정한 조건하에서 이것을 재배포 할 수 있습니다.
배포와 관련된 상세한 내용은 'license()' 또는 'licence()'을 통하여 확인할 수 있습니다.

R은 많은 기여자들이 참여하는 공동프로젝트입니다.
'contributors()'라고 입력하시면 이에 대한 더 많은 정보를 확인하실 수 있습니다.
그리고, R 또는 R 패키지들을 출판물에 인용하는 방법에 대해서는 'citation()'을 통해 확인하시길 부탁드립니다.

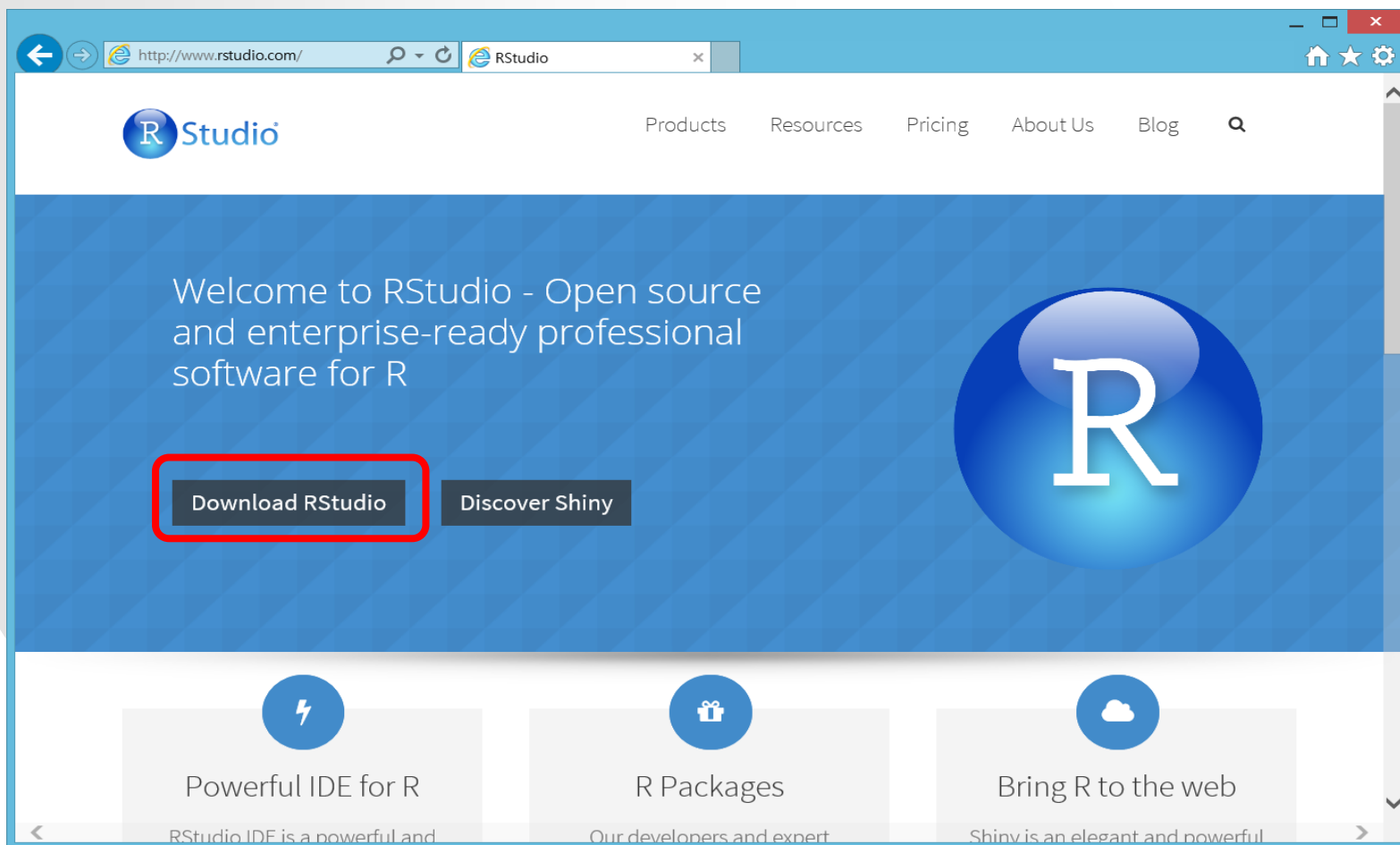
'demo()'를 입력하신다면 몇가지 데모를 보실 수 있으며, 'help()'를 입력하시면 온라인 도움말을 이용하실 수 있습니다.
또한, 'help.start()'의 입력을 통하여 HTML 브라우저에 의한 도움말을 이용하실 수 있습니다.
R의 종료를 원하시면 'q()'을 입력해주세요.

> v <- c(11,12,13,14,15)
> w <- c(1,2,3,4,5)
> v+w
[1] 12 14 16 18 20
> v*w
[1] 11 24 39 56 75
> v/w
[1] 11.000000  6.000000  4.333333  3.500000  3.000000
> v^w
[1]      11      144     2197    38416   759375
> |
```



1. R 설치 및 개요

<http://www.rstudio.com/>





1. R 설치 및 개요

Desktop 선택

The image displays two browser windows from the RStudio website. The left window shows the 'Products' page with the 'Desktop' option highlighted by a red box. The right window shows the 'Download RStudio' page with the 'RStudio 0.99.441 - Windows XP/Vista/7/8' installer link highlighted by a red box.

Left Window: RStudio Products Page

URL: <http://www.rstudio.com/products/R/>

RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian/Ubuntu, RedHat/CentOS, and SUSE Linux).

Desktop
Run RStudio on your desktop
[RStudio Desktop >](#)

Server
Centralize access and computation
[RStudio Server >](#)

RStudio Desktop

Open Source Edition

Right Window: RStudio Download Page

URL: <http://www.rstudio.com/products/rs>

Installers for Supported Platforms

Installers

RStudio 0.99.441 - Windows XP/Vista/7/8

RStudio 0.99.441 - Mac OS X 10.6+ (64-bit)

RStudio 0.99.441 - Ubuntu 12.04+/Debian 8+ (32-bit)

RStudio 0.99.441 - Ubuntu 12.04+/Debian 8+ (64-bit)

RStudio 0.99.441 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)

RStudio 0.99.441 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)

Zip/Tarballs

Zip/tar archives

RStudio 0.99.441 - Windows XP/Vista/7/8

RStudio 0.99.441 - Ubuntu 12.04+/Debian 8+ (32-bit)

RStudio 0.99.441 - Ubuntu 12.04+/Debian 8+ (64-bit)

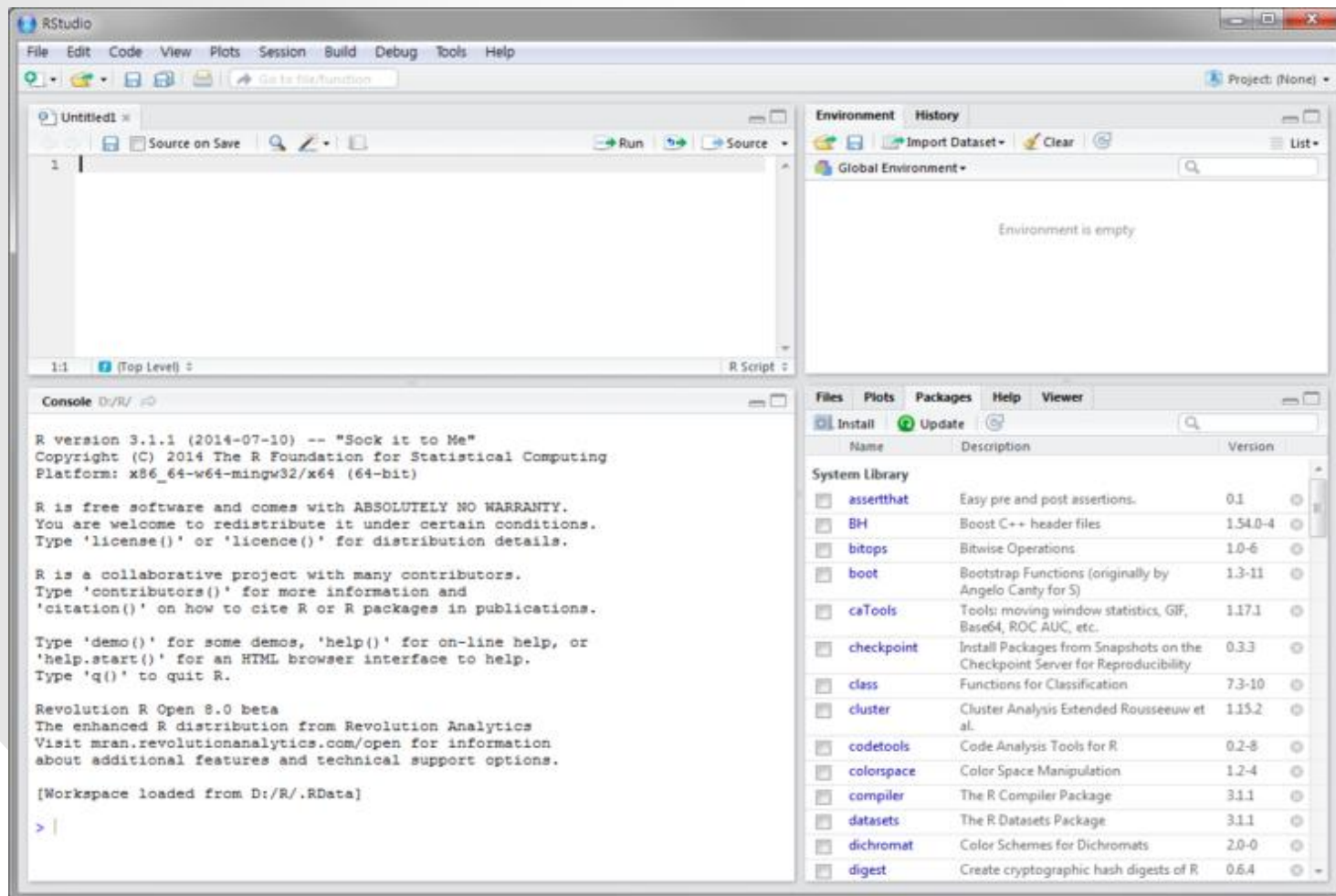
RStudio 0.99.441 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)

RStudio 0.99.441 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)



1. R 설치 및 개요

R Studio 실행 화면





1. R 설치 및 개요

● R Studio 4가지 패널 창 기능

R Script

