

Modul Praktik Statistika Untuk Perencanaan

Penggunaan R dalam Pengolahan Data Kuantitatif

Contents

Pengantar	3
Pengenalan R, RStudio, dan R Markdown	4
0.1 Apa itu R?	4
0.2 Apa itu RStudio?	5
0.3 Apa itu R Markdown?	5
Pengenalan Teknis R	5
0.4 Sintaks Dasar	5
0.5 Paket (<i>Packages</i>)	6
0.6 Fungsi (<i>Functions</i>)	6
0.7 Struktur Data Dasar	6
0.8 Operator Pipa (<code> ></code>)	7
0.9 Manajemen Data (Input/Output)	7
1 Modul-1: Pengolahan Data Terstruktur	8
1.1 Mengimpor set data (<i>dataset</i>) dari Excel	8
1.2 Analisis data terstruktur pada set data	19
1.3 Mengolah set data	20
1.4 Soal Latihan	28
2 Modul-2: Analisis Statistika Deskriptif	28
2.1 Persiapan	29
2.2 Analisis Statistika Deskriptif	41
2.3 Soal Latihan	45
3 Modul-3: Visualisasi Data Kuantitatif	46
3.1 Visualisasi Data dengan <code>ggplot2</code>	46
3.2 Soal Praktikum	66

4	Modul-4: Distribusi Sampling dan Interval Kepercayaan	66
4.1	Pendahuluan	66
4.2	Perangkat Lunak dan Pustaka (Libraries)	66
4.3	Memuat dan Mempersiapkan Data	66
4.4	Distribusi <i>Sampling</i> dan <i>Standard Error</i>	69
4.5	Estimasi Parameter-1: Interval Kepercayaan Rata-rata	72
4.6	Estimasi Parameter-2: Interval Kepercayaan Proporsi	74
4.7	Soal Latihan	76
5	Modul-5: Uji Hipotesis	76
5.1	Pendahuluan	76
5.2	Perangkat Lunak dan Pustaka (Libraries)	76
5.3	Memuat Data	77
5.4	Perkenalan Markdown: LaTeX	77
5.5	Uji Hipotesis Satu Populasi	78
5.6	Uji Hipotesis 2 Populasi Independen	85
5.7	Uji Hipotesis 2 Populasi Dependen	92
5.8	Soal Latihan	92
6	Modul-6: Analisis Hubungan Korelatif: Korelasi Variabel Nominal dan Ordinal	92
6.1	Pendahuluan	92
6.2	Pustaka (<i>Libraries</i>) yang Diperlukan dan Memuat Data	93
6.3	Pembuatan Tabel Silang	96
6.4	Analisis Korelasi Antarvariabel Nominal	98
6.5	Analisis Korelasi Antarvariabel Ordinal	101
7	Modul-7: Analisis Hubungan Korelatif: Korelasi Variabel Metrik	102
7.1	Pendahuluan	102
7.2	Pustaka (<i>Libraries</i>) yang Diperlukan dan Memuat Data	102
7.3	Pola Hubungan Data	103
7.4	6.3. Pola Hubungan Data	103
7.5	Analisis Korelasi Spearman's ρ	104
7.6	Analisis Korelasi Pearson's r	105

8	Modul-8: Analisis Hubungan Kausalitas: Regresi Linear Sederhana dan Berganda	107
8.1	Pendahuluan	107
8.2	Penjelasan Kasus	108
8.3	Memuat Pustaka (<i>Libraries</i>) yang Diperlukan	108
8.4	Memuat Dataset	109
8.5	Pola Hubungan Data	109
8.6	Model Regresi Linear Sederhana	112
8.7	Model Regresi Linear Berganda	119
9	Modul-9: Analisis Hubungan Multivariat Interdependensi - Analisis Faktor dan PCA	122
9.1	Pendahuluan	122
9.2	Studi Kasus	122
9.3	Memuat Pustaka (<i>Libraries</i>)	122
9.4	Asumsi Awal	122
9.5	Mempersiapkan Data	151
9.6	Uji Kelayakan Data (Asumsi)	152
9.7	Mengekstrak Dimensi Baru	153
9.8	Melakukan Analisis Faktor dan Rotasi	154
9.9	Interpretasi dan Penamaan Dimensi	157
10	Referensi	157

Pengantar

Selamat datang di modul daring Praktikum Statistika untuk Perencanaan. Modul ini digunakan untuk mendampingi kalian mengikuti Praktikum dalam mata kuliah Statistika untuk Perencanaan, mata kuliah di Program Studi Perencanaan Wilayah dan Kota, Institut Teknologi Sumatera.

Modul praktikum ini akan mendampingi Anda mempelajari pengolahan statistik menggunakan bahasa pemrograman R. Untuk itu, modul ini terbagi menjadi **9 modul** yang disusun secara bertahap, dimulai dari pengenalan dasar hingga analisis multivariat:

Modul	Topik	Deskripsi Singkat
1	Pengolahan Data Terstruktur	Dasar <i>data wrangling</i> : Impor, seleksi, standardisasi, penggabungan, dan ekspor data/dataset
2	Statistika Deskriptif	Perhitungan proporsi, mean, median, standar deviasi, dan IQV
3	Visualisasi Data	Pembuatan grafik dengan <code>ggplot2</code> : bar chart, histogram, boxplot, scatter plot
4	Distribusi Sampling & Interval Kepercayaan	Simulasi sampling, standard error, dan estimasi parameter
5	Uji Hipotesis	Uji hipotesis 1 dan 2 populasi untuk rata-rata dan proporsi
6	Korelasi Nominal & Ordinal	Koefisien V Cramer, Lambda, dan Gamma
7	Korelasi Metrik	Koefisien Spearman's dan Pearson's r
8	Regresi Linear	Regresi linear sederhana dan berganda

Modul	Topik	Deskripsi Singkat
9	Analisis Komponen Prinsip	PCA dan Analisis Faktor

Setiap modul dilengkapi dengan **capaian pembelajaran**, **contoh kode**, dan **soal latihan** yang harus dikerjakan.

Pengenalan R, RStudio, dan R Markdown

Selamat datang di dunia pengolahan data yang lebih canggih! Dalam praktikum ini, kalian akan berkenalan dengan teknik pengolahan dan analisis data yang lebih canggih, yakni dengan menggunakan bahasa pemrograman bernama R. R adalah bahasa pemrograman yang dibuat khusus untuk keperluan analisis statistik.

0.1 Apa itu R?

R adalah lingkungan perangkat lunak (*software environment*) dan bahasa pemrograman yang digunakan untuk komputasi statistik dan grafik. R dikembangkan oleh Ross Ihaka dan Robert Gentleman di Universitas Auckland, Selandia Baru, pada awal tahun 1990-an sebagai evolusi dari bahasa S yang sebelumnya dikembangkan di Bell Laboratories [Peng, 2026]. R bersifat *open-source* (berlisensi GNU GPL), yang berarti dapat digunakan secara gratis dan dikembangkan secara kolaboratif oleh komunitas global [Wikipedia contributors, 2026].

Keunggulan utama R dalam konteks perencanaan dan pemerintahan meliputi:

1. **Ekosistem Paket yang Luas:** R memiliki puluhan ribu paket (*packages*) di CRAN untuk berbagai keperluan, mulai dari manipulasi data, pemodelan statistik, hingga analisis geospasial menggunakan `sf` dan `terra` [UVA Library, 2026, R Spatial, 2026].
2. **Visualisasi Data:** Dengan paket seperti `ggplot2` dan `Shiny`, R memungkinkan pembuatan grafik berkualitas tinggi dan dashboard interaktif yang informatif [Posit, 2026b].
3. **Dukungan Kebijakan Berbasis Data:** R mendukung implementasi kebijakan “Satu Data Indonesia” dengan kemampuan standardisasi, interoperabilitas, dan transparansi metode analisis yang dapat diaudit [ResearchGate, 2026a]. R memfasilitasi proses ETL (*Extract-Transform-Load*) otomatis untuk membersihkan data dari berbagai instansi [Romanian Statistical Review, 2026].

