Modul 2 Data Wrangling

Mika Alvionita Sitinjak

09 September 2025

Pendahuluan Data Wrangling

Komputasi statistik berkaitan erat dengan teknik muging (atau data munging, sering juga disebut data wrangling), karena keduanya sama-sama berfokus pada pengolahan data sebelum menghasilkan analisis yang bermakna. Komputasi statistik menyediakan metode, algoritma, dan perangkat lunak untuk melakukan perhitungan numerik yang efisien—misalnya estimasi parameter, simulasi, atau uji hipotesis—yang hanya dapat berjalan optimal jika data dalam kondisi bersih dan terstruktur. Di sisi lain, teknik munging berperan sebagai tahap awal untuk menyiapkan data mentah: membersihkan kesalahan, mengubah format, menggabungkan sumber data, hingga mengekstraksi variabel yang relevan. Dengan demikian, munging dapat dipandang sebagai prasyarat praktis agar komputasi statistik dapat diterapkan secara efektif. Tanpa data munging, proses komputasi statistik sering kali menghasilkan output yang bias, tidak konsisten, atau bahkan tidak dapat dijalankan karena masalah kualitas data.

Dalam konteks R, kaitan antara komputasi statistik dan data munging terlihat sangat jelas. R adalah bahasa yang kuat untuk komputasi statistik—dengan fungsi bawaan maupun paket tambahan untuk analisis multivariat, regresi, time series, hingga machine learning. Namun, sebelum analisis tersebut dilakukan, data hampir selalu perlu melalui tahap munging agar sesuai dengan format analisis. Paket $dplyr$ dan $tidyr$ dari ekosistem tidyverse misalnya, menyediakan fungsi-fungsi praktis untuk memfilter, mengelompokkan, merangkum, dan merapikan data. Setelah data bersih, paket seperti MASS, car, atau ISLR dapat digunakan untuk melakukan analisis statistik lanjutan. Dengan kata lain, data munging dalam R berfungsi sebagai pintu masuk—menjamin data yang akan masuk ke proses komputasi statistik siap pakai—sehingga hasil perhitungan dan interpretasi lebih akurat, efisien, dan dapat direproduksi.

Teknik data munging mencakup serangkaian langkah yang bertujuan mengubah data mentah menjadi format yang bersih, konsisten, dan siap untuk dianalisis secara statistik. Proses ini biasanya dimulai dengan pembersihan data, misalnya memperbaiki nilai hilang, menghapus duplikat, atau menangani outlier. Setelah itu dilakukan transformasi data, seperti mengubah tipe variabel, menormalkan skala, atau membuat variabel baru yang lebih informatif. Pada tahap berikutnya, sering kali diperlukan integrasi data dari berbagai sumber melalui penggabungan tabel, disertai penyaringan dan seleksi untuk mempertahankan hanya variabel atau observasi yang relevan. Teknik lain yang penting adalah reshaping, yaitu mengubah struktur data dari bentuk wide ke long atau sebaliknya agar lebih mudah diolah. Selain itu, data munging juga mencakup rekayasa fitur, seperti mengkodekan variabel kategorikal atau membentuk indikator baru yang mendukung analisis lebih dalam. Seluruh proses ini diakhiri dengan validasi data guna memastikan

konsistensi logis, sehingga hasil komputasi statistik yang diterapkan pada data tersebut dapat diandalkan dan akurat.