```
87 # Nominal : level 순서의 값이 무의미 - 알파벳 순서 정렬
88 # Ordinal : level 순서의 값을 직접 정의 - 사용자가 순서 지정
89
90 # Factor Nominal 에
91 order <- c("third", "first", "second")
92 plot(order) # error
93
94 fa <- as.factor(order) # factor형으로 변환
95 plot(fa) # 막대차트 - 동일 문자열 수 카운터
96 fa
97 #####
98 # [1] third first second
99 # Levels: first second third <- 첫 글자 기준 오름차순 정렬
100 #####
101 # "stringsAsFactors = FALSE" <- Factor로 변경하지 말고, 문자 그대로 사용
102 <-
```

코딩 영역

작업내역/환경

Name	Type	Length	Size	Value
fa	fac	5	60	Factor w/ 5 levels
fr				
ga				
hflig...	tbl...	21	22...	227496 obs...
hong	num...	3	72...	num [1:3] ...
i	num...	1	48...	4
iris	dat...	5	6...	150 obs. ...
kang	num...	3	72...	num [1:3] ...
kor	ggm...	16...	12...	Large ggma...
kor.m...	gg	9	8...	List of 9
lat	num...	17	17...	num [1:17]...
lee	num...	3	72...	num [1:3] ...

콘솔(실행결과)

```
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project...
Type 'contributors()' for more information.
Type 'citation()' on how to cite R 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.
```

탐색기/그래프/도움말/패키지



1. R 설치 및 개요

- CRAN Site에서 제공하는 패키지 보기
`dim(available.packages())` # 9650 17 -> 6,650개
`available.packages()` # 패키지 상세보기
- R session 정보
 - ✓ Session : R 콘솔 시작 ~ 종료 전까지 기간 정보

`sessionInfo()` # 세션 정보 보기

`#R version 3.1.2 (2014-10-31)`

`#Platform: x86_64-w64-mingw32/x64 (64-bit)`

`#locale:`

`#attached base packages: 기본 패키지 - 7개`

`#[1] stats graphics`

`#[3] grDevices utils`

`#[5] datasets methods`

`#[7] base`

`# loaded via a namespace (and not attached): 로딩된 패키지`



1. R 설치 및 개요

- 변수(Variable)

- 1) 변수

- 값을 저장하는 메모리 이름
- R은 모두 변수가 객체(데이터+함수+차트 등)

- 2) 변수명 작성 규칙

- 영문자 시작
- 두번째 단어 : 숫자, 언더바(_), 점(member.id) 사용 가능
- 대소문자 구분
- 의미를 파악할 수 있는 이름으로 지정
- 정의된 변수는 재사용이 가능



1. R 설치 및 개요

- 명령어 실행 방법

- 1) Interaction 방식

- ✓ 줄 단위 실행
 - ✓ 줄 선택 + Ctrl + R 또는 Ctrl + Enter

- 2) Batch 방식

- ✓ 여러 줄 소스코드 일괄처리
 - ✓ Block 지정 + Ctrl + R

- Batch 방식 히스토그램 그리기 예

- ```
pdf("c:/Rwork/batch.pdf") # 지정된 경로의 파일에 결과 출력
```

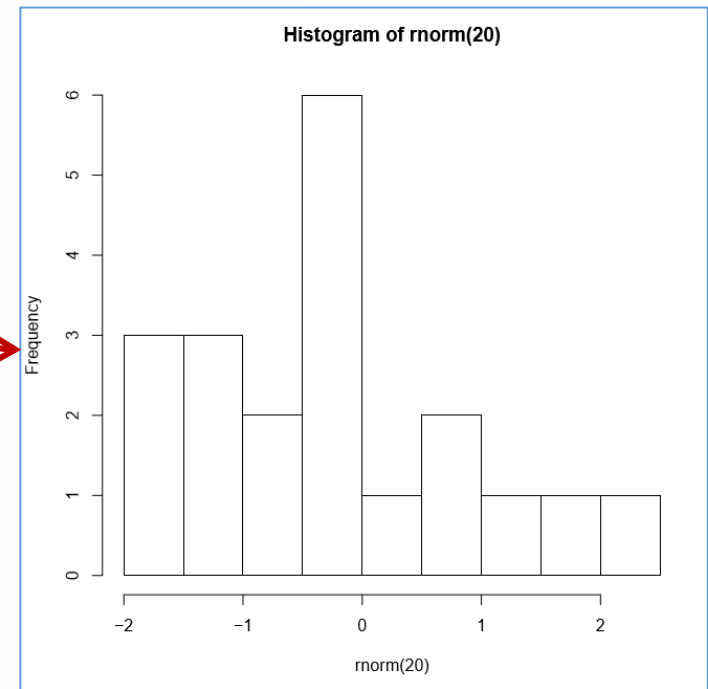
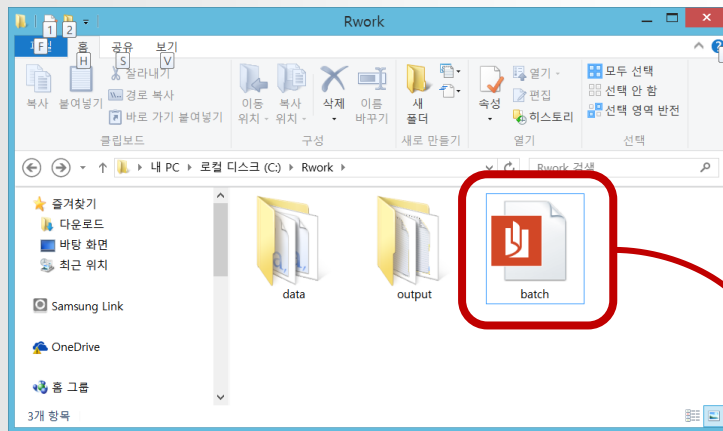
- ```
hist(rnorm(20)) # 난수에 대한 히스토그램 그리기
```

