KNU 강원대학교
KANGWON NATIONAL UNIVERSITY

11: 연구방법의 적용 IV. {tidyverse}로 데이터 다루기

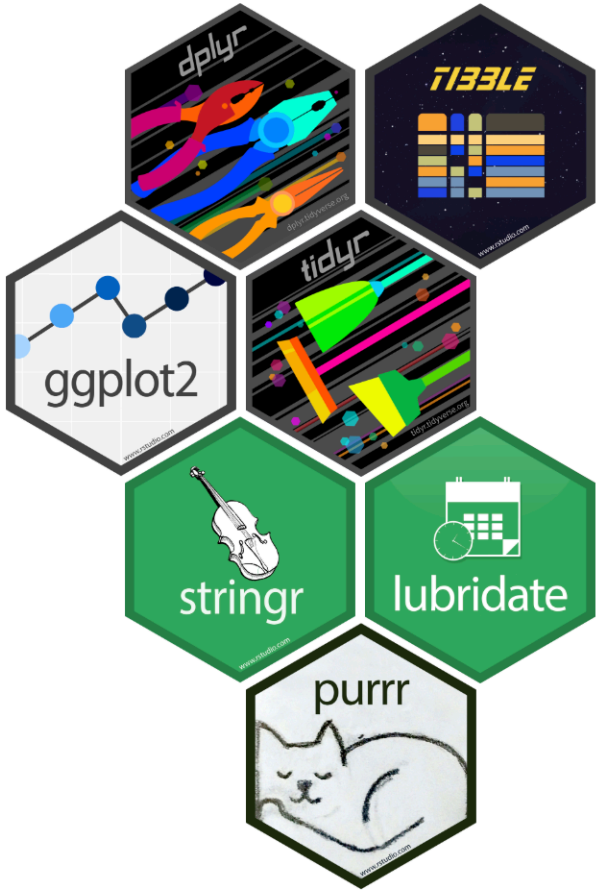
정치학연구방법론

박상훈 (sh.park.poli@gmail.com)
강원대학교

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{tidyverse}?

- 데이터 분석에 유용한 여러 패키지들의 모음
- 공식적인 {tidyverse}의 시작은 2016년
  - 이전부터 각각 구성 패키지들이 따로따로 사용되고 있었음.
  - 대표적인 {ggplot2} 패키지는 2005년에 만들어짐.



연구방법의 적용 IV. {tidyverse}로 데이터 다루기

핵심 구성 패키지

```
install.packages("tidyverse")
library(tidyverse)
```

- {ggpot2}: 데이터 시각화
- {dplyr}: 데이터 전처리
- {readr}: 데이터 불러오기
- {tibble}: 최신화된 데이터프레임
- {stringr}: 문자열 데이터 관리
- {forcats}: 요인(factor) 변수 관리
- {tidyr}: 데이터 정돈(tidy)
- {purrr}: 함수 프로그래밍

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

데이터 맛보기

서울시 생활이동 데이터

- 특정 시점에 서울 안에서 이동, 서울 외부에서 서울로 오고 간 이동
- 통근, 통학, 쇼핑, 여가 등 서울시의 행정수요를 유발하는 모든 이동을 의미
- 2021년 7월 (8시) 이동 데이터
- 자치구 코드 정보
- 학습 데이터 다운로드

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

데이터 불러오기

```
library(tidyverse)
moving_data <- read_csv("seoul_moving_202107_09_hr.csv")
reference_data <- readxl::read_excel("reference.xlsx")
```

데이터 파일 불러오기

이 경우에는 .csv 파일이므로 read_csv() 함수를 사용

만약 .xlsx 파일일 경우 다른 함수 사용. {readxl} 패키지의 read_xlsx() 혹은 read_excel() 함수 추천

- 이때, {readxl} 패키지에 속한 함수를 특정하여 부르고 싶을 경우에는 :: 사용
    - 예를 들어, readxl::read_excel()

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

데이터와 친해지기

기초 R에서는 데이터의 차원을 확인하기 위해 dim(), 데이터의 첫 부분을 확인하기 위해 head(), 뒷 부분을 확인하기 위해 tail() 함수를 사용

{tidyverse}에서는? glimpse() 함수!

```
dim(moving_data)
head(moving_data)
tail(moving_data)

# {tidyverse}
glimpse(moving_data)
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

데이터 변수 이름 바꾸기

변수 이름 설정

```
# reference_data
original_name_reference <-
  names(reference_data)
reference_data <-
  janitor::clean_names(reference_data)
reference_data |> names()
```

```
## [1] "sido"      "sigungu"   "name"
```

```
# moving_data
original_name_moving <-
  names(moving_data)
moving_data <- janitor::clean_names(movin
moving_data |> names()
```

```
##  [1] "daesang_yeon_wol"             "yoi
##  [3] "dochagsigan"                  "chu
##  [5] "dochag_sigungu_kodeu"         "sec
##  [7] "nai"                          "idc
##  [9] "pyeong_gyun_idong_sigan_bun" "idc
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

행(row) 관련 동사들

- distinct()
- filter()
- slice()

열(column) 관련 동사들

- select()
- rename()
- mutate()

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

시도 단위는 몇 개인가? 중복 없는 표본을 만들자

distinct(): .keep_all = TRUE 설정으로 딸려 있는 데이터 보관하기

```
reference_data |>
  distinct(sido) |>
  count()
```

```
## # A tibble: 1 × 1
##       n
##   <int>
## 1    17
```

```
reference_data |>
  distinct(sido, .keep_all = TRUE) |>
  dim()
```

```
## [1] 17  4
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

원하는 행들을 걸러(filter)내는 방법

사용 가능 함수들

- 연산자들
  - ==, >, >=, &, |, !
- 유용한 함수들:
  - is.na()
  - between(), near()