Dalam bidang perencanaan wilayah, R digunakan untuk:

- **Analisis Geospasial:** Mengelola data vektor dan raster untuk pemetaan dan pemodelan lingkungan [R Spatial, 2026].
- **Perencanaan Transportasi:** Menggunakan paket seperti `stplanr` untuk analisis jaringan, rute, dan pemodelan transportasi berkelanjutan [CRAN, 2026, rOpenSci, 2026].
- **Simulasi Kebijakan:** Melakukan simulasi dampak kebijakan publik, seperti perubahan penggunaan lahan atau intervensi sosial, sebelum diterapkan secara nyata [ResearchGate, 2026b].

0.2 Apa itu RStudio?

RStudio (yang kini bertransformasi menjadi **Posit**) adalah *Integrated Development Environment* (IDE) untuk R. Diluncurkan pertama kali pada tahun 2009, RStudio menyediakan antarmuka visual yang lebih ramah pengguna dibandingkan antarmuka baris perintah (*command line*) asli R [Posit, 2026a]. Fitur-fitur seperti penyorotan sintaksis (*syntax highlighting*), *auto-completion*, dan manajemen *workspace* membuat pengguna baru lebih mudah mempelajari dan menggunakan R. Transformasi menjadi Posit mencerminkan komitmen terhadap ekosistem *data science* modern yang lebih luas, termasuk dukungan untuk Python [Irorere, 2026].

0.3 Apa itu R Markdown?

R Markdown adalah format dokumen yang memungkinkan pengguna untuk menggabungkan kode R, hasil analisis (seperti tabel dan grafik), serta narasi teks dalam satu dokumen yang dinamis. Prinsip ini mendukung *reproducible data science*, di mana seluruh proses dari data mentah hingga laporan akhir dapat dilacak dan diulang. R Markdown sangat berguna untuk otomatisasi pelaporan di instansi pemerintahan, memungkinkan pembuatan laporan rutin yang efisien dan konsisten.

Pengenalan Teknis R

Sebelum memulai praktikum dengan set data sesungguhnya, ada baiknya kita memahami beberapa konsep teknis dasar dalam bahasa pemrograman R. Bagian ini merangkum keterampilan-keterampilan dasar yang akan kalian temui tersebar di modul-modul praktikum.

0.4 Sintaks Dasar

0.4.1 Komentar

Di R, segala sesuatu yang ditulis setelah tanda pagar (#) dianggap sebagai komentar dan tidak akan dieksekusi oleh komputer. Komentar berguna untuk memberikan catatan pada kode kita.

```
# Ini adalah komentar  
1 + 1 # Kode ini akan menghitung 1 + 1
```

0.4.2 Assignment (Pemberian Nilai)

Untuk menyimpan nilai ke dalam variabel/objek, kita menggunakan tanda panah kiri (<-). Kalian juga bisa menggunakan tanda sama dengan (=), namun <- adalah standar yang lebih disarankan di komunitas R.

```
x <- 10  
nama <- "Budi"
```

0.4.3 Case Sensitivity

R membedakan huruf besar dan huruf kecil (*case sensitive*). Variabel **Data** dianggap berbeda dengan **data**.

0.5 Paket (*Packages*)

R memiliki ribuan paket tambahan yang menyediakan fungsi-fungsi khusus.

- `install.packages("nama_paket")`: Digunakan untuk mengunduh dan menginstal paket dari internet ke komputer kita. Ini cukup dilakukan satu kali saja.
- `library(nama_paket)`: Digunakan untuk memuat paket yang sudah terinstal ke dalam sesi kerja kita agar isinya bisa digunakan. Ini harus dilakukan setiap kali kita membuka RStudio baru.

Paket utama yang akan sering kita gunakan adalah:

- **tidyverse**: Kumpulan paket untuk pengolahan data modern.
- **openxlsx**: Untuk membaca dan menulis file Excel.

0.6 Fungsi (*Functions*)

Fungsi adalah serangkaian instruksi yang dibungkus menjadi satu perintah untuk melakukan tugas tertentu. R memiliki banyak fungsi bawaan, dan kita juga bisa membuat fungsi sendiri.

0.6.1 Menggunakan Fungsi

Untuk menggunakan fungsi, kita memanggil namanya diikuti dengan tanda kurung (). Di dalam tanda kurung, kita bisa memasukkan **argumen** (input).

```
# Contoh fungsi bawaan: mean (rata-rata)
nilai <- c(10, 20, 30)
rata_rata <- mean(nilai) # Hasil: 20
```

0.6.2 Membuat Fungsi Sendiri

Kita dapat membuat fungsi kustom menggunakan perintah `function()`. Struktur dasarnya adalah:

```
nama_fungsi <- function(input) {
  # Proses yang dilakukan
  hasil <- input * 2
  return(hasil)
}
```

Ini akan sangat berguna ketika kita ingin melakukan perhitungan yang sama berulang kali, seperti pada perhitungan IQV di Modul 2.

0.7 Struktur Data Dasar

0.7.1 Vektor

Vektor adalah struktur data paling dasar di R yang menampung deretan nilai dengan tipe yang sama. Kita membuatnya dengan fungsi `c()` (*combine*).

```
angka <- c(1, 2, 3, 4, 5)
huruf <- c("a", "b", "c")
```

0.7.2 Tipe Data

Saat mengolah data, kalian akan sering menemukan singkatan tipe data berikut:

- **dbl (Double)**: Angka numerik (bisa desimal).
- **chr (Character)**: Teks atau *string*. Ditandai dengan tanda kutip.
- **fact (Factor)**: Data kategorik (bisa nominal atau ordinal). Factor menyimpan nilai sebagai angka di belakang layar namun menampilkan label teks, yang berguna untuk analisis statistik dan plot.

0.7.3 Data Frame dan Tibble

Data dalam R biasanya disimpan dalam bentuk tabel yang disebut *Data Frame*. Di era modern (tidyverse), kita sering menggunakan versi yang lebih canggih disebut **Tibble**. Strukturnya mirip spreadsheet: baris mewakili observasi, kolom mewakili variabel.

0.8 Operator Pipa (|>)

Dalam **tidyverse**, kita sering menggunakan operator pipa (|>) untuk merangkai beberapa perintah sekaligus. Bayangkan ini sebagai penyambung proses:

```
# Tanpa pipa
hasil <- fungsi_2(fungsi_1(data))

# Dengan pipa (lebih mudah dibaca)
data |>
  fungsi_1() |>
  fungsi_2()
```

Artinya: “Ambil data, lalu `fungsi_1`, kemudian hasilnya dimasukkan ke `fungsi_2`.”