- ```
dev.off() # 출력할 파일 닫음
```



# 1. R 설치 및 개요

## ■ Batch 방식 히스토그램 결과





# 1. R 설치 및 개요

## ● 데이터 유형(Data Type)

✓ R은 변수 선언 시 별도의 Type을 선언하지 않음

| 유형(Type)       | 값(value) | 예                       |
|----------------|----------|-------------------------|
| 숫자형(Numeric)   | 정수, 실수   | 10, 25, 55.23           |
| 문자형(Character) | 문자, 문자열  | "홍길동"                   |
| 논리형(Logical)   | 참, 거짓    | TRUE 또는 T<br>FALSE 또는 F |
| 기타             | 결측치, 비슷자 | NA, NaN                 |





# 1. R 설치 및 개요

- 데이터 유형 보기

✓ mod()함수 : 데이터 유형 보기 함수

|                  |           |               |         |
|------------------|-----------|---------------|---------|
| is.numeric(x)    | 수치형 여부    | is.integer(x) | 정수형 여부  |
| is.logical(x)    | 논리형 여부    | is.double(x)  | 실수형 여부  |
| is.character(x)  | 문자형 여부    | is.complex(x) | 복소수형 여부 |
| is.data.frame(x) | 데이터프레임 여부 | is.factor(x)  | 범주형 여부  |
| is.na(x)         | NA 여부     | is.nan(x)     | NaN 여부  |



# 1. R 설치 및 개요

- 데이터 타입 보기 함수
  - ✓ `is.character(string) # TRUE`
  - ✓ `x <- is.numeric(int)`
  - ✓ `x # TRUE`
  - ✓ `is.logical(boolean) # TRUE`
  - ✓ `is.logical(x) # TRUE -> TRUE`
  - ✓ `is.na(x) # NA 확인`
- 현재 사용중인 변수객체 보기
  - ✓ `ls()`

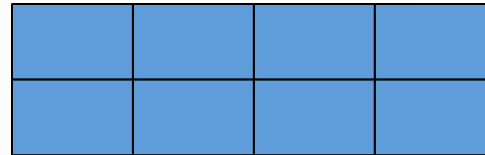


# 1. R 설치 및 개요

- 자료형과 자료구조(객체 타입) 보기
  - ✓ mode(변수) : 자료형(Data Type) 보기
  - ✓ class(변수) : 자료구조(객체) 보기
- mode() vs class()
  - ✓ ex) 변수 <- 데이터
  - ✓ mode : 데이터 성격(숫자형, 문자형, 논리형)
  - ✓ class : 데이터 자료구조 성격(Array, List, Table 등)



Array 자료구조

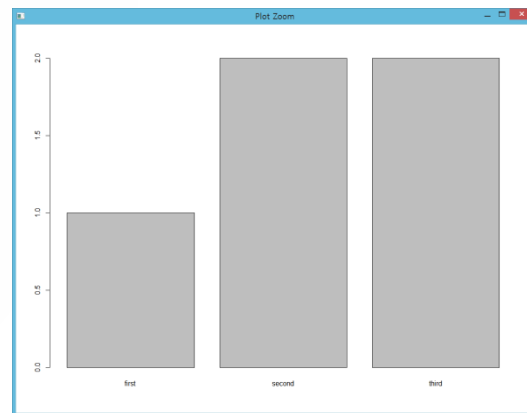


Matrix 자료구조



# 1. R 설치 및 개요

- Factor : 범주(값의 목록)를 갖는 자료구조
  - ✓ Nominal : level 순서의 값 무의미-알파벳 순서 정렬
  - ✓ Ordinal : level 순서의 값 의미 - 사용자가 직접 순서 지정
- ✓ # Factor Nominal 예
- ✓ `order <- c("third", "second", "first", "third", "second")`
- ✓ `plot(order)` # error
- ✓ `fa <- as.factor(order)` # factor형으로 변환
- ✓ `plot(fa)` # 막대차트 - 동일문자열 수 카운터





# 1. R 설치 및 개요

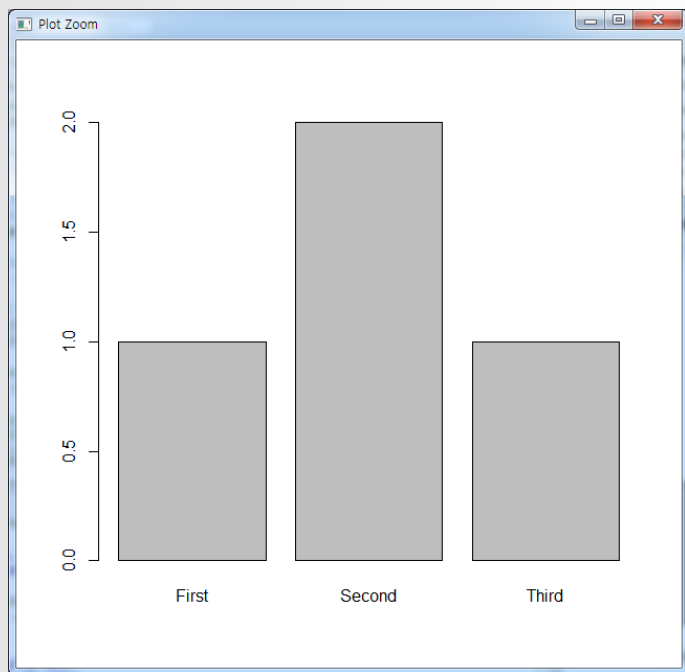
## ● 데이터 형 변환 함수

|                               |           |                            |         |
|-------------------------------|-----------|----------------------------|---------|
| <code>as.numeric(x)</code>    | 수치형 변환    | <code>as.integer(x)</code> | 정수형 변환  |
| <code>as.logical(x)</code>    | 논리형 변환    | <code>as.double(x)</code>  | 실수형 변환  |
| <code>as.character(x)</code>  | 문자형 변환    | <code>as.complex(x)</code> | 복소수형 변환 |
| <code>as.data.frame(x)</code> | 데이터프레임 변환 | <code>as.factor(x)</code>  | 범주형 변환  |
| <code>as.list(x)</code>       | 리스트형 변환   | <code>as.vector(x)</code>  | 벡터형 변환  |

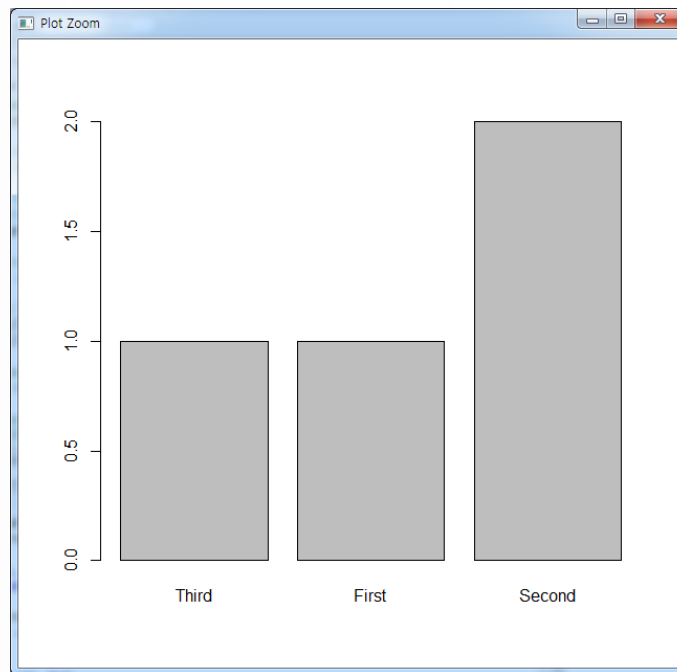


# 1. R 설치 및 개요

**plot(level) # Nominal인 경우**



**plot(level2) # Ordinal인 경우**





# 1. R 설치 및 개요

- 패키지(Package)

- ✓ 패키지 개념 : 데이터+기능(함수)+알고리즘 꾸러미(압축)
- ✓ R 실력은 패키지 활용 능력

- 패키지 설치

형식) `install.packages("패키지명")`

- ✓ `install.packages("plyr")`

- 패키지 설치 위치

- ✓ `C:\Users\jinsung\Documents\RWwin-library\3.1`
- ✓ 관련 패키지 : 특정 패키지 설치하면 관련 패키지도 함께 설치됨
- ✓ 기본 설치 패키지 : R을 설치하면 기본으로 설치되는 패키지  
예) `base`, `stats`





# 1. R 설치 및 개요

- 패키지 설치 확인

- ✓ `installed.packages()` # 현재 시스템에 설치된 전체 패키지 보임
- ✓ 패키지명                      버전
- ✓ `plyr`                              "3.1.3"

- 패키지 사용 : R은 in Memory -> 메모리 탑재해야 사용가능

형식) `library(패키지명)`

✓ `library(plyr)`

✓ `require(plyr)`

- 현재 로드 된 패키지 확인

✓ `search()`

- 패키지 제거 : 설치 폴더 제거 or `remove.packages("패키지명")`

✓ `remove.packages("plyr")`



# 1. R 설치 및 개요

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function

Project: (None)

Environment History

Global Environment

boolean TRUE

Bug\_Metri... xtabs [1:5, 1:5, 1:2] 1160...

fa Factor w/ 3 levels "first"...

int 20

level1 Factor w/ 3 levels "First"...

level2 ord.factor w/ 3 levels "Th..."

order chr [1:5] "third" "second"...

Severity\_... Named num [1:10] 11605 374...

status chr [1:4] "second" "First"...

string "홍길동"

x TRUE

Files Plots Packages Help Viewer

Install Update

| Name                                | Description                                    | Ver...   |
|-------------------------------------|------------------------------------------------|----------|
| <b>User Library</b>                 |                                                |          |
| <input type="checkbox"/> assertthat | Easy pre and post assertions.                  | 0.1      |
| <input type="checkbox"/> BH         | Boost C++ Header Files                         | 1.58.0-1 |
| <input type="checkbox"/> colorspace | Color Space Manipulation                       | 1.2-6    |
| <input type="checkbox"/> DBI        | R Database Interface                           | 0.3.1    |
| <input type="checkbox"/> dichromat  | Color Schemes for Dichromats                   | 2.0-0    |
| <input type="checkbox"/> digest     | Create Cryptographic Hash Digests of R Objects | 0.6.8    |
| <input type="checkbox"/> dplyr      | A Grammar of Data Manipulation                 | 0.4.2    |
| <input type="checkbox"/> ggplot2    | An Implementation of the Grammar of Graphics   | 1.0.1    |
| <input type="checkbox"/> gtable     | Arrange grobs in tables.                       | 0.1.2    |
| <input type="checkbox"/> hflights   | Flights that departed                          | 0.1      |