```
moving_data |>
  dplyr::filter(
    yoil == "일" & seongbyeol == "F"
  )
```

```
## # A tibble: 62,076 × 10
##    daesang_yeon_wol yoil  dochagsigan c
##               <dbl> <chr> <chr>
##  1           202107 일    09
##  2           202107 일    09
##  3           202107 일    09
##  4           202107 일    09
##  5           202107 일    09
##  6           202107 일    09
##  7           202107 일    09
##  8           202107 일    09
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

원하는 행들을 잘라내는(slice) 방법

기본 함수인 head()의 확장버전

- 기본적으로 인덱싱을 제공
  - 맨 마지막 인덱스: n()
- 유용한 함수들
  - slice_min(), slice_max(), slice_sample(), slice_head(), slice_tail()

```
moving_data |>
  slice(15:20)
```

```
## # A tibble: 6 × 10
##   daesang_yeon_wol yoil  dochagsigan ch
##              <dbl> <chr> <chr>
## 1           202107 일    09
## 2           202107 일    09
## 3           202107 일    09
## 4           202107 일    09
## 5           202107 일    09
## 6           202107 일    09
## # ℹ 5 more variables: seongbyeol <chr>,
## #   pyeong_gyun_idong_sigan_bun <dbl>,
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

내 맘대로 행을 정렬(arrange)하는 방법

```
df |>
  arrange(col1, desc(col2))
```

- 정렬 기준 우선 순위 순서대로 설정
- 내림차순은 desc() (decreasing)

```
moving_data |>
  dplyr::select(
    dochagsigan,
    pyeong_gyun_idong_sigan_bun) |>
  arrange(dochagsigan,
          desc(pyeong_gyun_idong_sigan_bu

moving_data |> mutate(yoil = factor(yoil,
  levels = c("월", "화", "수","목",
             "금", "토", "일"))) |>
  arrange(yoil)
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

원하는 열을 선택(select)하는 방법

사용할 수 있는 옵션들

사용가능 연산자들

- `:`, `!`, `&`, `|`, `c()` 사용가능

편리한 함수들

- `everything()`, `last_col()`
- `starts_with()`, `ends_with()`, `contains()`

```
reference_data |>
  dplyr::select(sido, full_name) |>
  head()
```

```
## # A tibble: 6 × 2
##    sido full_name
##   <dbl> <chr>
## 1 11000 서울특별시 종로구
## 2 11000 서울특별시 중구
## 3 11000 서울특별시 용산구
## 4 11000 서울특별시 성동구
## 5 11000 서울특별시 광진구
## 6 11000 서울특별시 동대문구
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

나만의 열을 생성(mutate)하는 방법

new = old 문법

```
mutate(new = old_1 + old_2)
```

궁합 좋은 함수들

- case_when(), if_else()

년도 정보만 빼내오기

```
moving_data |>
  mutate(
    year =
      substr(daesang_yeon_wol, 1, 4),
    year = as.integer(year)) |>
  dplyr::select(year, everything())
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

열이름 다시 정하기(rename)

```
df |> rename(new = old)
df |> rename_with(function)
```

A를 a_new로 바꾸기

```
df |> rename(a_new = A)
```

모든 열이름 대문자로 바꾸기

```
df |> rename_with(toupper)
```

실습하기

moving_data의 pyeong_gyun_idong_sigan_bun이라는 열 이름을 avg_time_min으로, idong_ingu_hab을 ppl_sum이라는 이름으로 다시 정해보기.

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

테이블 합치기

두 개의 테이블 형태의 데이터가 있다고 하자.

tab_a

id	score
1	35
3	70
4	63
5	80

tab_b

id	name
1	sanghoon
2	inhye
3	younghoon
4	yoojin

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

id 정보를 기준으로 테이블 2개를 합치기

- left_join(x, y): 테이블을 **왼쪽으로** 합쳐줘. $ x <- y $

```
tab_a |> left_join(
  tab_b, by = c("id")
)
```

```
##   id score      name
## 1  1    35  sanghoon
## 2  3    70 younghoon
## 3  4    63    yoojin
## 4  5    80      <NA>
```

옵션들

by: 합칠 때의 기준열

- by = c("col1" = "col2")

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

left_join()

좌측의 데이터를 기준으로 병합

```
tab_a |> left_join(
  tab_b, by = c("id")
)
```

```
##   id score       name
## 1  1    35  sanghoon
## 2  3    70 younghoon
## 3  4    63    yoojin
## 4  5    80       <NA>
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

right_join()

우측의 데이터를 기준으로 병합(left_join()과 방향만 다름

```
tab_a |> right_join(
  tab_b, by = c("id")
)
```

```
##   id score       name
## 1  1    35  sanghoon
## 2  3    70 younghoon
## 3  4    63    yoojin
## 4  2    NA     inhye
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

inner_join()

교집합만 빼서 테이블 생성

```
tab_a |> inner_join(
  tab_b, by = c("id")
)
```

```
##   id score      name
## 1  1    35  sanghoon
## 2  3    70 younghoon
## 3  4    63    yoojin
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

full_join()

모든 데이터를 보존하여 테이블을 병합, 생성

```
tab_a |> full_join(
  tab_b, by = c("id")
)
```

```
##   id score       name
## 1  1    35  sanghoon
## 2  3    70 younghoon
## 3  4    63    yoojin
## 4  5    80       <NA>
## 5  2    NA     inhye
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 기본 동사 학습