0.9 Manajemen Data (Input/Output)

0.9.1 Membaca Data

Kita sering menggunakan `read.xlsx()` dari paket **openxlsx** untuk membaca file Excel.

```
data <- read.xlsx("folder/nama_file.xlsx", sheet = "nama_sheet")
```

0.9.2 Menyimpan Data

Setelah diolah, data bisa disimpan kembali ke Excel atau CSV.

```
write.xlsx(data, "folder/nama_file_baru.xlsx")
write.csv2(data, "folder/nama_file_baru.csv") # Format CSV Excel Indonesia (pemisah titik koma)
```

1 Modul-1: Pengolahan Data Terstruktur

Setelah mempelajari modul ini Anda diharapkan dapat:

1. menganalisis variabel dan objek dalam sebuah format data terstruktur **STP-1.3**
2. mengelola data terstruktur dengan tepat menggunakan perangkat komputer **STP-1.4**

Kita akan mempelajari cara pengolahan data terstruktur atau *dataset*/set data dari hasil kuesioner kepada mahasiswa-mahasiswa Universitas Islam Negeri Raden Intan Lampung (UINRIL) dan Universitas Bandar Lampung (UBL) tentang pola perjalanan mereka.

Dalam R, pengolahan data terstruktur dipermudah dengan adanya paket **tidyverse**. Paket ini menyertakan paket-paket lain yang mempunyai kegunaan unik dalam pengolahan data terstruktur.

```
library(tidyverse)
```

Seperti yang kalian lihat, paket **tidyverse** menyertakan paket-paket berikut yang mempunyai kegunaan masing-masing:

- **dplyr**: menyediakan serangkaian fungsi inti untuk memanipulasi dan mentransformasi data secara efisien.
- **forcats**: memudahkan pekerjaan dengan data kategorikal atau yang dalam R disebut juga ‘factors’.
- **ggplot2**: untuk membuat grafik yang ciamik.
- **lubridate**: mempermudah pekerjaan terkait format tanggal
- **purrr**: meningkatkan kemampuan pemrograman fungsi dan vektor
- **readr**: untuk membaca data tabular seperti *.csv*
- **stringr**: menyederhanakan operasi teks/*string*
- **tibble**: versi modern dari *data.frame* yang lebih rapi dan mudah digunakan.
- **tidyr**: untuk merapikan data dengan mengubah format atau struktur tata letaknya.

Selain paket **tidyverse**, kita juga akan menggunakan paket **openxlsx** untuk mengoperasikan *file* Excel.

```
library(openxlsx)
```

1.1 Mengimpor set data (*dataset*) dari Excel

Untuk membuat set data di lingkungan kerja R kita, gunakan fungsi **read.xlsx()** seperti berikut. Fungsi **read.xlsx()** mempunyai atribut utama berupa *file path* lokasi file yang ditempatkan di dalam tanda kurung.

```
# Memuat dataset dari file Excel "Data Gabungan.xlsx" yang terletak di folder yang sama dengan file R M
data <- read.xlsx("datasets/Data Praktikum 01.xlsx")

# Menampilkan dataset
data
```

##	Kampus_PT	Nomor.urut	Jenis.Kelamin	Umur	Fakultas	
## 1	UINRIL	1	Laki-laki	22	Syariah	Hukum K
## 2	UINRIL	2	Laki-laki	25	Syariah	Hukum K
## 3	UINRIL	3	Perempuan	24	Tarbiyah dan Keguruan	Pendidikan Agama

## 4	UINRIL	4	Laki-laki	19	Dakwah dan Ilmu Komunikasi	Komunikasi dan Penyiaran
## 5	UINRIL	5	Laki-laki	23	Tarbiyah dan Keguruan	Manajemen Pendidikan
## 6	UINRIL	6	Laki-laki	21	Adab Ilmu Perpustakaan dan Informas	
## 7	UINRIL	7	Laki-laki	22	Dakwah dan Ilmu Komunikasi	Komunikasi dan Penyiaran
## 8	UINRIL	8	Laki-laki	21	Dakwah dan Ilmu Komunikasi	Pengembangan Masyarakat
## 9	UINRIL	9	Perempuan	21	Dakwah dan Ilmu Komunikasi	Komunikasi dan Penyiaran
## 10	UINRIL	10	Perempuan	21	Tarbiyah dan Keguruan	Pendidikan Agama
## 11	UINRIL	11	Perempuan	19	Ekonomi dan Bisnis Islam	Manajemen Bisnis S
## 12	UINRIL	12	Perempuan	19	Dakwah dan Ilmu Komunikasi	Bimbingan Konseling
## 13	UINRIL	13	Perempuan	19	Ekonomi dan Bisnis Islam	Manajemen Bisnis S
## 14	UINRIL	14	Perempuan	19	Ekonomi dan Bisnis Islam	Manajemen Bisnis S
## 15	UINRIL	15	Laki-laki	19	Dakwah dan Ilmu Komunikasi	Komunikasi dan Penyiaran
## 16	UINRIL	16	Perempuan	20	Syariah	Hukum Tata
## 17	UINRIL	17	Perempuan	19	Tarbiyah dan Keguruan	Pendidikan Bahasa I
## 18	UINRIL	18	Laki-laki	21	Ekonomi dan Bisnis Islam	Manajemen Bisnis S
## 19	UINRIL	19	Perempuan	20	Ekonomi dan Bisnis Islam	Manajemen Bisnis S
## 20	UINRIL	20	Laki-laki	20	Ekonomi dan Bisnis Islam	Manajemen Bisnis S
## 21	UINRIL	21	Perempuan	20	Tarbiyah dan Keguruan	Pendidikan Agama
## 22	UINRIL	22	Perempuan	20	Ekonomi dan Bisnis Islam	Manajemen Bisnis S
## 23	UINRIL	23	Perempuan	20	Syariah	Hukum Ke
## 24	UINRIL	24	Laki-laki	19	Tarbiyah dan Keguruan	Pendidikan Agama
## 25	UINRIL	25	Laki-laki	20	Ekonomi dan Bisnis Islam	Manajemen Bisnis S
## 26	UINRIL	26	Laki-laki	22	Tarbiyah dan Keguruan	Pendidikan Anak Us
## 27	UINRIL	27	Perempuan	22	Syariah	Hukum Ke
## 28	UINRIL	28	Laki-laki	22	Ekonomi dan Bisnis Islam	Ekonomi S
## 29	UINRIL	29	Perempuan	21	Ekonomi dan Bisnis Islam	Ekonomi S
## 30	UINRIL	30	Perempuan	20	Ekonomi dan Bisnis Islam	Manajemen Bisnis S
## 31	UINRIL	31	Laki-laki	21	Tarbiyah dan Keguruan	Pendidikan Mat
## 32	UINRIL	32	Laki-laki	21	Syariah	Hukum Ke
## 33	UINRIL	33	Laki-laki	20	Ekonomi dan Bisnis Islam	Manajemen Bisnis S
## 34	UINRIL	34	Perempuan	21	Ushuluddin dan Studi Agama	Psikologi
## 35	UINRIL	35	Perempuan	20	Ushuluddin dan Studi Agama	Psikologi
## 36	UINRIL	36	Perempuan	21	Tarbiyah dan Keguruan	Pendidikan Agama
## 37	UINRIL	37	Laki-laki	24	Syariah	Hukum I
## 38	UINRIL	38	Laki-laki	22	Tarbiyah dan Keguruan	Manajemen Pendidikan
## 39	UINRIL	39	Laki-laki	21	Tarbiyah dan Keguruan	Sistem In
## 40	UINRIL	40	Perempuan	24	Tarbiyah dan Keguruan	Bimbingan dan Kor
##	Uang.Saku kepemilikan.mobil kepemilikan.motor kepemilikan.sepedan					kendaraan.utama
## 1	1Jt - 2 Jt	0	1	0	Sepeda Motor Pribad	
## 2	< 1 Jt	1	4	0	Sepeda Motor Pribad	
## 3	< 1 Jt	0	0	0	Berjalan Kaki	
## 4	< 1 Jt	0	1	0	Sepeda Motor Pribad	
## 5	< 1 Jt	0	1	0	Sepeda Motor Pribad	
## 6	< 1 Jt	0	1	0	Sepeda Motor Pribad	
## 7	< 1 Jt	1	1	0	Sepeda Motor Pribad	
## 8	< 1 Jt	1	4	0	Sepeda Motor Pribad	
## 9	1Jt - 2 Jt	1	4	1	Sepeda Motor Pribad	
## 10	< 1 Jt	0	1	0	Sepeda Motor Pribad	
## 11	1Jt - 2 Jt	0	0	0	Kendaraan Bermotor (menumpang)	
## 12	< 1 Jt	0	1	0	Sepeda Motor Pribad	
## 13	1Jt - 2 Jt	0	3	1	Sepeda Motor Pribad	
## 14	1Jt - 2 Jt	0	1	0	Sepeda Motor Pribad	
## 15	< 1 Jt	0	0	0	Kendaraan Bermotor (menumpang)	
## 16	< 1 Jt	0	0	0	Berjalan Kaki	