```
177 # + 버튼 클릭 -> R 스크립트
178
179
180 # 2. 패키지 사용
181 # - 패키지 개념 : 데이터+기능(함수)+알고리즘 꾸러미(압축)
182 # - R 실력은 패키지 활용 능력이다.
183
184 # - 패키지 설치 : 형식) install.packages("패키지명")
185 install.packages("plyr")
186 # 패키지 설치 위치 : C:\Users\jinsung\Documents\R\win-library\3.1
187 # - 관련 패키지 : 특정 패키지(Rcpp)를 설치하면 관련 패키지도 함께 설치
188 # - 기본 설치 패키지 : R을 설치하면 기본으로 설치되는 패키지
189 # -> base, stats
190
191 # - 패키지 설치 확인
192 installed.packages() # 현재 시스템에 설치된 전체 패키지 보임
193 # 패키지명 버전
194 # plyr "3.1.3"
195
196 # -> 오른쪽 하단의 Packages 탭에서 확인
197
198 # - 패키지 사용 : R은 in Memory이다. -> 메모리 할재해야 사용가능
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Console

```
ordered = is.ordered(x), nmax = NA)
NULL
> installed.packages() # 현재 시스템에 설치된 전체 패키지 보임
```

| Package    | LibPath                                      |
|------------|----------------------------------------------|
| assertthat | "C:/Users/acorn/Documents/R/win-library/3.2" |
| BH         | "C:/Users/acorn/Documents/R/win-library/3.2" |
| colorspace | "C:/Users/acorn/Documents/R/win-library/3.2" |
| DBI        | "C:/Users/acorn/Documents/R/win-library/3.2" |
| dichromat  | "C:/Users/acorn/Documents/R/win-library/3.2" |
| digest     | "C:/Users/acorn/Documents/R/win-library/3.2" |
| dplyr      | "C:/Users/acorn/Documents/R/win-library/3.2" |
| ggplot2    | "C:/Users/acorn/Documents/R/win-library/3.2" |
| gtable     | "C:/Users/acorn/Documents/R/win-library/3.2" |
| hflights   | "C:/Users/acorn/Documents/R/win-library/3.2" |



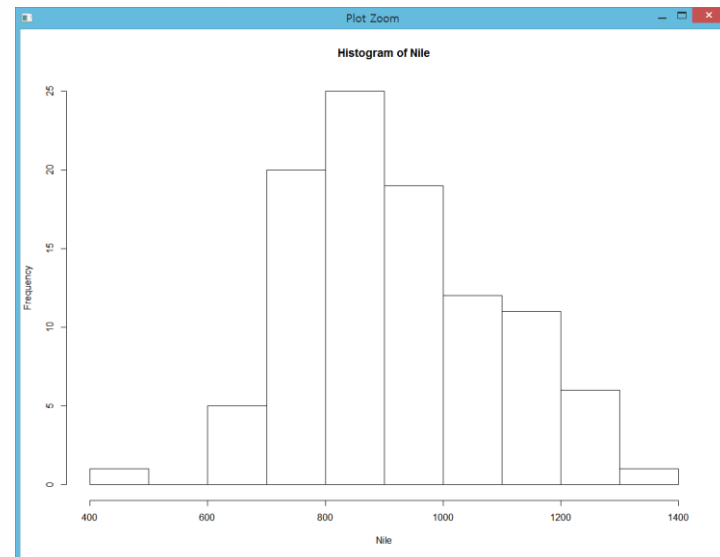
# 1. R 설치 및 개요

## ● 기본 함수 사용

- ✓ 함수 도움말 : `help(함수명)`, `?함수명`
- ✓ `help(mean)` # 브라우저 화면으로 제공
- ✓ `?sum`
- ✓ google에서 검색 : 함수() in r

## ● 데이터 세트와 리스트 보기

- ✓ `data()`
- ✓ `hist(Nile)` # 나일강 흐름에 대한 데이터 셋 히스토그램 그리기





# 1. R 설치 및 개요

data()

```
Data sets in package ;@arules;`:

Adult Adult Data Set
AdultUCI Adult Data Set
Epub Epub Data Set
Groceries Groceries Data Set
Income Income Data Set
IncomeESL Income Data Set

Data sets in package ;@datasets;`:

AirPassengers Monthly Airline Passenger Numbers 1949-1960
BJsales Sales Data with Leading Indicator
BJsales.lead (BJsales) Sales Data with Leading Indicator

BOD Biochemical Oxygen Demand
CO2 Carbon Dioxide Uptake in Grass Plants
ChickWeight Weight versus age of chicks on different diets

DNase Elisa assay of DNase
EuStockMarkets Daily Closing Prices of Major European Stock Indices, 1991-1998

Formaldehyde Determination of Formaldehyde
HairEyeColor Hair and Eye Color of Statistics Students
Harman23.cor Harman Example 2.3
Harman74.cor Harman Example 7.4
Indometh Pharmacokinetics of Indomethacin
InsectSprays Effectiveness of Insect Sprays
JohnsonJohnson Quarterly Earnings per Johnson & Johnson Share

LakeHuron Level of Lake Huron 1875-1972
LifeCycleSavings Intercountry Life-Cycle Savings Data
Loblolly Growth of Loblolly pine trees
Nile Flow of the River Nile
Orange Growth of Orange Trees
OrchardSprays Potency of Orchard Sprays
PlantGrowth Results from an Experiment on Plant Growth
Puromycin Reaction Velocity of an Enzymatic Reaction
Seatbelts Road Casualties in Great Britain 1969-84
Theoph Pharmacokinetics of Theophylline
Titanic Survival of passengers on the Titanic
ToothGrowth The Effect of Vitamin C on Tooth Growth in Guinea Pigs

UCBAdmissions Student Admissions at UC Berkeley
UKDriverDeaths Road Casualties in Great Britain 1969-84
```

R 제공  
데이터 셋



# 1. R 설치 및 개요

- 함수 파라미터 보기

형식) `args(함수명)`

✓ `args(max) #`

✓ `# max() 함수 파라미터 : (na.rm = FALSE) <- NA 제거 안됨`

✓ `max(10,20,NA,30) # NA 출력`

- 함수 예제 보기

형식) `example(함수명)` : 해당 함수의 사용 예 제공

✓ `example(seq)`

✓ `seq(10,20, by=2) # 10 12 14 16 18 20`

✓ `example(mean)`

✓ `mean(10:20) # 15`



# 1. R 설치 및 개요

- 작업공간 보기

- ✓ `getwd()` # 기본함수

- 작업공간 지정

- ✓ `setwd("C:/Rwork/Part-I")` # 구분자 : / or \\

- ✓ `data <- read.csv("test.csv", header=T)` # 지정된 경로에서 파일 로딩

- ✓ `args(read.csv)` # (file, header = TRUE)

