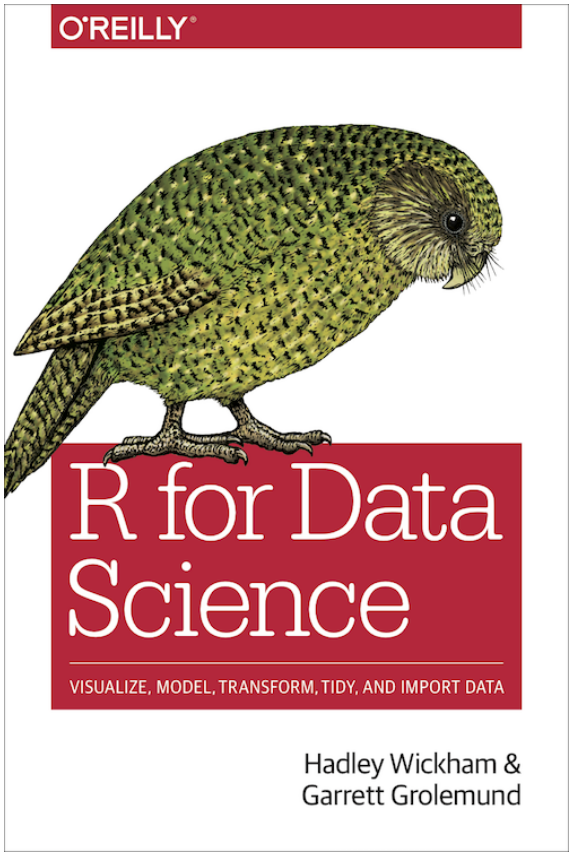
Socio-Informatics 348

Data Visualisation
Numbers

Dr Lisa Martin

Department of Information Science
Stellenbosch University

Today's Reading



R for Data Science, Chapter 13

Making Numbers

- Typically, you receive data with numbers already in numeric format
- You may sometimes encounter numbers as strings

```
x <- c("1.2", "5.6", "1e3")
parse_double(x)
#> [1]    1.2    5.6 1000.0
```

- `parse_number()` is useful for numbers with units/text

```
x <- c("$1,234", "USD 3,513", "59%")
parse_number(x)
#> [1] 1234 3513   59
```

Counts

- You've already been introduced to `count()` and `summarise(n)`
- Use `count()` for quicker calculations

```
flights |> count(dest, sort = TRUE)
#> # A tibble: 105 × 2
#>   dest      n
#>   <chr> <int>
#> 1 ORD   17283
#> 2 ATL   17215
#> 3 LAX   16174
#> 4 BOS   15508
#> 5 MCO   14082
#> 6 CLT   14064
#> # i 99 more rows
```

- Weighted counts:

```
flights |> count(tailnum, wt = distance)
```

Counts

- Use `summarise()` when you want to do multiple/more complex calculations.

```
flights |>
  group_by(dest) |>
  summarize(
    n = n(),
    delay = mean(arr_delay, na.rm = TRUE)
  )
#> # A tibble: 105 × 3
#>   dest      n delay
#>   <chr> <int> <dbl>
#> 1 ABQ     254  4.38
#> 2 ACK     265  4.85
#> 3 ALB     439 14.4
#> 4 ANC       8 -2.5
#> 5 ATL   17215 11.3
#> 6 AUS    2439  6.02
#> # i 99 more rows
```

Counts

- Count the number of unique values per group (`n_distinct()`).

```
flights |>
  group_by(dest) |>
  summarize(carriers = n_distinct(carrier)) |>
  arrange(desc(carriers))
#> # A tibble: 105 × 2
#>   dest  carriers
#>   <chr>    <int>
#> 1 ATL          7
#> 2 BOS          7
#> 3 CLT          7
#> 4 ORD          7
#> 5 TPA          7
#> 6 AUS          6
#> # i 99 more rows
```

Counts

- Combine `sum()` and `is.na()` to count missing values.

```
flights |>
  group_by(dest) |>
  summarize(n_cancelled = sum(is.na(dep_time)))
#> # A tibble: 105 × 2
#>   dest  n_cancelled
#>   <chr>       <int>
#> 1 ABQ             0
#> 2 ACK             0
#> 3 ALB            20
#> 4 ANC             0
#> 5 ATL           317
#> 6 AUS            21
#> # i 99 more rows
```

Numeric Transformations

- Min and Max – Row-wise

```
df <- tribble(
  ~x, ~y,
  1,  3,
  5,  2,
  7, NA,
)

df |>
  mutate(
    min = pmin(x, y, na.rm = TRUE),
    max = pmax(x, y, na.rm = TRUE)
  )
#> # A tibble: 3 × 4
#>       x     y   min   max
#>   <dbl> <dbl> <dbl> <dbl>
#> 1     1     3     1     3
#> 2     5     2     2     5
#> 3     7    NA     7     7
```