## 17	< 1 Jt	0	1	0	Sepeda Motor Pribadi
## 18	< 1 Jt	0	2	0	Sepeda Motor Pribadi
## 19	< 1 Jt	0	2	0	Sepeda Motor Pribadi
## 20	< 1 Jt	0	1	1	Sepeda Motor Pribadi
## 21	< 1 Jt	0	0	0	Berjalan Kaki
## 22	< 1 Jt	0	0	0	Kendaraan Bermotor (menumpang)
## 23	2 Jt - 3 Jt	0	1	0	Sepeda Motor Pribadi
## 24	< 1 Jt	0	0	1	Sepeda Motor Pribadi
## 25	< 1 Jt	0	1	0	Sepeda Motor Pribadi
## 26	1Jt - 2 Jt	1	3	1	Sepeda Motor Pribadi
## 27	1Jt - 2 Jt	1	2	0	Sepeda Motor Pribadi
## 28	< 1 Jt	1	2	3	Sepeda Motor Pribadi
## 29	< 1 Jt	0	0	0	Berjalan Kaki
## 30	< 1 Jt	0	1	0	Sepeda Motor Pribadi
## 31	1Jt - 2 Jt	0	1	0	Sepeda Motor Pribadi
## 32	< 1 Jt	0	1	0	Sepeda Motor Pribadi
## 33	< 1 Jt	1	2	4	Sepeda Motor Pribadi
## 34	1Jt - 2 Jt	0	1	0	Sepeda Motor Pribadi
## 35	< 1 Jt	0	1	0	Sepeda Motor Pribadi
## 36	< 1 Jt	0	0	0	Transportasi Online
## 37	1Jt - 2 Jt	0	0	0	Berjalan Kaki
## 38	1Jt - 2 Jt	1	2	0	Sepeda Motor Pribadi
## 39	< 1 Jt	0	3	1	Sepeda Motor Pribadi
## 40	< 1 Jt	0	2	0	Sepeda Motor Pribadi
##	jenis.tempat.tinggal				nama.jalan
## 1	Rumah Bersama Saudara				
## 2	Kos Sendiri				jl. n
## 3	Kos Sendiri				
## 4	Kos Sendiri				Jl.pulau pisan
## 5	Rumah Mengontrak Bersama-sama				
## 6	Rumah Pribadi/Rumah Keluarga				Jln Nunyai Gg. Hi. Ismai
## 7	Rumah Pribadi/Rumah Keluarga				Jalan Alimudin Umar
## 8	Rumah Pribadi/Rumah Keluarga				Jl. Kicau, n
## 9	Kos Sendiri				Jl. s
## 10	Rumah Pribadi/Rumah Keluarga				Jal
## 11	Kos Bersama-sama				jalan terusan
## 12	Kos Sendiri				jl. terusan bawean 2 rt
## 13	Rumah Mengontrak Pribadi				j
## 14	Rumah Pribadi/Rumah Keluarga				Jln
## 15	Kos Bersama-sama				Jl. Pulau Legund
## 16	Kos Sendiri				
## 17	Rumah Mengontrak Bersama-sama				Jl. Pandawa 4,sukarame
## 18	Rumah Pribadi/Rumah Keluarga				Jl. Cumi-cumi No. 34 Gg. Cempaka, Teluk
## 19	Rumah Pribadi/Rumah Keluarga				Jl.
## 20	Rumah Bersama Saudara				
## 21	Kos Bersama-sama				
## 22	Kos Bersama-sama				
## 23	Kos Sendiri				way
## 24	Kos Sendiri				
## 25	Rumah Pribadi/Rumah Keluarga				Jalan airan ra
## 26	Rumah Pribadi/Rumah Keluarga				perumahan bukit alam permai
## 27	Rumah Pribadi/Rumah Keluarga				Perum Bukit Bilabong Jaya, Kec.Langkapura, Kel. Bilabong Jaya, Kota
## 28	Rumah Pribadi/Rumah Keluarga				Jl. Padjajaran gg boy
## 29	Kos Sendiri				

## 30	Rumah Mengontrak Bersama-sama	Jl. Endro Suratmin GG H20, bedeng asrama		
## 31	Rumah Bersama Saudara	Jl. Sa		
## 32	Kos Sendiri	J		
## 33	Rumah Pribadi/Rumah Keluarga	J		
## 34	Rumah Pribadi/Rumah Keluarga	jln		
## 35	Kos Sendiri	jln nakul		
## 36	Rumah Pribadi/Rumah Keluarga			
## 37	Rumah Mengontrak Bersama-sama	Jl. Pulau Tegal No.13, Way Dadi, Kec. Sukarame, Kota Bandar Lampung		
## 38	Rumah Pribadi/Rumah Keluarga			
## 39	Rumah Pribadi/Rumah Keluarga			
## 40	Rumah Pribadi/Rumah Keluarga	Perum. Karunia indah blok o no.17, Sukabumi		
##	alasan.pemilihan.lokasi.tempat.tinggal	jarak.(km)	biaya.dalam.ribu2	Jumlah.Perjalanan.Senin
## 1	Lain-lain	19.27	100	3
## 2	Dekat dengan kampus	0.58	80	3
## 3	Dekat dengan kampus	0.56	0	3
## 4	Dekat dengan kampus	1.05	50	0
## 5	Dekat dengan kampus	1.69	50	4
## 6	Dekat dengan kampus	7.91	50	0
## 7	Bersama keluarga/saudara/teman	2.58	35	3
## 8	Bersama keluarga/saudara/teman	2.32	50	4
## 9	Dekat dengan kampus	0.77	40	2
## 10	Bersama keluarga/saudara/teman	6.56	50	3
## 11	Dekat dengan kampus	0.63	0	4
## 12	Dekat dengan kampus	0.53	50	2
## 13	Biaya tempat tinggal murah	0.91	50	3
## 14	Bersama keluarga/saudara/teman	2.52	40	2
## 15	Biaya tempat tinggal murah	1.31	0	3
## 16	Dekat dengan kampus	0.83	0	2
## 17	Lain-lain	0.95	30	2
## 18	Bersama keluarga/saudara/teman	7.67	50	2
## 19	Lingkungan aman dan nyaman	4.34	60	3
## 20	Biaya tempat tinggal murah	4.46	15	2
## 21	Dekat dengan kampus	0.71	0	0
## 22	Dekat dengan kampus	0.92	0	2
## 23	Lingkungan aman dan nyaman	1.37	40	0
## 24	Biaya tempat tinggal murah	0.56	15	3
## 25	Bersama keluarga/saudara/teman	3.02	37	3
## 26	Bersama keluarga/saudara/teman	8.56	25	2
## 27	Bersama keluarga/saudara/teman	8.49	60	3
## 28	Lingkungan aman dan nyaman	4.77	60	1
## 29	Dekat dengan kampus	0.34	0	2
## 30	Lingkungan aman dan nyaman	2.40	15	2
## 31	Bersama keluarga/saudara/teman	4.30	70	3
## 32	Dekat dengan kampus	0.27	25	2
## 33	Bersama keluarga/saudara/teman	2.45	70	2
## 34	Bersama keluarga/saudara/teman	26.14	70	1
## 35	Dekat dengan kampus	1.27	70	4
## 36	Lingkungan aman dan nyaman	9.09	60	0
## 37	Dekat dengan kampus	1.39	0	5
## 38	Bersama keluarga/saudara/teman	0.63	65	2
## 39	Lingkungan aman dan nyaman	4.56	50	5
## 40	Dekat dengan kampus	2.06	50	2
##	Jumlah.Perjalanan.Rabu	Jumlah.Perjalanan.Kamis	Jumlah.Perjalanan.Jumat	Jumlah.Perjalanan.Sabtu
## 1	0	0	0	0

```
## 2          0          3          0          0
## 3          3          0          0          0
## 4          0          2          2          0
## 5          0          0          3          2
## 6          2          0          2          2
## 7          3          0          2          2
## 8          4          4          4          2
## 9          3          0          0          2
## 10         2          2          0          0
## 11         4          2          2          2
## 12         2          2          2          2
## 13         2          2          2          0
## 14         2          2          2          0
## 15         2          0          2          0
## 16         2          3          0          0
## 17         2          2          2          0
## 18         0          2          2          2
## 19         3          2          0          0
## 20         2          2          2          2
## 21         2          2          2          0
## 22         2          2          2          0
## 23         2          2          2          0
## 24         3          3          2          0
## 25         3          2          2          0
## 26         0          2          2          2
## 27         2          3          3          2
## 28         2          2          3          2
## 29         0          2          0          0
## 30         4          4          0          0
## 31         3          3          4          2
## 32         2          2          2          2
## 33         2          2          2          0
## 34         2          2          1          0
## 35         4          4          1          0
## 36         0          0          0          0
## 37         5          5          5          5
## 38         3          3          2          2
## 39         5          2          2          4
## 40         4          2          2          2
## [ reached 'max' / getOption("max.print") -- omitted 360 rows ]
```