- ✓ data

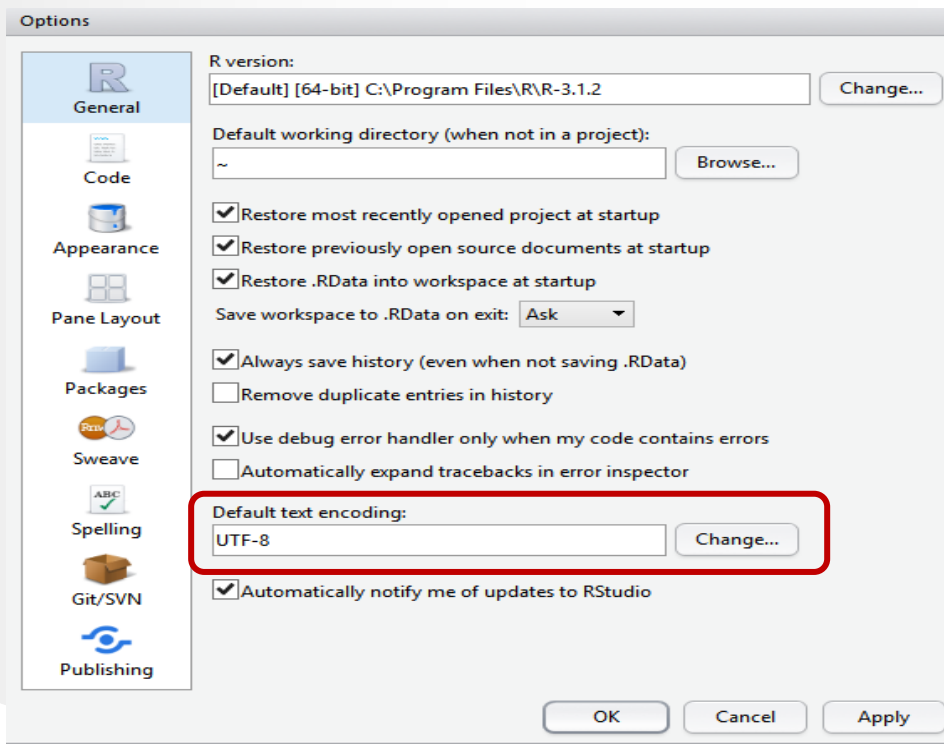
```
> data
 A B C D E
1 2 4 4 2 2
2 1 2 2 2 2
3 2 3 4 3 3
4 3 5 5 3 3
5 3 2 4 4 4
6 4 3 3 4 2
```

test.csv  
파일 내용



# 1. R 설치 및 개요

- 스크립트 파일 저장 및 깨짐 현상 처리 방법
  - ✓ 문자 인코딩 방식 지정 : UTF-8
  - ✓ 메뉴 : Tools > General Options > General Default text encoding-UTF-8
  - ✓ 메뉴 : File > Save with Encoding > UTF-8







## 2. 데이터 유형과 구조

### Chap02\_DataStructure 수업내용

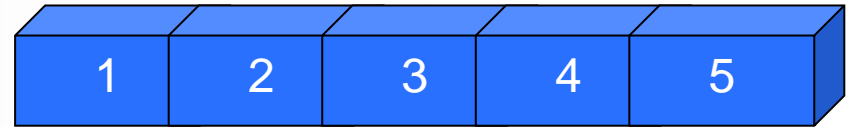
- R에서 제공하는 주요 자료구조(객체 타입)
  1. Vector : 동일 데이터 타입을 갖는 1차원 배열
  2. Matrix : 동일 데이터 타입을 갖는 2차원 배열
  3. Data Frame : 열 단위로 서로 다른 데이터 타입을 갖는 배열
    - > 2차원 테이블 구조(DB 테이블과 유사)
  4. Array : 동일 데이터 타입을 갖는 다차원 배열
  5. List : 서로 다른 데이터 구조(Vector, Data Frame, Array, List) 중첩



## 2. 데이터 유형과 구조

### 1) Vector 자료구조

- ✓ R의 기본 데이터 구조
- ✓ 1차원 배열 형태
  - 접근 : [index] : 1부터 시작
- ✓ 동일한 타입의 데이터만 저장 가능
- ✓ 벡터 데이터 생성 함수 : `c()`, `seq()`, `rep()`
- ✓ 벡터 데이터 처리 함수 : `setdiff()`, `intersect()`



Vector



## 2. 데이터 유형과 구조

### 2) Matrix 자료구조

- ✓ 동일 데이터 타입을 갖는 2차원 배열
- ✓ 행렬(matrix) 객체
- ✓ Matrix 데이터 생성 함수
  - `rbind()` : 행 묶음
  - `cbind()` : 컬럼 묶음
- ✓ Matrix 데이터 처리 함수
  - `apply()` : 함수적용

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

Vector

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |

Matrix



## 2. 데이터 유형과 구조

- c()함수 이용 matrix 생성

```
m <- matrix(c(1:5))
m # 열 기준으로 행렬 생성

 [,1]
[1,] 1
[2,] 2
[3,] 3
[4,] 4
[5,] 5
#####
```

```
m <- matrix(c(1:10), nrow=2) # 열 우선 2행2열 생성
m

 [,1] [,2] [,3] [,4] [,5]
[1,] 1 3 5 7 9
[2,] 2 4 6 8 10

m <- matrix(c(1:10), nrow=2, byrow=T) # by=T : 행 우선
m

 [,1] [,2] [,3] [,4] [,5]
[1,] 1 2 3 4 5
[2,] 6 7 8 9 10
#####
```



## 2. 데이터 유형과 구조

3) Array : 동일 데이터 타입을 갖는 다차원 배열

- ✓ 3차원 배열 객체 생성
- ✓ R에서 활용도 낮음

```
d <- c(1:12) # 12개 벡터 객체 생성
arr <- array(d, c(3,2,2)) # 3행2열 구조 2개
arr #1~6(1면), 7~12(2면) -> 3차원 배열 객체
#####
```

```
,, 1
```

```
[,1] [,2]
[1,] 1 4
[2,] 2 5
[3,] 3 6
```

```
,, 2
```

```
[,1] [,2]
[1,] 7 10
[2,] 8 11
[3,] 9 12
```

```
#####
```

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|

Vector

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |

Matrix

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |

Array



## 2. 데이터 유형과 구조

### 4) List : 서로 다른 데이터 구조

- ✓ Vector, Data Frame, Array, List의 중첩 구조
- ✓ c(구조체), python(딕셔너리)
- ✓ 함수 내에서 여러 값을 하나의 키로 묶어서 반환할 경우 유용함

```
member <- list(name="홍길동",
 age = 35,address="한양",
 gender="남자", htype="아파트")
```

| Key     | Value |
|---------|-------|
| name    | 홍길동   |
| age     | 35    |
| address | 한양시   |
| gender  | 남자    |
| htype   | 아파트   |



## 2. 데이터 유형과 구조

### 5) Data Frame

- ✓ 리스트 보다 활용범위 넓다.
- ✓ DBMS의 테이블 구조와 유사  
(서로 다른 데이터 타입을 갖는 컬럼)
- ✓ 가장 많이 사용하는 객체 타입
- ✓ list와 Vector 혼합형
  - 컬럼 구성 : list, list 구성 : vector
- ✓ data frame 생성방법
  - Vector, Matrix, txt/excel/csv 파일



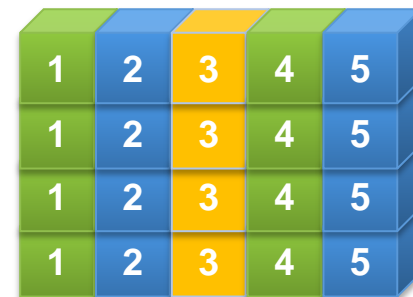
Vector



Array



Matrix



Data Frame





## 2. 데이터 유형과 구조

- Data Frame 특징

1. 형식) data.frame(컬럼1, 컬럼2.. 컬럼n)
2. 컬럼 단위로 서로 다른 자료형 가능
3. 모든 컬럼은 크기가 동일해야 함

### [ 컬럼 구성 예 ]

```
id <- c("hong", "lee", "kang")
```

```
name <- c("홍길동", "이순신", "강감찬")
```

```
age <- c(30, 35, 45)
```