Numeric Transformations

- Min and Max – Column-wise

```
df |>
  mutate(
    min = min(x, y, na.rm = TRUE),
    max = max(x, y, na.rm = TRUE)
  )
#> # A tibble: 3 × 4
#>       x     y   min   max
#>   <dbl> <dbl> <dbl> <dbl>
#> 1     1     3     1     7
#> 2     5     2     1     7
#> 3     7    NA     1     7
```

Numeric Transformations

- Modular arithmetic

```
1:10 %/% 3
#>  [1] 0 0 1 1 1 2 2 2 3 3
1:10 %% 3
#>  [1] 1 2 0 1 2 0 1 2 0 1
```

Numeric Transformations

- Modular arithmetic

```
flights |>
  mutate(
    hour = sched_dep_time %/% 100,
    minute = sched_dep_time %% 100,
    .keep = "used"
  )
#> # A tibble: 336,776 × 3
#>   sched_dep_time  hour minute
#>            <int> <dbl>  <dbl>
#> 1            515     5     15
#> 2            529     5     29
#> 3            540     5     40
#> 4            545     5     45
#> 5            600     6      0
#> 6            558     5     58
#> # ℹ 336,770 more rows
```

Numeric Transformations

- Rounding

```
round(123.456)
#> [1] 123
```

```
round(123.456, 2)  # two digits
#> [1] 123.46
round(123.456, 1)  # one digit
#> [1] 123.5
round(123.456, -1) # round to nearest ten
#> [1] 120
round(123.456, -2) # round to nearest hundred
#> [1] 100
```

```
x <- 123.456

floor(x)
#> [1] 123
ceiling(x)
#> [1] 124
```

Numeric Transformations

- Cutting ranges

```
x <- c(1, 2, 5, 10, 15, 20)
cut(x, breaks = c(0, 5, 10, 15, 20))
#> [1] (0,5]   (0,5]   (0,5]   (5,10]  (10,15] (15,20]
#> Levels: (0,5] (5,10] (10,15] (15,20]
```

```
cut(x, breaks = c(0, 5, 10, 100))
#> [1] (0,5]    (0,5]    (0,5]    (5,10]   (10,100] (10,100]
#> Levels: (0,5] (5,10] (10,100]
```

```
y <- c(NA, -10, 5, 10, 30)
cut(y, breaks = c(0, 5, 10, 15, 20))
#> [1] <NA>   <NA>   (0,5]  (5,10] <NA>
#> Levels: (0,5] (5,10] (10,15] (15,20]
```

Numeric Transformations

- Cutting ranges – with labels
- One less label than number of breaks

```
cut(x,
  breaks = c(0, 5, 10, 15, 20),
  labels = c("sm", "md", "lg", "xl")
)
#> [1] sm sm sm md lg xl
#> Levels: sm md lg xl
```

General Transformations

- Transformations typically used with numerical variables
  - Ranks

```
x <- c(1, 2, 2, 3, 4, NA)
min_rank(x)
#> [1]  1  2  2  4  5 NA
```

```
min_rank(desc(x))
#> [1]  5  3  3  2  1 NA
```

  - Offsets
  - Consecutive identifiers

General Transformations

- Transformations typically used with numerical variables
  - Ranks
  - Offsets

```
x <- c(2, 5, 11, 11, 19, 35)
lag(x)
#> [1] NA  2  5 11 11 19
lead(x)
#> [1]  5 11 11 19 35 NA
```

  - Consecutive identifiers

General Transformations

- Transformations typically used with numerical variables
  - Consecutive identifiers

```
df <- tibble(
  x = c("a", "a", "a", "b", "c", "c", "d", "e", "a", "a", "b", "b"),
  y = c(1, 2, 3, 2, 4, 1, 3, 9, 4, 8, 10, 199)
)
```

```
df |>
  group_by(id = consecutive_id(x)) |>
  slice_head(n = 1)
#> # A tibble: 7 × 3
#> # Groups:   id [7]
#>   x         y    id
#>   <chr> <dbl> <int>
#> 1 a         1     1
#> 2 b         2     2
#> 3 c         4     3
#> 4 d         3     4
#> 5 e         9     5
#> 6 a         4     6
#> # ℹ 1 more row
```

Numeric Summaries

Summary functions to use with `summarise()`

- Centre
  - `mean()`, `median()`
- Min, Max and Quantiles
  - `min()`, `max()`, `quantile()`

```
summarize(
  max = max(dep_delay, na.rm = TRUE),
  q95 = quantile(dep_delay, 0.95, na.rm = TRUE),
  .groups = "drop"
)
```