Menciptakan variabel baru dalam dataframe

Menciptakan variabel baru dalam sebuah dataframe merupakan salah satu teknik penting dalam data munging karena sering kali analisis statistik membutuhkan informasi tambahan yang tidak tersedia secara langsung pada data mentah. Proses ini dikenal sebagai rekayasa fitur (feature engineering), di mana peneliti menambahkan kolom baru berdasarkan kombinasi, transformasi, atau perhitungan dari variabel yang sudah ada. Misalnya, dalam dataset Titanic, kita dapat membuat variabel baru seperti kategori umur (anak, dewasa, lansia) berdasarkan variabel Age, atau membuat variabel biner seperti FamilyOnBoard yang menunjukkan apakah seorang penumpang bepergian bersama keluarga berdasarkan jumlah SibSp (saudara/istri) dan Parch (orang tua/anak). Variabel tambahan ini dapat memperkaya analisis, membantu model prediktif bekerja lebih baik, dan memberikan wawasan baru tentang pola dalam data.

```
library(titanic)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data("titanic_train")
titanic_new <- titanic_train %>%
  mutate(
    AgeGroup = case_when(
      Age < 18 ~ "Child",
      Age >= 18 & Age < 60 ~ "Adult",
      Age >= 60 ~ "Senior",
      TRUE ~ "Unknown"
    ),
    FamilySize = SibSp + Parch + 1,
    FamilyOnBoard = ifelse(FamilySize > 1, 1, 0)
  )
head(titanic_new[, c("Name", "Age", "AgeGroup", "FamilySize", "FamilyOnBoard"
)], 10)
```

```
##                                                 Name Age AgeGroup Family
Size
## 1                            Braund, Mr. Owen Harris  22    Adult
2
```

```
## 2  Cumings, Mrs. John Bradley (Florence Briggs Thayer)  38    Adult
2
## 3                              Heikkinen, Miss. Laina  26    Adult
1
## 4         Futrelle, Mrs. Jacques Heath (Lily May Peel)  35    Adult
2
## 5                            Allen, Mr. William Henry  35    Adult
1
## 6                                    Moran, Mr. James  NA  Unknown
1
## 7                             McCarthy, Mr. Timothy J  54    Adult
1
## 8                      Palsson, Master. Gosta Leonard   2    Child
5
## 9    Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)  27    Adult
3
## 10                Nasser, Mrs. Nicholas (Adele Achem)  14    Child
2
##    FamilyOnBoard
## 1              1
## 2              1
## 3              0
## 4              1
## 5              0
## 6              0
## 7              0
## 8              1
## 9              1
## 10             1
```

Kode R di atas menunjukkan bagaimana cara menciptakan variabel baru dalam dataframe menggunakan dataset Titanic. Pertama, data Titanic diambil dari paket titanic lalu diproses dengan fungsi $mutate(\ \ )$ dari paket $dplyr$. Di dalam $mutate(\ \ )$, dibuat variabel baru AgeGroup yang mengklasifikasikan penumpang ke dalam kelompok umur: “Child” untuk usia di bawah 18, “Adult” untuk 18–59, “Senior” untuk 60 ke atas, serta “Unknown” jika nilai Age kosong. Selanjutnya, dibuat variabel FamilySize yang dihitung dari jumlah saudara/istri (SibSp) ditambah jumlah orang tua/anak (Parch) lalu ditambah 1 untuk memasukkan penumpang itu sendiri. Dari FamilySize, dibuat variabel biner FamilyOnBoard yang bernilai 1 jika penumpang bepergian dengan keluarga (lebih dari 1 orang) dan 0 jika sendirian. Akhirnya, hasil pengolahan ditampilkan dengan memilih beberapa kolom, yaitu Name, Age, AgeGroup, FamilySize, dan FamilyOnBoard, sehingga kita bisa melihat sepuluh baris pertama data yang sudah diperkaya dengan variabel tambahan.

Subsetting Data

Subsetting data adalah teknik untuk memilih sebagian data dari suatu himpunan berdasarkan kriteria tertentu, baik dalam bentuk baris (observasi) maupun kolom (variabel). Tujuan dari subsetting adalah agar analisis lebih terfokus pada data yang relevan, sehingga