## 2. 데이터 유형과 구조

- Data Frame 생성

- 1) Vector|용 객체 생성

```
no <- c(1,2,3)
```

```
name <- c("hong", "lee", "kim")
```

```
pay <- c(150,250,300)
```

```
vemp <- data.frame(NO=no, Name=name, Pay=pay) #컬럼명 지정
```

```
vemp
```

```
#####
```

```
 NO Name Pay
```

```
1 1 hong 150
```

```
2 2 lee 250
```

```
3 3 kim 300
```

```
#####
```



## 2. 데이터 유형과 구조

- Data Frame 생성

2) matrix이용 객체 생성

```
m <- matrix(
 c(1,"hong",150,
 2, "lee", 250,
 3, "kim", 300) ,3 ,by=T) # 행우선, 3개 리스트 생성
memp <- data.frame(m)
memp
#####
 X1 X2 X3 <- 기본 컬럼명
1 1 hong 150
2 2 lee 250
3 3 kim 300
#####
```



## 2. 데이터 유형과 구조

- Data Frame 생성

- 3) txt파일 이용 객체 생성

```
getwd()
```

```
setwd("c:/Rwork/Part-I")
```

```
txttemp <- read.table('emp.txt', header=T, sep="")
```

```
제목있음, 공백구분
```

```
txttemp
```

```
#####
```

```
사번 이름 급여
```

```
1 101 hong 150
```

```
2 201 lee 250
```

```
3 301 kim 300
```

```
#####
```



## 2. 데이터 유형과 구조

- Data Frame 생성

- 4) csv파일 이용 객체 생성

```
getwd()
```

```
csvtemp <- read.csv('emp.csv', header=T) # 제목있음,
```

```
컴마구분
```

```
csvtemp
```

```
#####
```

```
사번 이름 급여
```

```
1 101 홍길동 150
```

```
2 102 이순신 450
```

```
3 103 강감찬 500
```

```
4 104 유관순 350
```

```
5 105 김유신 400
```

```
#####
```



## 2. 데이터 유형과 구조

```
데이터 프레임 검색 -> 벡터 결과 반환
```

```
df$x # 형식)변수$컬럼
```

```
데이터 프레임 처리함수
```

```
str(df) # 테이블 구조보기
```

```
summary(df) # 요약함수
```

```
#####
```

|          | x  | y          | z   |
|----------|----|------------|-----|
| Min.     | :1 | Min. : 2   | a:1 |
| 1st Qu.: | 2  | 1st Qu.: 4 | b:1 |
| Median   | :3 | Median : 6 | c:1 |
| Mean     | :3 | Mean : 6   | d:1 |
| 3rd Qu.: | 4  | 3rd Qu.: 8 | e:1 |
| Max.     | :5 | Max. :10   |     |

```
#####
```



## 2. 데이터 유형과 구조

```
apply(df[, c(1,2)],2, sum) # 컬럼(열)단위 합계
```

```
x y
```

```
15 30
```

```
데이터프레임 대상 조건에 만족하는 서브셋 만들기
```

```
x1 <- subset(df, x>=3) # x가 3이상인 레코드 대상 서브셋 생성
```

```
x1
```

```
y1 <- subset(df, y<=8) # y가 8이하인 레코드 대상 서브셋 생성
```

```
y1
```

```
xy <- subset(df, x>=2 & y<=6) # 2개 조건이 참인 레코드 대상 서브셋 생성
```

```
xy
```



## 2. 데이터 유형과 구조

### ● Data Join

```
h <- data.frame(id=c(1,2), h=c(180,175))
w <- data.frame(id=c(1,2), w=c(80,75))

id 컬럼으로 data.frame 조인
user3 <- merge(h, w, by.x="id", by.y="id")
user3
id h w
1 1 180 80
2 2 175 75
```





## 2. 데이터 유형과 구조

- stringr()과 정규표현식

```
install.packages("stringr") # 패키지 설치
```

```
library(stringr) # in memory
```

```
str_extract("abcd12aaa33", "[1-9]{2}") # 연속된 숫자2개가 첫번째 발견 항목
```

```
str_extract_all("abcd12aaa33", "[1-9]{2}") # 모두
```

```
d <- c("김길동","유관순","강감찬","김길동")
```

```
str_replace(d, "김길동","홍길동") # 문자열 교체
```

```
subs <- str_sub("abcd12aaa33", 3,6) # 서브스트링 생성
```

```
subs # "cd12"
```



## 2. 데이터 유형과 구조

- Stringr 패키지에서 제공하는 주요 함수

**str\_length ()** : 문자열 길이 리턴

**str\_join** : 문자열 연결

**str\_sub()** : 범위에 해당하는 부분 문자열 출력

**str\_split()** : 기준문자를 중심으로 부분 문자열 리스트 출력

**str\_replace** : 문자열 교체

**str\_extract()** : 문자열의 위치(index) 리턴

**str\_locate()** : 문자열에서 특정 문자열 패턴의 첫번째 위치 찾기

**str\_locate\_all()** : 문자열에서 특정 문자열 패턴의 전체 위치 찾기

✓ 문자열 패턴은 정규표현식(Regular Expression) 이용

✓ 참고 사이트 : 위키백과 정규표현식



## 2. 데이터 유형과 구조

### ● 위키백과 정규 표현식

[http://ko.wikipedia.org/wiki/%EC%A0%95%EA%B7%9C\\_%ED%91%9C%ED%98%84%EC%8B%9D](http://ko.wikipedia.org/wiki/%EC%A0%95%EA%B7%9C_%ED%91%9C%ED%98%84%EC%8B%9D)

| 식      | 기능          | 설명                                                                                                                                                                      |
|--------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .      | 문자          | 1개의 문자와 일치한다. 단일행 모드에서는 <b>새줄 문자</b> 를 제외한다.                                                                                                                            |
| \w     | 이스케이프       | 특수 문자를 식에 문자 자체로 포함한다.                                                                                                                                                  |
|        | 선택          | 여러 식 중에서 하나를 선택한다. 예를 들어, "abc adc"는 abc와 adc 문자열을 모두 포함한다.                                                                                                             |
| ^      | 부정          | 문자 클래스 안의 문자를 제외한 나머지를 선택한다. 예를 들면 [^abc]d는 ad, bd, cd는 포함하지 않고 ed, fd 등을 포함한다. [^a-z]는 알파벳 소문자로 시작하지 않는 모든 문자를 의미한다.                                                   |
| []     | 문자 클래스      | "["과 "]" 사이의 문자 중 하나를 선택한다. " "를 여러 개 쓴 것과 같은 의미이다. 예를 들면 [abc]d는 ad, bd, cd를 뜻한다. 또한, "-" 기호와 함께 쓰면 범위를 지정할 수 있다. "[a-z]"는 a부터 z까지 중 하나, "[1-9]"는 1부터 9까지 중의 하나를 의미한다. |
| ()     | 하위식         | 여러 식을 하나로 묶을 수 있다. "abc adc"와 "a(b d)c"는 같은 의미를 가진다.                                                                                                                    |
| *      | 0회 이상       | 0개 이상의 문자를 포함한다. "a*b"는 "b", "ab", "aab", "aaab"를 포함한다.                                                                                                                 |
| +      | 1회 이상       | "a+b"는 "ab", "aab", "aaab"를 포함하지만 "b"는 포함하지 않는다.                                                                                                                        |
| ?      | 0 또는 1회     | "a?b"는 "b", "ab"를 포함한다.                                                                                                                                                 |
| {m}    | m회          | "a{3}b"는 "aaab"만 포함한다.                                                                                                                                                  |
| {m,}   | m회 이상       | "a{2,}b"는 "aab", "aaab", "aaaab"를 포함한다. "ab"는 포함되지 않는다.                                                                                                                 |
| {m, n} | m회 이상 n회 이하 | "a{1,3}b"는 "ab", "aab", "aaab"를 포함하지만, "b"나 "aaaab"는 포함하지 않는다.                                                                                                          |



# 3. 데이터 입출력

## Chap03\_DataIO 수업내용

- 데이터 불러오기
  - (1) 키보드 입력
  - (2) 파일 데이터 가져오기
  - (3) 웹 문서 가져오기
- 데이터 저장하기
  - (1) 화면(콘솔) 출력
  - (2) 파일에 데이터 저장



### 3. 데이터 입출력

- 데이터 불러오기

- (1) 키보드 입력

- 1) scan() 함수를 이용

- # 숫자입력

- ```
x <- scan()
```

- ```
1: 1
```

- ```
2: 2
```

- ```
3: 3
```

- ```
4: 4
```

- ```
5: 5
```

- ```
6:
```