하나의 테이블을 다른 테이블로 필터(Filtering join)

`semi_join()`: 매칭 데이터를 보존하는 방식으로 테이블을 생성

`anti_join()`: 매칭 데이터를 제거하는 방식으로 테이블을 생성

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

if_else()

특정한 조건으로 값 만들기

```
tab_a |>
  mutate(
    status = if_else(
      score < 40,
      "low", "high"
    )
  )
```

```
##   id score status
## 1  1    35    low
## 2  3    70   high
## 3  4    63   high
## 4  5    80   high
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

if_else()

특정한 조건으로 값 만들기

```
tab_a |>
  mutate(
    status = case_when(
      score < 40 ~ "low",
      score >= 40 & score < 70 ~ "middle"
      score >= 70 ~ "high",
      T ~ NA_character_
    )
  )
```

```
##   id score status
## 1  1    35    low
## 2  3    70   high
## 3  4    63 middle
## 4  5    80   high
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

실습하기

이동시간을 이용한 trip 분류

- 이동시간을 시간단위로 바꾸는 새로운 변수를 만들어 보세요: trip_time_hr
- trip_time_hr을 이용해서 trip_time_class를 만들어 보세요.

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

그룹데이터 다루기: group_by()

grouping이 된 데이터 프레임은 프린트를 하면 표시가 됨.

- 궁합이 좋은 함수로는 count()와 tally() 함수가 있음.

```
tab_a |> group_by(status)
```

```
## # A tibble: 4 × 3
## # Groups:   status [3]
##      id score status
##   <dbl> <dbl> <chr>
## 1     1    35 low
## 2     3    70 high
## 3     4    63 middle
## 4     5    80 high
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

그룹데이터 다루기: group_by()

count()와 tally()는 그룹 변수별로 몇 개의 관측치가 있는지를 세어 줌.

```
tab_a |> group_by(status) |> tally()
```

```
## # A tibble: 3 × 2
##   status     n
##   <chr>  <int>
## 1 high       2
## 2 low        1
## 3 middle     1
```

```
tab_a |> group_by(status) |> count()
```

```
## # A tibble: 3 × 2
## # Groups:   status [3]
##   status     n
##   <chr>  <int>
## 1 high       2
## 2 low        1
## 3 middle     1
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

그룹데이터 다루기: group_by()

그리고 집단별로 정렬 순서를 바꿀 수도 있음.

```
tab_a |> group_by(status) |>
  arrange(id)
```

```
## # A tibble: 4 × 3
## # Groups:   status [3]
##      id score status
##   <dbl> <dbl> <chr>
## 1     1    35 low
## 2     3    70 high
## 3     4    63 middle
## 4     5    80 high
```

```
tab_a |> group_by(status) |>
  arrange(id, .by_group = T)
```

```
## # A tibble: 4 × 3
## # Groups:   status [3]
##      id score status
##   <dbl> <dbl> <chr>
## 1     3    70 high
## 2     5    80 high
## 3     1    35 low
## 4     4    63 middle
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

그룹데이터 다루기: 그룹 키와 그룹 해제

- group_keys(): 현재 묶인 그룹의 고유한 값을 보여줌.
- ungroup()을 사용해서 그룹을 해제해줄 수 있음.

```
tab_a |> group_by(status) |> group_keys()
```

```
## # A tibble: 3 × 1
##   status
##   <chr>
## 1 high
## 2 low
## 3 middle
```

```
tab_a |> group_by(status) |> ungroup()
```

```
## # A tibble: 4 × 3
##      id score status
##   <dbl> <dbl> <chr>
## 1     1    35 low
## 2     3    70 high
## 3     4    63 middle
## 4     5    80 high
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

실습문제

서울시 안에서 short trip이 가장 많이 일어나는 곳은?

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

시각화의 문법: ggplot2()

그래프를 우리가 직관적으로 언어의 문법을 이해하듯이 바라보자.

> **나는** *이 세상에서* 가난하고 외롭고 높고 쓸쓸하니 살아가도록 태어났다.

우리가 일상생활 속에서 사용하는 문장들은 각각의 문법 요소들로 구성되어 있음.

- 한 문장을 바라볼 때, 자연스럽게 의미를 띄는 부분들로 이해함.
- 그래프를 만들 때에도 그래프를 여러 요소들로 쪼개어 그리게 됨.

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

palmerpenguins 데이터

R에서 가장 유명한 팽귄들

```
# devtools::install_github("allisonhorst/palmerpenguins")
library(palmerpenguins)
```

- 기초적인 데이터 핸들링과 시각화에 좋은 데이터

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

그룹데이터 다루기: group_by()