perhitungan menjadi lebih efisien dan hasil interpretasi lebih tepat. Misalnya, dalam analisis dataset Titanic, kita dapat melakukan subsetting untuk melihat hanya penumpang wanita, hanya penumpang kelas tertentu, atau memilih variabel tertentu saja seperti umur, jenis kelamin, dan status selamat. Dengan subsetting, peneliti bisa menyaring informasi yang penting dan mengabaikan bagian data yang tidak dibutuhkan dalam konteks analisis tertentu.

```
data <- titanic_train
subset_data1 <- data %>% select(Sex, Age, Survived)
subset_data2 <- data %>% filter(Sex == "female")
subset_data3 <- data %>% filter(Sex == "female", Pclass == 1)
head(subset_data1)
```

```
##      Sex Age Survived
## 1   male  22        0
## 2 female  38        1
## 3 female  26        1
## 4 female  35        1
## 5   male  35        0
## 6   male  NA        0
```

```
head(subset_data2)
```

```
##   PassengerId Survived Pclass
## 1           2        1      1
## 2           3        1      3
## 3           4        1      1
## 4           9        1      3
## 5          10        1      2
## 6          11        1      3
##                                                 Name    Sex Age SibSp Par
ch
## 1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1
0
## 2                              Heikkinen, Miss. Laina female  26     0
0
## 3        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1
0
## 4   Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg) female  27     0
2
## 5                 Nasser, Mrs. Nicholas (Adele Achem) female  14     1
0
## 6                     Sandstrom, Miss. Marguerite Rut female   4     1
1
##             Ticket    Fare Cabin Embarked
## 1         PC 17599 71.2833   C85        C
## 2 STON/O2. 3101282  7.9250              S
## 3           113803 53.1000  C123        S
## 4           347742 11.1333              S
```

```
## 5           237736 30.0708              C
## 6          PP 9549 16.7000    G6        S
```

```
head(subset_data3)
```

```
##   PassengerId Survived Pclass
## 1           2        1      1
## 2           4        1      1
## 3          12        1      1
## 4          32        1      1
## 5          53        1      1
## 6          62        1      1
##                                                 Name    Sex Age SibSp Par
ch
## 1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1
0
## 2        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1
0
## 3                            Bonnell, Miss. Elizabeth female  58     0
0
## 4      Spencer, Mrs. William Augustus (Marie Eugenie) female  NA     1
0
## 5            Harper, Mrs. Henry Sleeper (Myna Haxtun) female  49     1
0
## 6                                  Icard, Miss. Amelie female  38     0
0
##     Ticket     Fare Cabin Embarked
## 1 PC 17599  71.2833   C85        C
## 2   113803  53.1000  C123        S
## 3   113783  26.5500  C103        S
## 4 PC 17569 146.5208   B78        C
## 5 PC 17572  76.7292   D33        C
## 6   113572  80.0000   B28
```

Kode R di atas menunjukkan bagaimana teknik subsetting data diterapkan pada dataset Titanic. Pertama, paket titanic dimuat untuk mengakses dataset titanic_train, sementara dplyr digunakan untuk memudahkan manipulasi data. Pada langkah pertama (select), dilakukan subsetting kolom dengan hanya mengambil variabel penting seperti jenis kelamin (Sex), umur (Age), dan status kelangsungan hidup (Survived). Langkah kedua menggunakan filter untuk melakukan subsetting baris, yakni memilih hanya penumpang wanita. Kemudian, pada langkah ketiga dilakukan subsetting lebih spesifik dengan memfilter penumpang yang berjenis kelamin wanita sekaligus berasal dari kelas pertama (Pclass == 1). Akhirnya, fungsi head() digunakan untuk menampilkan beberapa baris pertama dari hasil subsetting tersebut. Dengan demikian, kode ini menggambarkan bagaimana subsetting dapat diterapkan untuk mengambil bagian tertentu dari data, baik berupa kolom maupun baris, sesuai dengan tujuan analisis.