- Read 5 items

- ```
x # [1] 1 2 3 4 5
```



### 3. 데이터 입출력

```
문자입력
y <- scan(what="")
1: a
2: b
3: c
4: d
5: e
6:
 Read 5 items
y # [1] "a" "b" "c" "d" "e"
```



### 3. 데이터 입출력

#### 2) edit() 함수를 이용한 입력

```
df <- data.frame() #빈 데이터프레임 생성
exam = edit(df) # 데이터 편집기
exam
```

```
104 # 데이터 프레임에서 사용
105
106 df <- data.frame() #빈 데이터프레임 생성
107 exam = edit(df) # 성적 데이터 입력
108
109 # 성적 데이터
110 학번 성명 국어 수학 영어
111 1 홍길동 80 87 72
112 2 이순신 90 85 87
113 3 강강찬 95 84 79
114 4 유관순 94 85 77
115 5 김유신 85 88 79
116 -----
117 > which(성적$성명=="은미")
118 [1] 6
119 [출처] 10. 조건문, 반복문 (R 프로그래밍)
```

| 파일 | 편집 | 도움말 |    |    |      |  |
|----|----|-----|----|----|------|--|
| 학번 | 성명 | 국어  | 영어 | 수학 | var6 |  |
| 1  | 1  | 홍길동 | 80 | 87 | 72   |  |
| 2  | 2  | 이순신 | 90 | 85 | 87   |  |
| 3  | 3  | 강강찬 | 95 | 84 | 79   |  |
| 4  | 4  | 유관순 | 94 | 85 | 77   |  |
| 5  | 5  | 김유신 | 85 | 88 | 79   |  |
| 6  |    |     |    |    |      |  |
| 7  |    |     |    |    |      |  |
| 8  |    |     |    |    |      |  |
| 9  |    |     |    |    |      |  |
| 10 |    |     |    |    |      |  |
| 11 |    |     |    |    |      |  |
| 12 |    |     |    |    |      |  |
| 13 |    |     |    |    |      |  |
| 14 |    |     |    |    |      |  |
| 15 |    |     |    |    |      |  |



### 3. 데이터 입출력

#### (2) 파일 데이터 가져오기

##### 1) read.table() 함수 이용

- 각 값들은 빈칸, 탭, 콜론(:), 세미콜론(;), 콤마(,) 등으로 구분
- 구분자가 공백이나 탭인 경우가 sep 의 기본값, 'header=F' 기본값

##### ① 컬럼명이 없는 파일 불러오기

```
setwd("C:/Rwork/Part-I") # setwd("경로명")
student <- read.table(file="student.txt")
student # 컬럼명이 없으면 V1, V2, V3, V4 형태로 기본 이름 지정
V1 V2 V3 V4
#1 101 hong 175 65
#2 201 lee 185 85
#3 301 kim 173 60
#4 401 park 180 70
names(student) <- c("번호","이름","키","몸무게")# 컬럼명 변경
student
```





### 3. 데이터 입출력

#### ② 컬럼명이 있는 파일 불러오기

```
student1 <- read.table(file="student1.txt", header=TRUE)
```

```
#student1 <-read.table(file.choose(), header=TRUE) # 파일열기대화상자
```

```
student1
```



### 3. 데이터 입출력

(3) 구분자가 있는 경우(세미콜론, 탭)

```
student2 <- read.table(file="student2.txt", sep=";", header=TRUE)
```

# 세미콜론 구분자

```
student2
```

```
#student2 <- read.table(file="student2.txt", sep="\t", header=TRUE)
```

# 탭키 구분자



### 3. 데이터 입출력

(4) 특정문자 NA 처리(- 문자열을 NA로 처리)

```
read.table(file="student3.txt", sep=" ", header=TRUE, na.strings="-")
```

# - 문자를 NA 처리



### 3. 데이터 입출력

#### 2) read.csv() 함수 이용

- 구분자 ","가 sep의 기본값이며, header=TRUE가 기본값

```
student4 <- read.csv(file="student4.txt", sep=";", na.strings="-")
```

```
student4
```

```
student4 <- read.csv(file.choose(), sep=";", na.strings="-") # 파일열기
```



### 3. 데이터 입출력

3) read.xlsx() 함수를 이용해서 엑셀데이터 읽어오기

# xlsx 패키지 설치

```
install.packages("xlsx")
```

```
install.packages("rJava")
```

# java의 설치경로를 지정한다.

```
Sys.setenv(JAVA_HOME='C:\\Program Files\\Java\\jre1.8.0_31')
```

```
library(rJava) # 로딩
```

# xlsx 패키지 로드 시 주의사항

```
library(xlsx) # rJava를 로드하기 때문에 rJava 패키지 설치 필요
```

```
studenex <- read.xlsx(file.choose(), sheetIndex=1, encoding="UTF-8")
```

#한글인코딩, file="C:/Rwork/Part-I/studentexcel.xlsx"

```
studenex
```



### 3. 데이터 입출력

#### 4) 웹문서 가져오기

```
install.packages("XML")
```

```
library(XML)
```

```
주소확인 : http://www.infoplease.com/ipa/A0104652.html
```

```
미국의 각 주별 1인당 소득자료
```

```
info.url <- "http://www.infoplease.com/ipa/A0104652.html"
```

```
readHTMLTable() 함수 역할 - <table>, <tr>, <td> 태그 이용
```

```
info.df <- readHTMLTable(info.url, header=T, which=1, stringsAsFactors=F)
```

```
header=T : 컬럼명 있음, which=1 : 첫번째
```

```
stringsAsFactors 문자는 범주(값의 목록)처리 안함
```

```
info.df
```

```
str(info.df) # ipContentTable:'data.frame': 58 obs. of 12 variables:
```

```
dim(info.df) #[1] 54 12
```

```
names(info.df) # table 제목 출력
```



### 3. 데이터 입출력

- 데이터 저장하기

- (1) 화면(콘솔) 출력

- 1) cat() 함수

- ```
x <- 10
```

- ```
y <- 20
```

- ```
z <- x * y
```

- ```
cat("x*y의 결과는 ", z, " 입니다.\n") # \n 줄바꿈
```

- ```
cat("x*y = ", z)
```

- 2) print() 함수

- ```
print(z) # 변수 또는 수식만 가능
```



### 3. 데이터 입출력

#### (2) 파일에 데이터 저장

##### 1) sink() 함수를 이용 파일 저장

`setwd("C:/Rwork/output")` # 현재 작업디렉토리 확인/폴더 생성

`sink("savework.txt")` # 저장할 파일명

# 파일 선택 창에서 `studentexcel.xlsx` 파일 선택

`studentx <- read.xlsx(file.choose(), sheetIndex=1, encoding="UTF-8")`

`studentx` #출력되는 값이 화면에 나타나지 않고 파일에 저장됨

`sink()` # 해제





### 3. 데이터 입출력

2) write.table()함수 이용 파일 저장

getwd()

① 기본옵션으로 저장 - 행 이름과 따옴표 붙음

```
write.table(studentx, "stdt.txt") # 행 번호와 따옴표 출력
```

② 행 이름 제거하여 저장

```
write.table(studentx, "stdt2.txt", row.names=FALSE) # 행번호 제거
```

③ 따옴표 제거하여 저장

```
write.table(studentx, "stdt3.txt", row.names=FALSE, quote=FALSE)
```

```
행 이름과 따옴표 제거
```



## 3. 데이터 입출력

### 3) write.xlsx() 함수 이용 파일 저장

- 엑셀 파일로 데이터 저장 함수

```
library(xlsx) # excel data 입출력 함수 제공
```

```
studentexcel.xlsx 파일 선택
```

```
st.df <- read.xlsx(file.choose(), sheetIndex=1, encoding="UTF-8")
```

```
str(st.df) # data.frame
```

```
write.xlsx(st.df, "studentx.xlsx") # excel형식으로 저장
```



### 3. 데이터 입출력

#### 4) write.csv() 함수 이용 파일 저장

# data.frame 형식의 데이터를 csv 형식으로 저장

```
setwd("C:/Rwork/Part-I")
```

```
st.df
```

```
write.csv(st.df,"stdf.csv", row.names=F, quote=F) # 행 이름 제거
```



# 4. 제어문과 함수

## Chap04\_1\_Control 수업내용

- 조건문
  - ✓ if(), ifelse(), switch()
- 반복문
  - ✓ for(), while(), repeat()



## 4. 제어문과 함수

### 1) if()함수

```
if(x*y > 40){
 cat("x*y의 결과는 40 이상입니다.\n") # \n 줄바꿈
 cat("x*y =", z)
}else{
 cat("x*y의 결과는 40 미만입니다. x*y =", z, "\n")
}
```



## 4. 제어문과 함수

2) **ifelse(조건, 참, 거짓) - 3항 연산자 기능**

```
ifelse(score>=80, "우수","노력") #우수
```

```
ifelse(score<=80, "우수","노력") #노력
```



## 4. 제어문과 함수

### 3) switch 문

형식) **switch**(비교 구문, 실행구문1, 실행구문2, 실행구문3)

```
switch("name", age=105, name="홍길동", id="hong",
 pwd="1234")
```

### 4) which 문

형식) **which()**의 괄호내의 조건에 해당하는 위치(인덱스)를  
출력한다.