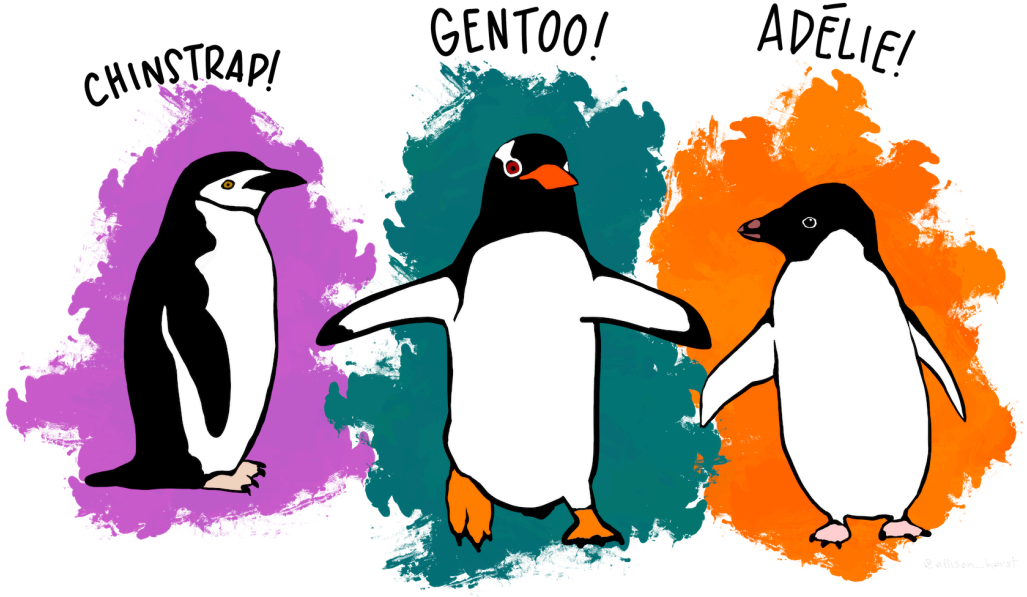
연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

palmerpenguins 데이터

팽귄의 종류별 부리 길이 측정

- 팽귄 종류
    - Chinstrap, Gentoo, Adelie



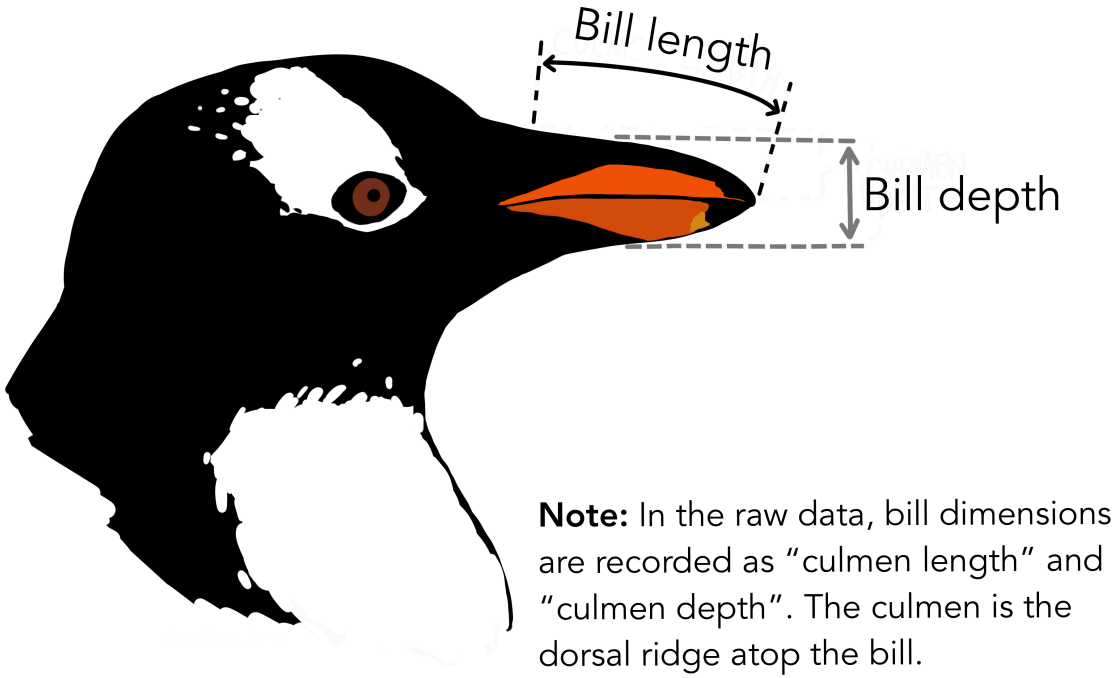
연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

palmerpenguins 데이터

팽귄의 종류별 부리 길이 측정

- 팽귄의 서식지
- 팽귄의 부리길이, 부리깊이, 날개길이
  - 측정단위: mm
  - flipper: 팽귄의 날개
- 성별 및 몸무게 정보



연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

팽귄데이터의 구조

주요 정보

344개의 관측치를 가진 표본

NA가 섞여 있음.

8개의 변수

```
glimpse(penguins)
```

```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie
## $ island            <fct> Torgersen, Torgersen, Torgerse
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 1
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 36
## $ sex               <fct> male, female, female, NA, fema
## $ year              <int> 2007, 2007, 2007, 2007, 2007,
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

기본적인 전처리

```
# 결측치 제거
penguins |>
  drop_na() -> penguins_nona
penguins_nona
```

```
## # A tibble: 333 × 8
##    species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##    <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
##  1 Adelie  Torgersen           39.1          18.7               181        3750
##  2 Adelie  Torgersen           39.5          17.4               186        3800
##  3 Adelie  Torgersen           40.3          18                 195        3250
##  4 Adelie  Torgersen           36.7          19.3               193        3450
##  5 Adelie  Torgersen           39.3          20.6               190        3650
##  6 Adelie  Torgersen           38.9          17.8               181        3625
##  7 Adelie  Torgersen           39.2          19.6               195        4675
##  8 Adelie  Torgersen           41.1          17.6               182        3200
##  9 Adelie  Torgersen           38.6          21.2               191        3800
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