Fungsi tersebut akan mengimpor seluruh isi dari file Excel “Data Gabungan.xlsx” pada *sheet* pertama saja.

Apabila kita ingin mengimpor set data dari *sheet* lainnya, kita harus menyertakan detail nama *sheet* yang kita ingin impor pada atribut “*sheet*”.

Dataset yang kita impor akan menjadi objek dengan jenis *tibble*.

Penting

Jangan mengimpor data dari file Excel yang sedang terbuka karena akan menghasilkan pesan error.

```
# Menyimpan file sebagai variabel tersendiri sehingga lebih fleksibel
file_dibaca <- "datasets/Data Praktikum 01.xlsx" # nama variabel bukan 'file' saja karena sudah ada fun.

# Mengimpor data dari file_dibaca dan menyimpannya sebagai variabel
```

```
data_ubl <- read.xlsx(file_dibaca, sheet = "DataUtama_mhsUBL")
```

```
# Melihat data tersebut
```

```
data_ubl
```

##	Timestamp	Kampus_PT	Nomor.urut	Jenis.Kelamin	Umur	Fakultas	
## 1	45413.83	UBL	1	Perempuan	21	Fakultas Ilmu Sosial dan Politik	Ilmu Komun
## 2	45413.83	UBL	2	Laki-Laki	20	Fakultas Hukum	Ilmu K
## 3	45413.85	UBL	3	Laki-Laki	21	Fakultas Ekonomi dan Bisnis	Manaj
## 4	45413.85	UBL	4	Laki-Laki	22	Fakultas Ekonomi dan Bisnis	Akun
## 5	45413.88	UBL	5	Laki-Laki	21	Fakultas Ekonomi dan Bisnis	Manaj
## 6	45413.92	UBL	6	Perempuan	21	Fakultas Ekonomi dan Bisnis	Akun
## 7	45413.93	UBL	7	Perempuan	22	Fakultas Ilmu Sosial dan Politik	Administrasi P
## 8	45413.93	UBL	8	Perempuan	22	Fakultas Ilmu Sosial dan Politik	Administrasi P
## 9	45413.93	UBL	9	Laki-Laki	22	Fakultas Ekonomi dan Bisnis	Manaj
## 10	45413.94	UBL	10	Perempuan	22	Fakultas Ilmu Sosial dan Politik	Administrasi P
## 11	45414.00	UBL	11	Laki-Laki	21	Fakultas Hukum	Ilmu K
## 12	45414.01	UBL	12	Laki-Laki	22	Fakultas Hukum	Ilmu K
## 13	45414.01	UBL	13	Laki-Laki	22	Fakultas Hukum	Ilmu K
## 14	45414.02	UBL	14	Perempuan	22	Fakultas Ilmu Sosial dan Politik	Administrasi P
## 15	45414.22	UBL	15	Laki-Laki	19	Fakultas Ekonomi dan Bisnis	Administrasi B
## 16	45414.30	UBL	16	Perempuan	22	Fakultas Ekonomi dan Bisnis	Manaj
## 17	45414.31	UBL	17	Perempuan	23	Fakultas Ekonomi dan Bisnis	Administrasi B
## 18	45414.38	UBL	18	Perempuan	22	Fakultas Ekonomi dan Bisnis	Akun
## 19	45414.39	UBL	19	Laki-Laki	23	Fakultas Hukum	Ilmu K
## 20	45414.44	UBL	20	Perempuan	22	Fakultas Ekonomi dan Bisnis	Manaj
## 21	45414.46	UBL	21	Laki-Laki	20	Fakultas Teknik	Teknik S
## 22	45414.46	UBL	22	Laki-Laki	19	Fakultas Ilmu Komputer	Sistem Infor
Semester 4)							
## 23	45414.46	UBL	23	Laki-Laki	21	Fakultas Ilmu Komputer	Sistem Infor
Semester 4)							
## 24	45414.48	UBL	24	Laki-Laki	22	Fakultas Ekonomi dan Bisnis	Manaj
## 25	45414.48	UBL	25	Laki-Laki	18	Fakultas Ekonomi dan Bisnis	Manaj
Semester 2)							
## 26	45414.53	UBL	26	Perempuan	20	Fakultas Ekonomi dan Bisnis	Manaj
Semester 4)							
## 27	45414.57	UBL	27	Perempuan	22	Fakultas Ekonomi dan Bisnis	Manaj
## 28	45414.59	UBL	28	Laki-Laki	21	Fakultas Ekonomi dan Bisnis	Administrasi B
## 29	45414.60	UBL	29	Laki-Laki	20	Fakultas Ekonomi dan Bisnis	Manaj
Semester 4)							
## 30	45414.62	UBL	30	Laki-Laki	20	Fakultas Hukum	Ilmu K
## 31	45414.63	UBL	31	Perempuan	22	Fakultas Teknik	Arsite
## 32	45414.63	UBL	32	Laki-Laki	21	Fakultas Teknik	Teknik S
## 33	45414.65	UBL	33	Laki-Laki	21	Fakultas Hukum	Ilmu K
## 34	45414.69	UBL	34	Laki-Laki	22	Fakultas Teknik	Teknik S
Semester 4)							
## 35	45414.70	UBL	35	Perempuan	22	Fakultas Ekonomi dan Bisnis	Manaj
Semester 4)							
## 36	45414.71	UBL	36	Laki-Laki	23	Fakultas Teknik	Teknik K
## 37	45414.71	UBL	37	Perempuan	23	Fakultas Teknik	Teknik K
## 38	45414.71	UBL	38	Perempuan	19	Fakultas Hukum	Ilmu K
Semester 4)							
##	Uang.Saku kepemilikan.mobil kepemilikan.motor kepemilikan.sepeda						