Sorting data

Sorting data adalah teknik dasar dalam pengolahan data untuk mengurutkan baris (observasi) berdasarkan satu atau lebih variabel, baik secara menaik (ascending) maupun menurun (descending). Proses ini penting karena dapat membantu peneliti melihat pola tertentu, mengidentifikasi nilai ekstrim, atau menyiapkan data sebelum dilakukan analisis lanjutan. Dalam konteks komputasi statistik, pengurutan data juga memudahkan dalam pembuatan tabel ringkasan, grafik, maupun saat melakukan ranking. Misalnya pada dataset Titanic, kita dapat mengurutkan penumpang berdasarkan umur, kelas tiket, atau status kelangsungan hidup untuk menganalisis karakteristik populasi penumpang.

```
data("Titanic")
titanic_df <- as.data.frame(Titanic)
head(titanic_df)
```

```
##   Class    Sex   Age Survived Freq
## 1   1st   Male Child       No    0
## 2   2nd   Male Child       No    0
## 3   3rd   Male Child       No   35
## 4  Crew   Male Child       No    0
## 5   1st Female Child       No    0
## 6   2nd Female Child       No    0
```

```
titanic_sorted_age <- titanic_df %>%
  arrange(Age)
titanic_sorted_class <- titanic_df %>%
  arrange(desc(Class))
titanic_sorted_combo <- titanic_df %>%
  arrange(Survived, Age)
head(titanic_sorted_age)
```

```
##   Class    Sex   Age Survived Freq
## 1   1st   Male Child       No    0
## 2   2nd   Male Child       No    0
## 3   3rd   Male Child       No   35
## 4  Crew   Male Child       No    0
## 5   1st Female Child       No    0
## 6   2nd Female Child       No    0
```

```
head(titanic_sorted_class)
```

```
##   Class    Sex   Age Survived Freq
## 1  Crew   Male Child       No    0
## 2  Crew Female Child       No    0
## 3  Crew   Male Adult       No  670
## 4  Crew Female Adult       No    3
## 5  Crew   Male Child      Yes    0
## 6  Crew Female Child      Yes    0
```

```
head(titanic_sorted_combo)
```

```
##   Class    Sex   Age Survived Freq
## 1   1st   Male Child       No    0
```

```
## 2   2nd   Male Child       No    0
## 3   3rd   Male Child       No   35
## 4  Crew   Male Child       No    0
## 5   1st Female Child       No    0
## 6   2nd Female Child       No    0
```

Kode di atas menunjukkan bagaimana melakukan sorting data Titanic menggunakan R dan paket dplyr. Pertama, dataset bawaan Titanic yang awalnya berbentuk tabel kontingensi diubah menjadi data frame dengan fungsi as.data.frame() agar lebih mudah diproses. Setelah itu, fungsi head() digunakan untuk melihat isi awal dataset. Proses sorting dilakukan dengan arrange(), misalnya arrange(Age) untuk mengurutkan data berdasarkan umur dari kategori Child ke Adult, arrange(desc(Class)) untuk mengurutkan kelas tiket secara menurun dari First hingga Crew, serta arrange(Survived, Age) untuk mengurutkan terlebih dahulu berdasarkan status kelangsungan hidup (No kemudian Yes), lalu di dalam masing-masing kelompok diurutkan lagi berdasarkan umur. Hasil pengurutan ditampilkan dengan head() untuk melihat enam baris pertama dari setiap versi data yang telah disortir. Dengan cara ini, peneliti dapat dengan cepat mengamati pola dalam data Titanic, misalnya distribusi umur atau kelas tiket dari penumpang yang selamat dan tidak selamat.

Recording Data

Recording Data berarti proses pencatatan dan penyimpanan data hasil observasi atau pengukuran ke dalam bentuk yang terstruktur sehingga dapat dianalisis lebih lanjut. Dalam konteks statistik, data dapat direkam dalam bentuk tabel, spreadsheet, atau data frame pada perangkat lunak seperti R. Pentingnya recording data adalah menjaga agar informasi yang dikumpulkan tetap lengkap, konsisten, dan dapat digunakan kembali. Pada tahap ini biasanya ditentukan variabel apa saja yang akan dicatat (misalnya umur, jenis kelamin, status bertahan hidup), bagaimana format penyimpanannya (angka, teks, kategori), serta bagaimana menangani data yang hilang atau tidak valid. Dengan pencatatan yang baik, analisis statistik maupun visualisasi dapat dilakukan secara efisien.

```r
data("Titanic")
titanic_df <- as.data.frame(Titanic)
head(titanic_df)
```

```
##   Class    Sex   Age Survived Freq
## 1   1st   Male Child       No    0
## 2   2nd   Male Child       No    0
## 3   3rd   Male Child       No   35
## 4  Crew   Male Child       No    0
## 5   1st Female Child       No    0
## 6   2nd Female Child       No    0
```

```r
recorded_data <- titanic_df[, c("Class", "Sex", "Age", "Survived", "Freq")]
head(recorded_data)
```

```
##   Class    Sex   Age Survived Freq
## 1   1st   Male Child       No    0
## 2   2nd   Male Child       No    0
```

```
## 3   3rd   Male Child       No   35
## 4  Crew   Male Child       No    0
## 5   1st Female Child       No    0
## 6   2nd Female Child       No    0
```

```
sum(recorded_data$Freq)
```

```
## [1] 2201
```