그래프와 코드 비교

```
penguins |> ggplot() +
  aes(x = bill_length_mm,
      y = bill_depth_mm,
      color = species) +
  geom_point(shape = "circle",
             size = 1.5) +
  scale_color_manual(
    values = c(Adelie = "#F8766D",
               Chinstrap = "#00C19F",
               Gentoo = "#FF61C3")) +
ggthemes::theme_fivethirtyeight() +
theme(legend.position = "bottom")
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

ggplot()의 구성

그래프를 구성하는 여러 레이어

- X축과 Y축을 이어주는 레이어
- 그래프의 요소들을 설정하는 레이어
- 그래프의 색을 설정하는 레이어
- 그래프의 테마를 설정하는 레이어
- 범례의 위치를 설정하는 레이어

```
penguins |> ggplot() +
  aes(x = bill_length_mm,
      y = bill_depth_mm,
      color = species) +
  geom_point(shape = "circle",
             size = 1.5) +
  scale_color_manual(
    values = c(Adelie = "#F8766D",
               Chinstrap = "#00C19F",
               Gentoo = "#FF61C3")) +
ggthemes::theme_fivethirtyeight() +
theme(legend.position = "bottom")
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

도화지 준비 단계

그래프의 밑바닥 준비

정보의 원천이 되는 자료들을 등록

- ggplot()은 입력 자료가 data.frame 또는 tibble의 형태를 띄고 있어야 함.
- 자료는 무조건 사각형의 tidy 형태
  - 변수를 나타내는 열
  - 표본을 나타내는 행

```
p <- ggplot(data = penguins)
p
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

aesthetics 레이어

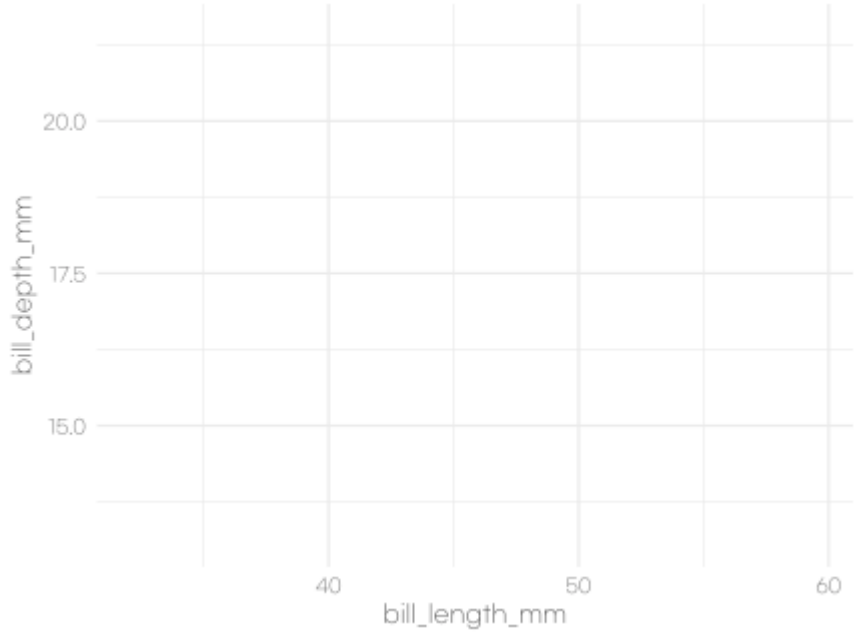
그래프의 속성을 데이터의 정보와 연결하는 레이어

aesthetics에서 설정하는 요소들

- x축과 y축
- alpha: 투명도
- color, fill: 색과 채우기 속성
- shape, size: 모양과 크기 등

데이터의 **정보**와 **요소**들을 이어줌.

```
p <- p + aes(x = bill_length_mm,
             y = bill_depth_mm)
p
```



연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

기하객체(Geometric objects) 레이어

그래프의 메인 요소: 어떤 기하학적인 요소를 사용하여 정보를 표현할 것인가?

- 선, 면, 공간
- 삼각형, 사각형, 다각형, 원 등

ggplot2 패키지의 함수들

- `geom_point()`, `geom_path()`, `geom_bar()`, `geom_ribbon()` 등

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

도화지 준비 단계

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

도화지 준비 단계

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

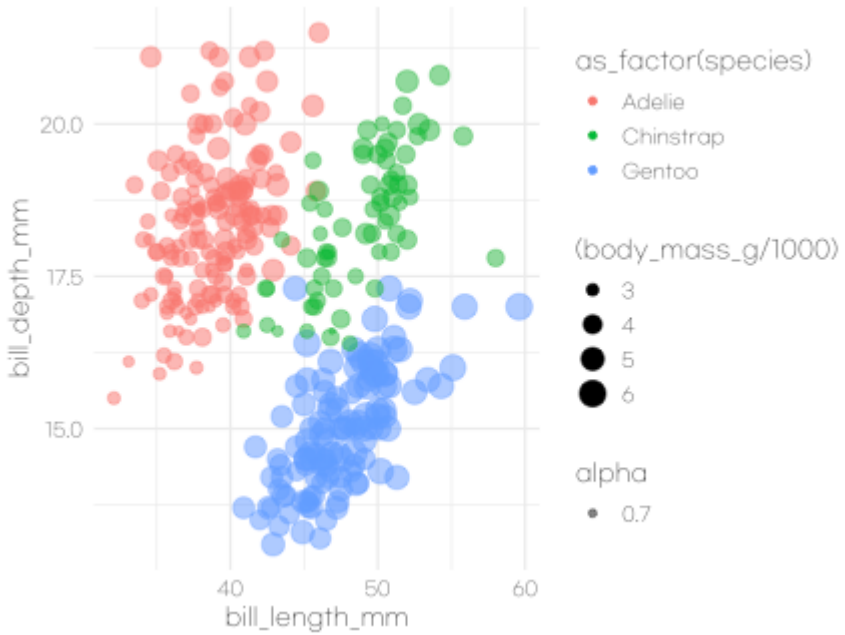
{dplyr} 패키지 고급 동사 학습

기하객체(Geometric objects) 레이어