## 1	< 1 jt	1	2	1	
## 2	1 jt - 2 jt	2	1	1	
## 3	1 jt - 2 jt	1	2	2	
## 4	1 jt - 2 jt	1	4	1	Kendaraan Bermotor (menumpang)
## 5	2,1 jt - 3 jt	4	4	1	
## 6	1 jt - 2 jt	1	2	1	
## 7	1 jt - 2 jt	1	1	1	
## 8	< 1 jt	1	1	1	
## 9	1 jt - 2 jt	1	1	1	
## 10	1 jt - 2 jt	1	1	1	
## 11	1 jt - 2 jt	NA	1	NA	
## 12	1 jt - 2 jt	0	2	0	
## 13	1 jt - 2 jt	0	0	0	
## 14	1 jt - 2 jt	1	1	0	
## 15	< 1 jt	0	1	0	
## 16	< 1 jt	0	1	0	
## 17	1 jt - 2 jt	0	0	0	
## 18	2,1 jt - 3 jt	0	1	0	
## 19	< 1 jt	1	0	0	
## 20	2,1 jt - 3 jt	1	2	0	
## 21	2,1 jt - 3 jt	1	2	3	
## 22	< 1 jt	2	1	1	
## 23	< 1 jt	0	1	0	
## 24	< 1 jt	0	0	0	Kendaraan Bermotor (menumpang)
## 25	< 1 jt	0	1	1	
## 26	1 jt - 2 jt	0	0	0	
## 27	1 jt - 2 jt	0	1	0	
## 28	2,1 jt - 3 jt	2	3	3	
## 29	1 jt - 2 jt	1	1	1	
## 30	2,1 jt - 3 jt	1	3	0	
## 31	2,1 jt - 3 jt	2	2	0	
## 32	1 jt - 2 jt	1	2	0	
## 33	1 jt - 2 jt	2	2	0	
## 34	1 jt - 2 jt	1	2	0	
## 35	2,1 jt - 3 jt	3	2	0	
## 36	1 jt - 2 jt	2	2	1	
## 37	2,1 jt - 3 jt	2	3	1	
## 38	1 jt - 2 jt	1	2	2	
##	kelurahan		jenis.tempat.tinggal		
## 1	Kalibalau Kencana		Rumah pribadi/rumah keluarga		
## 2	Gunung sulah		Rumah pribadi/rumah keluarga		
## 3	langkapura		Rumah pribadi/rumah keluarga		
## 4	Bilabong JAYA JAYA JAYA		Rumah pribadi/rumah keluarga		
## 5	sukarame		Rumah pribadi/rumah keluarga		
## 6	Way halim		Rumah pribadi/rumah keluarga		
## 7	Segala Mider		Rumah pribadi/rumah keluarga		
## 8	pidada		Rumah pribadi/rumah keluarga		
## 9	Sukarame		Rumah pribadi/rumah keluarga		
## 10	kupang kota		Rumah pribadi/rumah keluarga		
## 11	Labuhan dalam		Rumah mengontrak pribadi		
## 12	Labuhan dalam		Rumah mengontrak bersama-sama		
## 13	Labuhan dalam		Rumah mengontrak pribadi		
## 14	Sukaraja		Rumah pribadi/rumah keluarga		
## 15	Labuhan dalam		Rumah mengontrak pribadi		

## 16	Seputih Banyak	Kos Sendiri	
## 17	Wah Areng	Kos Sendiri	
## 18	waydadi	Kos bersama-sama	
## 19	natar	Rumah bersama saudara	
## 20	Gang harapan 1	Kos Sendiri	
## 21	Kalianda	Kos bersama-sama	
## 22	Gunung terang	Kos bersama-sama	
## 23	Gunung terang gang swadaya IV	Kos Sendiri	
## 24	Jl BDN 2 TANJUNG SENANG BANDAR LAMPUNG	Kos bersama-sama	
## 25	Sawah Brebes	Rumah pribadi/rumah keluarga	
## 26	pringsewu	Asrama	
## 27	labuhan dalam	Kos bersama-sama	
## 28	dempo 1 , labuhan ratu	Kos Sendiri	
## 29	Tanjung harapan kec.sep banyak Lampung tengah	Rumah pribadi/rumah keluarga	
## 30	gunung terang	Rumah pribadi/rumah keluarga	
## 31	Kedaton Belakang Ubl	Kos Sendiri	
## 32	sukamenanti	Rumah pribadi/rumah keluarga	
## 33	Langkapura Baru	Rumah pribadi/rumah keluarga	
## 34	Rajabasa	Rumah pribadi/rumah keluarga	
## 35	Gotong royong	Rumah pribadi/rumah keluarga	
## 36	Kelurahan talang	Rumah pribadi/rumah keluarga	
## 37	pahoman	Rumah pribadi/rumah keluarga	
## 38	Jatiagung Lampung selatan	Rumah pribadi/rumah keluarga	
##		nama.jalan.tempat.tinggal	jarak.(km)
## 1		eLBe Laundry	4.2237967
## 2		Jl.Urip Sumoharjo no 88	2.7033310
## 3		JL DARUSSALAM GG LANGGAR LK II	3.2370722
## 4		Jl Darussalam bilabong bila bolong di jait dong	3.0023362
## 5		jl.cendrawasih sukarama bandar lampung	5.9619250
## 6		Jl P tabuan nomor 26	2.9253306
## 7		Jalan Blora	2.0588294
## 8		jl.teluk ambon LK.III Panjang,kota bandar lampung	12.0078709
## 9		Jl. Pembangunan A5 ujung	4.7355783
## 10		jl. drs warsito	6.7413074
## 11		Gang demokrasi jl Flamboyan	2.4239314
## 12		Jl flyamboyan	2.4118160
## 13		Jl Cendana	2.7787467
## 14		Jl. Yos Sudarso	12.7777211
## 15		Jl Flamboyan	2.4118160
## 16		Jl. Harapan I Labuhan Ratu	0.7819544
## 17		Jl.Z.A Pagar Alam, Gg Harapan III	0.3231065
## 18		jl untung suropati gg rukun 1	1.9110358
## 19		jl,n,moh ali,pemanggilan,natar	6.8909558
## 20		Jl.ZA Pagar Alam gang harapan 1 labuhan ratu	0.1092263
## 21		Gunung terang langkapura	43.5451922
## 22		gang suadaya IV	1.9653510
## 23		Gang sawadaya IV gunung terang	1.9653510
## 24		Jl BDN 2	2.5078988
## 25		Pasar Tugu	3.7533554
## 26		Jl.jalan aja dulu	30.5272138
## 27		Gg. BDN 2	2.5078988
## 28		dempo 1	0.2986108
## 29	Zainal Abidin pagar alam, Rajabasa perumahan Kenzo residence, samping deler wuling		3.0204809
## 30		iya	1.2517151