Kode R di atas menunjukkan bagaimana proses recording data dilakukan menggunakan dataset Titanic bawaan R. Pertama, fungsi data("Titanic") memanggil dataset Titanic yang tersimpan dalam bentuk tabel kontingensi (array multidimensi). Karena bentuk ini kurang fleksibel untuk analisis, data tersebut kemudian diubah menjadi data frame dengan as.data.frame(Titanic), sehingga setiap baris mewakili kombinasi atribut penumpang (kelas, jenis kelamin, umur, status selamat) beserta frekuensinya. Selanjutnya, fungsi head() digunakan untuk menampilkan sebagian kecil baris agar kita bisa melihat struktur data. Bagian titanic_df[, c("Class", "Sex", "Age", "Survived", "Freq")] mengekstrak variabel utama yang direkam, yaitu kelas, jenis kelamin, umur, status bertahan hidup, dan frekuensi jumlah penumpang. Terakhir, sum(recorded_data$Freq) menghitung total keseluruhan penumpang yang terekam di dataset. Dengan langkah ini, data Titanic berhasil dicatat kembali dalam format yang lebih terstruktur dan siap dipakai untuk analisis statistik lebih lanjut.

Merging Data

Merging data adalah teknik dalam data munging yang digunakan untuk menggabungkan dua atau lebih tabel berdasarkan kunci tertentu agar informasi yang semula terpisah bisa dianalisis secara terpadu. Proses ini sangat penting ketika data berasal dari berbagai sumber atau disimpan dalam beberapa tabel. Misalnya, sebuah tabel bisa berisi data identitas penumpang Titanic, sedangkan tabel lain berisi informasi tiket atau biaya perjalanan. Dengan melakukan merge menggunakan kolom kunci (misalnya PassengerId), kita bisa memperoleh dataset yang lebih lengkap yang menggabungkan atribut dari kedua tabel. Dalam R, penggabungan data dapat dilakukan menggunakan fungsi seperti merge() atau dengan operator join dari paket dplyr (left_join, right_join, inner_join, full_join).

```
data1 <- titanic_train[, c("PassengerId", "Pclass", "Sex", "Age")]
data2 <- titanic_train[, c("PassengerId", "Fare")]
merged_data <- left_join(data1, data2, by = "PassengerId")
head(merged_data)
```

```
##   PassengerId Pclass    Sex Age    Fare
## 1           1      3   male  22  7.2500
## 2           2      1 female  38 71.2833
## 3           3      3 female  26  7.9250
## 4           4      1 female  35 53.1000
## 5           5      3   male  35  8.0500
## 6           6      3   male  NA  8.4583
```

Kode di atas menunjukkan contoh sederhana bagaimana melakukan merging data menggunakan dataset Titanic dalam R. Pertama, kita memanggil pustaka titanic untuk memuat dataset dan dplyr untuk memanfaatkan fungsi join. Dari dataset titanic_train, kita membentuk dua tabel berbeda: data1 yang hanya berisi kolom PassengerId, Pclass, Sex, dan Age, serta data2 yang berisi PassengerId dan Fare. Kedua tabel ini kemudian digabungkan menggunakan fungsi left_join() dengan kunci penghubung PassengerId. Hasil penggabungan tersebut disimpan pada objek merged_data, yang kini memuat informasi gabungan tentang kelas, jenis kelamin, usia, dan tarif penumpang dalam satu tabel. Fungsi head() dipakai untuk menampilkan enam baris pertama, sehingga kita bisa melihat bahwa data dari data1 dan data2 telah berhasil disatukan sesuai dengan identitas penumpangnya.