```
p <- p +
  geom_point(
    aes(color = as_factor(species),
        size = (body_mass_g / 1000),
        alpha = 0.7))
p
```

geom_point()를 사용

레이어별로 특정한 aesthetic을 연결하여 그릴 수 있음.



연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

aes() + geom() 콤보

상황에 맞게 선택

1. 데이터 1개에 맵핑 1개
2. 데이터 1개, 맵핑 여러개
3. 데이터 여러 개, 맵핑 여러 개

```
# Method 1
ggplot(data = penguins,
       aes(x = body_mass_g, y = bill_length_mm))

# Method 2
ggplot(data = penguins) +
  geom_point(aes(x = body_mass_g, y = bill_length_mm))

# Method 3
ggplot() +
  geom_point(data = penguins,
             aes(x = body_mass_g, y = bill_length_mm))
```

연구방법의 적용 IV. {tidyverse}로 데이터 다루기

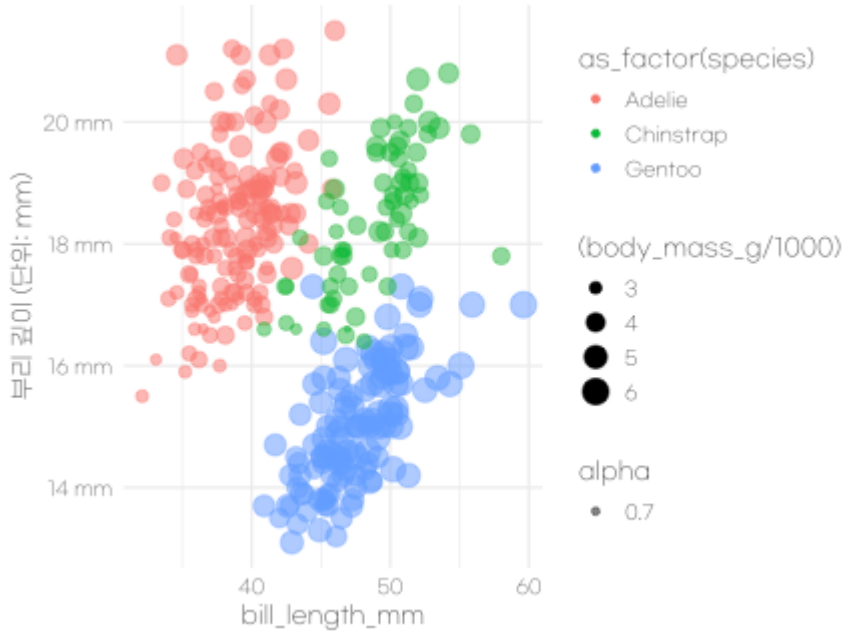
{dplyr} 패키지 고급 동사 학습

{scales} 레이어: y-축 설정

```
p + scale_y_continuous(
  "부리 깊이 (단위: mm)",
  breaks = seq(0, 30, by = 2),
  labels = paste(seq(0, 30, by = 2), "mm"
  minor_breaks = NULL)
```

구조: scale_<aes>_<type>()

- scale_y_continuous(): 연속형 Y 축 변수 관련 속성을 확정

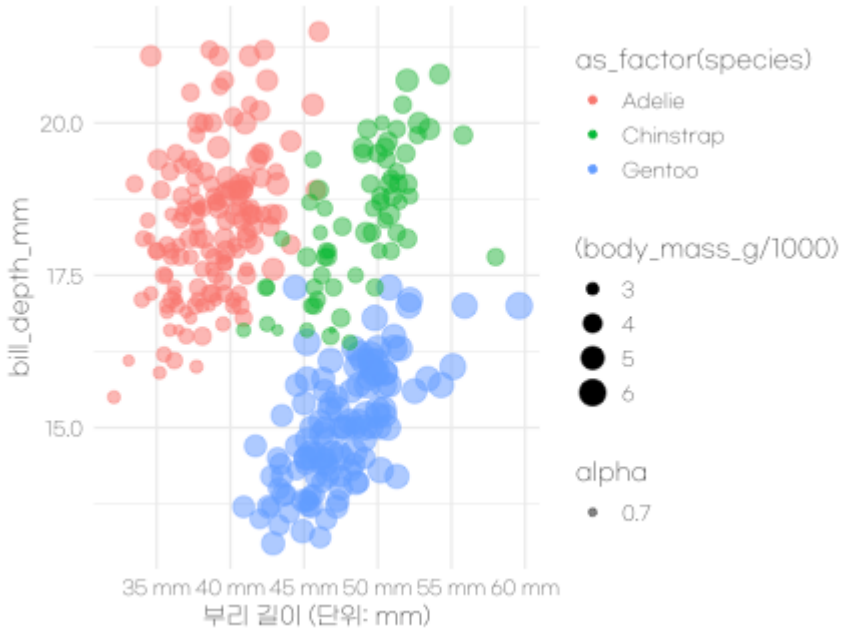


연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

{scales} 레이어: x-축 설정