## 31		Jl. Palapa IV no.31	0.3299862
## 32		Gg. beruang no 73	1.7784158
## 33		Jl. Darussalam Gg.langgar no.21	3.2370722
## 34		jl. nunyai no.64	2.1408915
## 35		Jl.Amir Hamzah 3 no.73	5.2967552
## 36		Jl. w.r Supratman gg beringin 2 no.43	7.2033834
## 37		jl way pengubuan no 23	5.6545326
## 38		Sidodadiasri jayiagung Lampung selatan	24.0211211
##	alasan.pemilihan.lokasi.tempat.tinggal		
## 1	Bersama keluarga/saudara/teman		
## 2	Mudahnya akses berpergian dari tempat tinggal		
## 3	Bersama keluarga/saudara/teman		
## 4	Lingkungan nyaman karna aman dari kejahatan		
## 5	Dekat dengan fasilitas umum		
## 6	Bersama keluarga/saudara/teman		
## 7	Bersama keluarga/saudara/teman		
## 8	Bersama keluarga/saudara/teman		
## 9	Bersama keluarga/saudara/teman		
## 10	Bersama keluarga/saudara/teman		
## 11	Dekat dengan fasilitas umum		
## 12	Lingkungan nyaman karna aman dari kejahatan		
## 13	Lingkungan nyaman karna aman dari kejahatan		
## 14	Bersama keluarga/saudara/teman		
## 15	Lain-lain		
## 16	Dekat dengan kampus		
## 17	Dekat dengan kampus		
## 18	Lingkungan nyaman karna aman dari kejahatan		
## 19	Bersama keluarga/saudara/teman		
## 20	Dekat dengan kampus		
## 21	Mudahnya akses berpergian dari tempat tinggal		
## 22	Dekat dengan fasilitas umum		
## 23	Dekat dengan fasilitas umum		
## 24	Lain-lain		
## 25	Bersama keluarga/saudara/teman		
## 26	Lingkungan nyaman karna aman dari kejahatan		
## 27	Lingkungan nyaman karna aman dari kejahatan		
## 28	Dekat dengan kampus		
## 29	Dekat dengan fasilitas umum		
## 30	Bersama keluarga/saudara/teman		
## 31	Dekat dengan kampus		
## 32	Bersama keluarga/saudara/teman		
## 33	Bersama keluarga/saudara/teman		
## 34	Bersama keluarga/saudara/teman		
## 35	Bersama keluarga/saudara/teman		
## 36	Bersama keluarga/saudara/teman		
## 37	Bersama keluarga/saudara/teman		
## 38	Bersama keluarga/saudara/teman		
##			
## 1			
## 2			
## 3			
## 4			
## 5			
## 6			


```

## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30
## 31
## 32
## 33
## 34
## 35
## 36
## 37
## 38
##
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21

```

jika di akumulasikan untuk bensin mobil saja bisa 500ribu seminggu, untuk makan 50ribu 1hari, jika

mengg

	Jumlah.perjalanan.Senin	Jumlah.Perjalanan.Selasa	Jumlah.Perjalanan.Rabu	Jumlah.Perjalanan.Kamis	J
1	2	3	2	3	
2	3	3	3	3	
3	1	1	1	1	
4	2	2	2	2	
5	1	1	1	1	
6	3	3	1	1	
7	2	2	2	2	
8	4	4	4	4	
9	4	4	4	4	
10	3	3	3	3	
11	Tidak Valid	3	1	1	
12	2	1	2	1	
13	Tidak Valid	1	1	1	
14	4	4	4	4	
15	2	1	3	3	
16	2	3	2	2	
17	2	3	4	2	
18	1	1	1	1	
19	1	1	1	1	
20	4	3	3	3	
21	4	1	1	1	

## 22	Tidak Valid	1	1	1
## 23	Tidak Valid	1	1	1
## 24	4	3	3	3
## 25	2	4	3	3
## 26	Tidak Valid	1	1	1
## 27	5	5	5	5
## 28	Tidak Valid	1	1	1
## 29	Tidak Valid	1	1	2
## 30	4	3	3	3
## 31	2	3	2	1
## 32	3	3	3	2
## 33	3	3	1	4
## 34	3	4	3	2
## 35	4	3	2	3
## 36	2	4	3	1
## 37	4	1	1	1
## 38	1	1	1	1
##	Jumlah.Perjalanan.Sabtu	Jumlah.Perjalanan.Ahad		
## 1	2	2		
## 2	3	3		
## 3	1	1		
## 4	2	2		
## 5	1	1		
## 6	1	1		
## 7	2	2		
## 8	3	3		
## 9	4	4		
## 10	2	1		
## 11	1	1		
## 12	1	1		
## 13	1	1		
## 14	1	1		
## 15	1	1		
## 16	2	1		
## 17	1	2		
## 18	1	1		
## 19	1	1		
## 20	3	3		
## 21	1	1		
## 22	1	1		
## 23	1	1		
## 24	3	3		
## 25	3	3		
## 26	1	1		
## 27	3	3		
## 28	1	1		
## 29	1	1		
## 30	3	1		
## 31	1	1		
## 32	4	1		
## 33	3	1		
## 34	3	1		
## 35	2	1		
## 36	3	1		

```
## 37          2          1
## 38          1          1
## [ reached 'max' / getOption("max.print") -- omitted 341 rows ]
```

Kita juga dapat melihat *dataset* kita secara lengkap dengan perintah `View()`

```
# Melihat set data UINIRL
View(data)

# Melihat set data UBL
View(data_ubl)
```

1.2 Analisis data terstruktur pada set data

Kita sudah mempelajari bahwa data terstruktur terdiri atas **objek**, **variabel**, dan **nilai**. Satu objek diwakili oleh satu baris, variabel oleh kolom, dan nilai berada di dalam sel-sel yang merupakan perpotongan kolom dan baris.

Penting

Di modul ini kita menggunakan istilah “variabel” untuk menyebut dua hal yang berbeda. Makna pertama mengacu pada penyimpan nilai di R, sementara yang kedua adalah elemen data terstruktur yang menjadi kolom-kolom.

Dalam modul ini kita akan menggunakan istilah yang sama untuk kedua makna, jadi pahami konteks kalimatnya terlebih dahulu ketika menemukan kata ini.