Reshaping data

Reshaping data adalah proses mengubah bentuk atau struktur data agar lebih sesuai dengan kebutuhan analisis. Dalam praktik analisis data, sering kali kita menemui dataset yang berbentuk wide (setiap variabel atau kategori dalam kolom terpisah) atau long (setiap pengamatan direpresentasikan sebagai baris dengan variabel kategori sebagai penanda). Teknik reshaping memungkinkan peneliti untuk berpindah dari satu bentuk ke bentuk lain, misalnya dari wide ke long untuk mempermudah analisis regresi atau visualisasi, dan dari long ke wide untuk mempermudah interpretasi tabulasi. Di R, fungsi-fungsi dari paket tidyr seperti pivot_longer() dan pivot_wider() sering digunakan untuk melakukan reshaping data.

```r
library(tidyr)
library(dplyr)
data("Titanic")
titanic_df <- as.data.frame(Titanic)
head(titanic_df)
```

```
##   Class    Sex   Age Survived Freq
## 1   1st   Male Child       No    0
## 2   2nd   Male Child       No    0
## 3   3rd   Male Child       No   35
## 4  Crew   Male Child       No    0
## 5   1st Female Child       No    0
## 6   2nd Female Child       No    0
```

```r
titanic_wide <- titanic_df %>%
  pivot_wider(names_from = Survived, values_from = Freq)

head(titanic_wide)
```

```
## # A tibble: 6 × 5
##   Class Sex    Age      No   Yes
##   <fct> <fct>  <fct> <dbl> <dbl>
## 1 1st   Male   Child     0     5
## 2 2nd   Male   Child     0    11
## 3 3rd   Male   Child    35    13
## 4 Crew  Male   Child     0     0
```

```
## 5 1st   Female Child     0     1
## 6 2nd   Female Child     0    13
```

```
titanic_long <- titanic_wide %>%
  pivot_longer(cols = c("No", "Yes"), names_to = "Survived", values_to = "Fre
q")
```

```
head(titanic_long)
```

```
## # A tibble: 6 × 5
##   Class Sex   Age   Survived  Freq
##   <fct> <fct> <fct> <chr>    <dbl>
## 1 1st   Male  Child No           0
## 2 1st   Male  Child Yes          5
## 3 2nd   Male  Child No           0
## 4 2nd   Male  Child Yes         11
## 5 3rd   Male  Child No          35
## 6 3rd   Male  Child Yes         13
```

Kode R di atas menunjukkan bagaimana proses reshaping data dilakukan menggunakan dataset Titanic bawaan R. Pertama, data Titanic yang awalnya berupa objek tabulasi diubah menjadi data frame (titanic_df) sehingga lebih mudah diolah. Data ini secara default berbentuk long format, di mana setiap kombinasi variabel Class, Sex, Age, dan Survived direpresentasikan sebagai baris, dengan kolom Freq menunjukkan jumlah penumpang pada kategori tersebut. Selanjutnya, fungsi pivot_wider() digunakan untuk mengubah data ke dalam wide format dengan menjadikan kategori pada variabel Survived (“Yes” dan “No”) sebagai kolom baru, sehingga nilai pada kolom Freq terbagi ke dalam kolom-kolom tersebut. Hasilnya, kita dapat lebih mudah membandingkan jumlah penumpang yang selamat dan tidak selamat dalam satu baris. Terakhir, fungsi pivot_longer() digunakan untuk mengembalikan data ke bentuk long format semula, dengan menggabungkan kembali kolom “Yes” dan “No” menjadi satu variabel Survived dan nilai frekuensinya tersimpan pada kolom Freq. Dengan demikian, kode ini memperlihatkan fleksibilitas reshaping dalam R yang memungkinkan kita menyesuaikan struktur data sesuai kebutuhan analisis.