```
p + scale_x_continuous(
  "부리 길이 (단위: mm)",
  breaks = seq(30, 60, by = 5),
  labels = paste(seq(30, 60, by = 5), "mm
  minor_breaks = NULL)
```

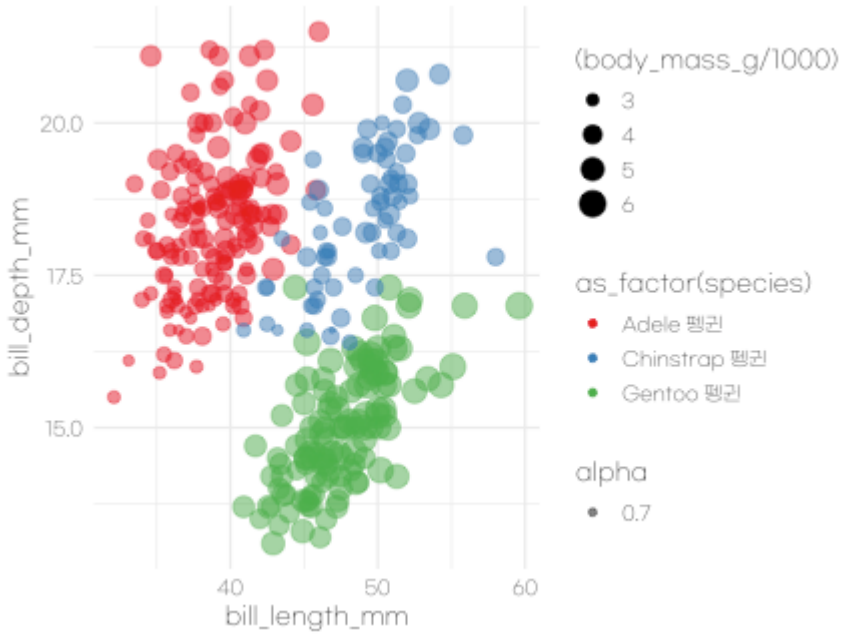


연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

{scales} 레이어: color 설정

```
p + scale_color_brewer(
  palette = "Set1",
  labels = c("Adele 펭귄",
             "Chinstrap 펭귄",
             "Gentoo 펭귄"))
```



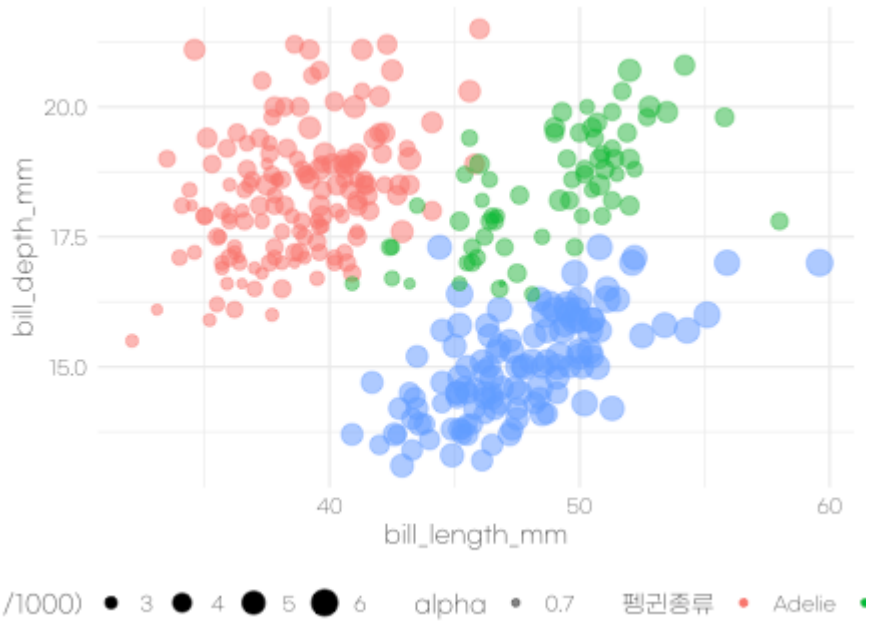
연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

범례(Legend) 설정

```
my_species <- guide_legend(
  title = "펭귄종류", ncol = 3)
p <- p +
  guides(color = my_species) +
  theme(legend.position = "bottom")
```

범례 이름은 guide_legend()를 사용



연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

facet 레이어를 통한 변수별 그래프

facet?: a particular aspect or feature of something

- 데이터 안의 특정 변수를 사용하여 여러 개의 데이터 특성을 나타낼 때
- 하나의 변수로 펼칠 때: `facet_wrap()`
- 2개의 변수 조합으로 펼칠 때: `facet_grid()`

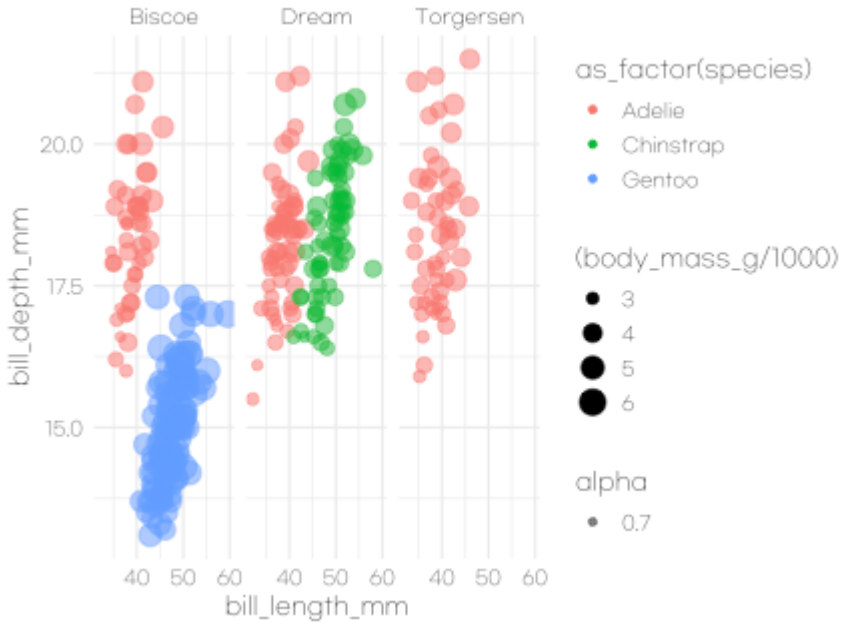
연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

각 서식지별 펭귄 종류 분석