- Spread
  - `IQR()`

```
summarize(
  distance_iqr = IQR(distance),
  n = n(),
  .groups = "drop"
) |>
```

Numeric Summaries

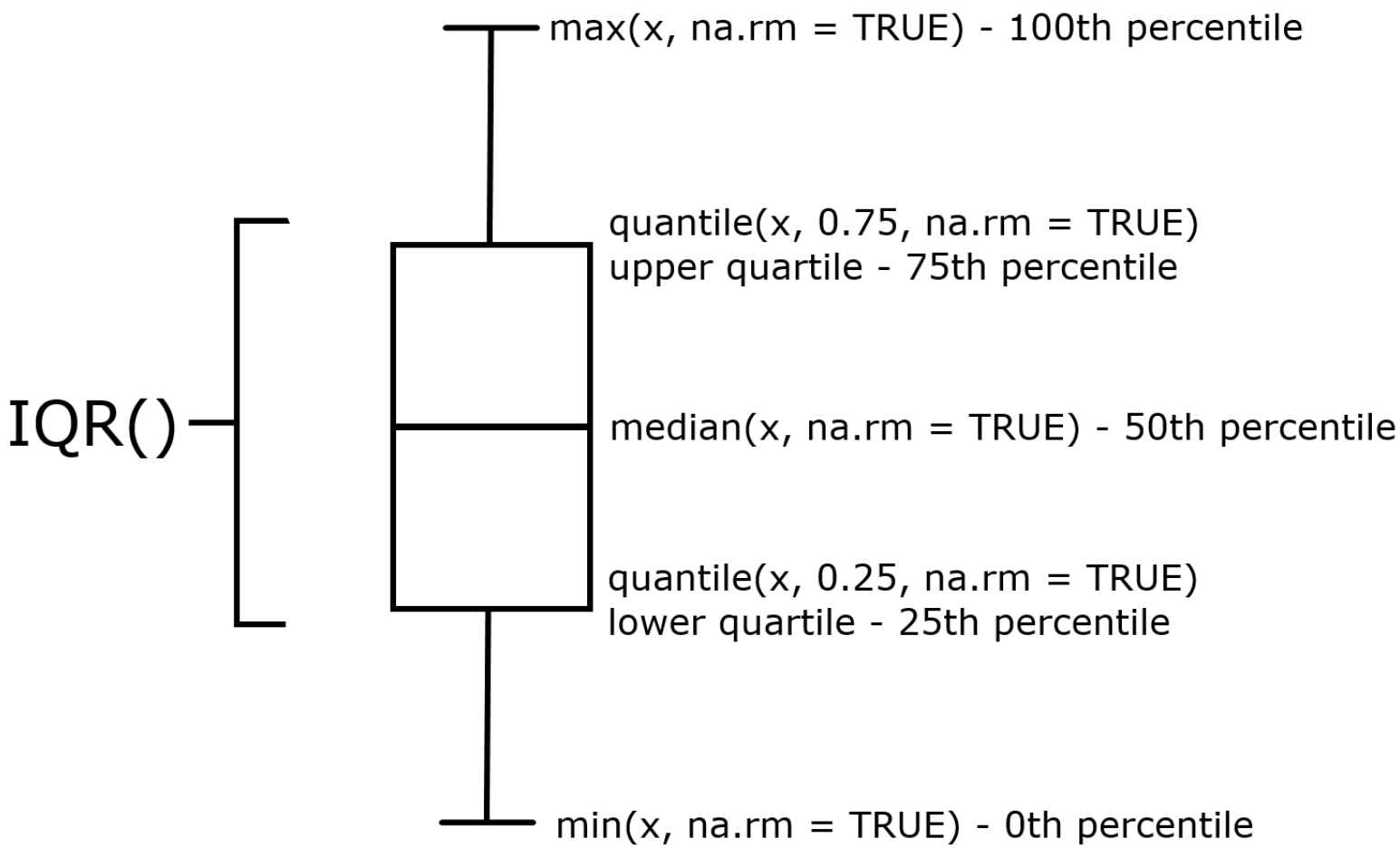
Summary functions to use with `summarise()`

- Positions
  - `first()`, `last()`, `nth()`

```
summarize(
  first_dep = first(dep_time, na_rm = TRUE),
  fifth_dep = nth(dep_time, 5, na_rm = TRUE),
  last_dep = last(dep_time, na_rm = TRUE)
)
```

Numeric Summaries

Summary functions to use with `summarise()`



with `mutate()`

- `x / sum(x)` to get proportion of the total
- `(x - mean(x)) / sd(x)` to get z-scores
- `(x - min(x)) / (max(x) - min(x))` to get min-max normalization; range [0, 1]
- `x / first(x)` to create an index based on first value/observation