# 벡터에서 사용

```
name <- c("kim","lee","choi","park")
```

```
which(name=="choi") # [1] 3
```



## 4. 제어문과 함수

# 데이터프레임에서 사용

```
no <- c(1:5)
```

```
name <- c("홍길동","이순신","강감찬","유관순","김유신")
```

```
score <- c(85,78,89,90,74)
```

```
exam <- data.frame(학번=no,이름=name,성적=score)
```

```
exam
```

```
which(exam$이름=="유관순") # [1] 4, 없으면 0
```

```
exam[4,] # 4번째 레코드 보기
```





## 4. 제어문과 함수

### 1) 반복문

형식) `for(변수 in 값) {표현식} - 단일문{} 생략 가능`

```
i <- c(1:10)
```

```
for(n in i){ # 10회 반복
```

```
 print(n * 10) # 계산식(numeric만 가능) 출력
```

```
 print(n)
```

```
}
```



## 4. 제어문과 함수

```
for(n in i){
 if(n%%2==0){
 next # 다음문장 skip -> 반복문 계속
 }else{
 print(n) # 홀수만 출력
 }
}
```

```
[1] 1
[1] 3
[1] 5
[1] 7
[1] 9
```



## 4. 제어문과 함수

```
데이터 파일의 변수명 출력
name <- c(names(exam))
for(n in name){ # 변수명 출력
 print(n)
}
```

```
[1] "학번"
[1] "이름"
[1] "성적"
```



## 4. 제어문과 함수

```
score = c(85, 95, 98)
```

```
name = c('홍길동', '이순신', '강감찬')
```

```
i <- 1
```

```
for (s in score){
```

```
 cat(name[i], " -> ", s, "\n")
```

```
 i <- i + 1
```

```
}
```

```
홍길동 -> 85
```

```
이순신 -> 95
```

```
강감찬 -> 98
```



## 4. 제어문과 함수

### 2) 반복문 - while(조건){표현식}

```
i = 0
while(i < 10){
 i <- i + 1
 print(i) # 1~10까지 출력됨
}
```



## 4. 제어문과 함수

### 3) 반복문 - repeat{ 탈출조건 }

```
cnt <- 1
repeat{
 print(cnt)
 cnt <- cnt + 2
 if(cnt > 15) break # cnt가 15보다 크면 탈출 - continue없음
}
```



## 4. 제어문과 함수

### Chap04\_2\_Function 수업내용

- 사용자 정의함수 형식
  - ✓ 사용자가 정의한 함수
- R 내장함수
  - ✓ R 설치 시 제공하는 함수



## 4. 제어문과 함수

- 사용자 정의함수 형식

(형식)

```
함수명 <- function(매개변수){ }
```

# 매개변수가 없는 함수 예

```
f1 <- function(){
 cat("매개변수가 없는 함수")
}
f1() # 함수 호출
```





## 4. 제어문과 함수

- 매개변수가 있는 함수 예

```
f2<- function(x){
 cat("x의 값 = ",x, "\n") # \n 줄바꿈
 print(x) # 변수만 사용
}
f2(15) # 함수 호출
```



## 4. 제어문과 함수

- 피타고라스 정의 증명- 식 :  $a^2 + b^2 = c^2$

```
pytha <- function(s,t){
 a <- s^2 - t^2
 b <- 2*s*t
 c <- s^2 + t^2
 cat("피타고라스의 정리 : 3개의 변수 : ",a,b,c)
}
```

```
pytha(2,1) # s,t는 양의 정수 -> 3 4 5
```



## 4. 제어문과 함수

- # 구구단 출력하기

```
gugu <- function(i,j){
 for(x in i){
 cat("***", x , "단 **\n")
 for(y in j){
 cat(x, "*", y, "=", x*y, "\n")
 }
 cat("\n")
 }
}
i<- c(2:9)
j<- c(1:9)
gugu(i,j)
```



## 4. 제어문과 함수

- 기술통계량 처리 내장함수

`min(vec)` # 벡터 대상 최소값

`max(vec)` # 벡터 대상 최대값

`range(vec)` # 벡터 대상 범위 값

`mean(vec)` # 벡터 대상 평균값

`median(vec)` # 벡터 대상 사분위수

`sum(vec)` # 벡터 대상 합계

`sort(x)` : 벡터 정렬 (단, 원래의 값을 바꾸지는 않음)

`order(x)` : 벡터의 정렬된 값의 인덱스를 보여줌

`rank(x)` : 벡터의 각 원소의 순위를 알려줌

`sd(x)` # 표준편차

`summary(x)` : 데이터에 대한 기본적인 통계 정보 요약

`table(x)` : 데이터 빈도수



## 4. 제어문과 함수

- 수학과 관련된 내장함수

`abs(x)` # 절대값

`sqrt(x)` # 제곱근

`ceiling(x)`, `floor()`, `round()` # 값의 올림, 내림, 반올림

`factorial(x)` # 팩토리얼 함수

`which.min(x)`, `which.max(x)` # 벡터 내의 최소값과 최대값의 인덱스

`pmin(x)`, `pmax(x)` # 여러 벡터에서의 원소 단위 최소값과 최대값

`prod()` # 벡터의 원소들의 곱

`cumsum()`, `cumprod()` # 벡터의 원소들의 누적합과 누적곱

`cos(x)`, `sin(x)`, `tan(x)` # 삼각함수 (also `acos(x)`, `cosh(x)`, `acosh(x)`, etc)

`log(x)` # 자연로그(natural logarithm)

`log10(x)` # 10을 밑으로 하는 일반로그 함수( $e^x$ )



## 4. 제어문과 함수

- 행렬연산 내장함수

`ncol(x)` # 열의 수

`nrow(x)` # 행의 수

`t(x)` # 전치행렬

`cbind(...)` # 열을 더할 때 이용되는 함수

`rbind(...)` # 행을 더할 때 이용되는 함수

`diag(x)` # 대각행렬

`det(x)` # 행렬식

`apply(x, m, fun)` # 행 또는 열에 함수 적용

`x %*% y` # 두 행렬의 곱

`solve(x)` # 역 행렬

`svd(x)` # Singular Value Decomposition

`qr(x)` # QR Decomposition (QR 분해)

`eigen(x)` # Eigenvalues(고유값)

`chol(x)` # choleski decomposition(Choleski 분해)



## 4. 제어문과 함수

- 집합연산 내장함수

`union (x, y)` # 집합  $x$ 와  $y$ 의 합집합

`intersect (x, y)` # 집합  $x$ 와  $y$ 의 교집합

`setdiff (x, y)` #  $x$ 의 모든 원소 중  $y$ 에는 없는  $x$ 와  $y$ 의 차집합

`setequal (x, y)` #  $x$ 와  $y$ 의 동일성 테스트

`c %in% y` #  $c$ 가 집합  $y$ 의 원소인지 테스트

`choose (n, k)` # 크기  $n$ 의 집합에서 크기  $k$ 의 가능한 부분 집합 개수



## 4. 제어문과 함수

- 기초 통계량 관련 함수

```
getwd()
```

```
setwd("c:/Rwork/Part-I")
```

```
#excel에서 csv(쉼표로 분리)형식으로 저장한 파일 가져오기
```

```
excel <- read.csv("excel.csv", header=TRUE)
```

```
head()함수이용 앞쪽 10줄 출력
```

```
head(excel,10) # q1 q2 q3 q4 q5
```

```
#colMeans()함수 이용 각 열의 평균 계산
```

```
colMeans(excel[1:5])
```

```
#q1 q2 q3 q4 q5
```

```
#2.733831 2.907960 3.621891 2.509950 3.385572
```

```
summary()함수 이용 각 열단위 기초 통계량
```

```
summary(excel)
```