```
p + facet_wrap(~island)
```

facet으로 펼칠 변수를 ~로 구분하여 준다.



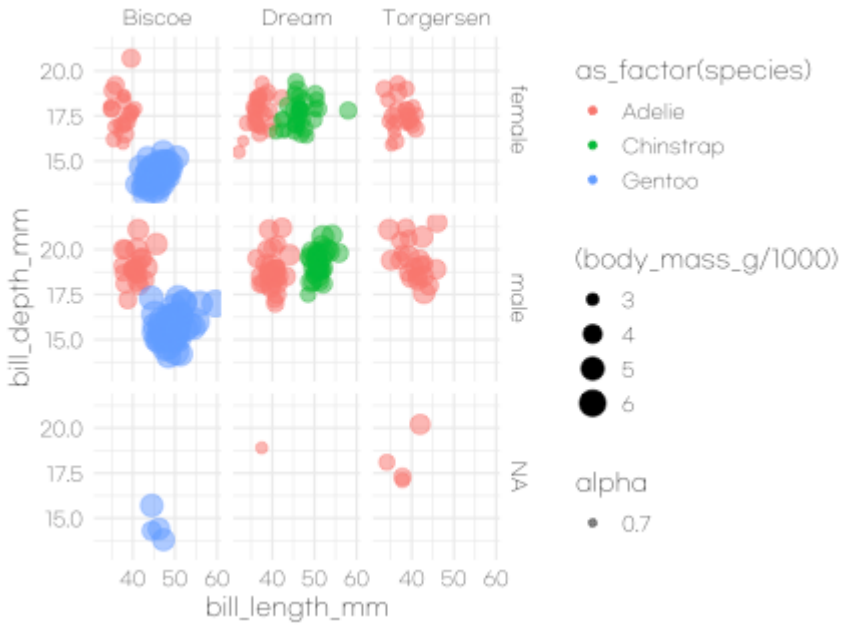
연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

각 서식지별 펭귄 종류 분석

```
p + facet_grid(sex~island)
```

facet_grid()는 두 변수를 사용하여 데이터의 특성을 보여줌.

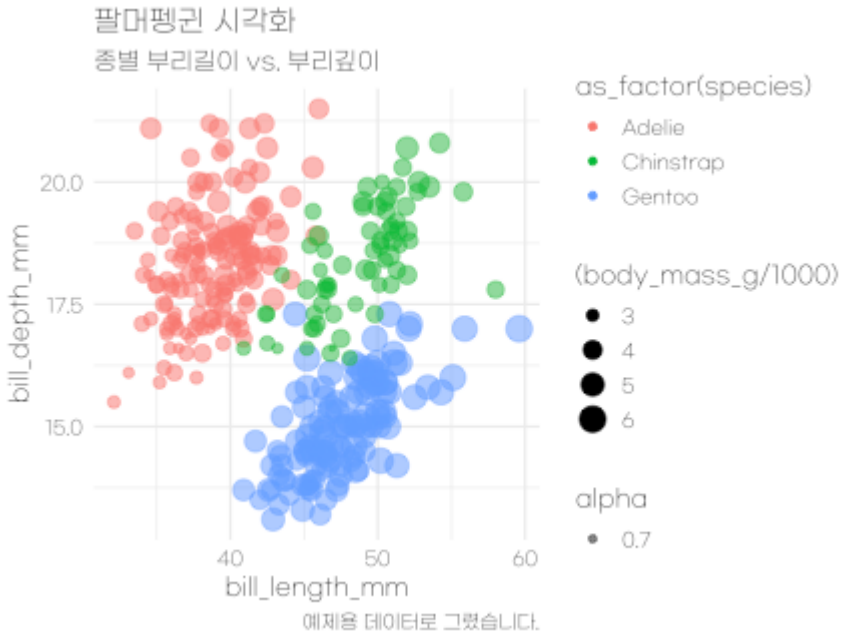


연구방법의 적용 IV. {tidyverse}로 데이터 다루기

{dplyr} 패키지 고급 동사 학습

제목, 부제목, 캡션 넣기

```
p + labs(
  title = "팔머펭귄 시각화",
  subtitle = "종별 부리길이 vs. 부리깊이",
  caption = "예제용 데이터로 그렸습니다.")
```



`labs()` 함수를 사용한 제목 설정

- X, Y축 제목도 설정 가능함: `x = "x 축 제목"`

KNU 강원대학교
KANGWON NATIONAL UNIVERSITY

감사합니다!

궁금한 것이 있으면 언제든 연락하세요.

강사 연락처

연락처	박상훈
	sh.park.poli@gmail.com
	sanghoon-park.com/
	영상바이오관 405