Untuk mengetahui ringkasan struktur data, kita dapat menggunakan fungsi `glimpse()` yang membutuhkan set data yang kita simpan sebagai argumen.

```
# ringkasan struktur data UIN RIL
glimpse(data)
```

```
## Rows: 400
## Columns: 25
## $ Kampus_PT                <chr> "UINRIL", "UINRIL", "UINRIL", "UINRIL", "UINRIL", "UI
## $ Nomor.urut               <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
## $ Jenis.Kelamin            <chr> "Laki-laki", "Laki-laki", "Perempuan", "Laki-laki", "
## $ Umur                     <dbl> 22, 25, 24, 19, 23, 21, 22, 21, 21, 21, 19, 19, 19, 1
## $ Fakultas                 <chr> "Syariah", "Syariah", "Tarbiyah dan Keguruan", "Dakwah
## $ Prodi                    <chr> "Hukum Keluarga", "Hukum Keluarga", "Pendidikan Agama
## $ Tingkat.Semester        <chr> "4 (Semester 7 - Semester 8)", "4 (Semester 7 - Semes
## $ Uang.Saku                 <chr> "1Jt - 2 Jt", "< 1 Jt", "< 1 Jt", "< 1 Jt", "< 1 Jt",
## $ kepemilikan.mobil        <dbl> 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
## $ kepemilikan.motor        <dbl> 1, 4, 0, 1, 1, 1, 1, 4, 4, 1, 0, 1, 3, 1, 0, 0, 1, 2,
## $ kepemilikan.sepedan      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
## $ kendaraan.utama          <chr> "Sepeda Motor Pribadi", "Sepeda Motor Pribadi", "Berj
## $ kelurahan                 <chr> "Wiyono", "Sukarama", "Sukarama", "Harapan Jaya", "Ko
## $ jenis.tempat.tinggal     <chr> "Rumah Bersama Saudara", "Kos Sendiri", "Kos Sendiri"
## $ nama.jalan.tempat.tinggal <chr> "Pasar Wiyono", "jln endro suratmin ", "Jl. Pulau amb
## $ alasan.pemilihan.lokasi.tempat.tinggal <chr> "Lain-lain", "Dekat dengan kampus", "Dekat dengan kamp
## $ `jarak.(km)`             <dbl> 19.27, 0.58, 0.56, 1.05, 1.69, 7.91, 2.58, 2.32, 0.77
## $ biaya.dalam.ribu2        <dbl> 100, 80, 0, 50, 50, 50, 35, 50, 40, 50, 0, 50, 50, 40
```

```
## $ Jumlah.Perjalanan.Senin      <dbl> 3, 3, 3, 0, 4, 0, 3, 4, 2, 3, 4, 2, 3, 2, 3, 2, 2, 2,
## $ Jumlah.Perjalanan.Selasa     <dbl> 3, 3, 0, 2, 4, 2, 0, 4, 0, 2, 4, 2, 2, 2, 3, 2, 2, 2,
## $ Jumlah.Perjalanan.Rabu       <dbl> 0, 0, 3, 0, 0, 2, 3, 4, 3, 2, 4, 2, 2, 2, 2, 2, 2, 0,
## $ Jumlah.Perjalanan.Kamis      <dbl> 0, 3, 0, 2, 0, 0, 0, 4, 0, 2, 2, 2, 2, 2, 0, 3, 2, 2,
## $ Jumlah.Perjalanan.Jumat      <dbl> 0, 0, 0, 2, 3, 2, 2, 4, 0, 0, 2, 2, 2, 2, 2, 0, 2, 2,
## $ Jumlah.Perjalanan.Sabtu      <dbl> 0, 0, 0, 0, 2, 2, 2, 2, 2, 0, 2, 2, 0, 0, 0, 0, 0, 2,
## $ Jumlah.Perjalanan.Ahad       <dbl> 0, 0, 0, 2, 0, 2, 2, 2, 2, 0, 2, 2, 0, 2, 2, 0, 0, 1,
```

Hasil dari fungsi `glimpse()` tersebut menunjukkan jumlah baris, jumlah kolom, dan kolom-kolom yang ada dalam set data kita beserta rinciannya. Set data kita memiliki 400 baris yang ditunjukkan oleh **Rows** dan 25 kolom yang ditunjukkan oleh **Columns**. Ini artinya, set data kita memiliki 400 objek dan 25 variabel.

Rincian nama-nama variabel ditunjukkan pada baris keluaran yang diawali oleh simbol `\$`. Singkatan-singkatan di dalam kurung sudut `<>` bermakna jenis nilai dari variabel tersebut. Laman ini memberikan penjelasan makna setiap singkatan tersebut.

Terlihat dalam data kita bahwa hanya ada dua tipe nilai variabel: `dbl` dan `chr` yang berarti *double* (bilangan desimal) dan *character* (huruf/string).

1.3 Mengolah set data

Dalam bagian ini kita akan mengolah set data kita yang bertujuan untuk memenuhi kebutuhan pengolahan data kita. Sebelum melakukan pengolahan, maka kita perlu menjawab pertanyaan terlebih dahulu “apa yang kita butuhkan untuk pengolahan data kita?”

Dalam praktikum ini kita sudah mengetahui bahwa terdapat dua jenis set data hasil kuesioner, yakni set data mahasiswa Universitas Islam Negeri Raden Intan Lampung (UINRIL) dan juga Universitas Bandar Lampung (UBL). Kedua jenis set data tersebut disimpan dalam variabel yang berbeda. Kita butuh untuk *menggabungkan kedua set data* tersebut menjadi satu set data yang mencakup kedua kampus agar nanti kita bisa melakukan analisis dengan lebih leluasa.

Pengolahan set data seperti ini punya istilah yang keren: *data wrangling* (bisa dibaca “data rengling”)

1.3.1 Merancang Algoritma (Urutan kerja)

Karena R adalah program yang mayoritasnya *character user interface* (CUI) alih-alih *graphical user interface*, maka pekerjaan kita tidak bisa selemuasa GUI yang dapat kita bolak-balik urutannya. Kita sebaiknya memiliki “rencana kerja” yang jelas, yang dalam bahasa pemrograman disebut **algoritma**.

Algoritma yang akan kita lakukan dalam praktikum ini adalah sebagai berikut:

1. Mengidentifikasi nama-nama kolom di set data UIN RIL
2. Mengidentifikasi nama-nama kolom di set data UBL
3. Menentukan kolom-kolom yang akan diambil di set data baru
4. Menyeleksi kolom-kolom yang akan dipakai di set data UIN RIL dan UBL,
5. Mengubah nama kolom-kolom di kedua set data menjadi nama-nama yang sudah ditentukan di langkah-3.
6. Menyatukan kedua set data.

Tip

Sebisanya mungkin rancang setiap langkah pada urutan kerja kita dengan rinci agar kita dapat menerjemahkan langkah-langkah tersebut dengan lebih mudah ke baris kode.

1.3.2 Mengidentifikasi Nama-nama Kolom

Untuk mengidentifikasi nama-nama kolom kita dapat menggunakan fungsi `colnames()` yang mengambil argumen nama set data.

```
# data UIN RIL
colnames(data)
```

```
## [1] "Kampus_PT" "Nomor.urut" "Jenis.Kelamin"
## [4] "Umur" "Fakultas" "Prodi"
## [7] "Tingkat.Semester" "Uang.Saku" "kepemilikan.m"
## [10] "kepemilikan.motor" "kepemilikan.sepedan" "kendaraan.uta"
## [13] "kelurahan" "jenis.tempat.tinggal" "nama.jalan.t"
## [16] "alasan.pemilihan.lokasi.tempat.tinggal" "jarak.(km)" "biaya.dalam."
## [19] "Jumlah.Perjalanan.Senin" "Jumlah.Perjalanan.Selasa" "Jumlah.Perja"
## [22] "Jumlah.Perjalanan.Kamis" "Jumlah.Perjalanan.Jumat" "Jumlah.Perja"
## [25] "Jumlah.Perjalanan.Ahad"
```

```
# data UIN RIL
colnames(data_ubl)
```

```
## [1] "Timestamp" "Kampus_PT" "Nomor.urut"
## [4] "Jenis.Kelamin" "Umur" "Fakultas"
## [7] "Prodi" "Tingkat.Semester" "Uang.Saku"
## [10] "kepemilikan.mobil" "kepemilikan.motor" "kepemilikan."
## [13] "kendaraan.utama" "kelurahan" "jenis.tempat"
## [16] "nama.jalan.tempat.tinggal" "jarak.(km)" "alasan.pemil"
## [19] "biaya.dalam.seminggu" "Jumlah.perjalanan.Senin" "Jumlah.Perja"
## [22] "Jumlah.Perjalanan.Rabu" "Jumlah.Perjalanan.Kamis" "Jumlah.Perja"
## [25] "Jumlah.Perjalanan.Sabtu" "Jumlah.Perjalanan.Ahad"
```

Kita dapat mengetahui jumlah kolom setiap set data secara langsung dengan fungsi `length()`.

```
length(colnames(data)) # jumlah kolom set data UIN RIL
```

```
## [1] 25
```

```
length(colnames(data_ubl)) # jumlah kolom set data UBL
```

```
## [1] 26
```

Penting!

Fungsi `length()` pada dasarnya adalah fungsi penghitung jumlah elemen dalam suatu vektor atau jenis data lain berupa daftar.

```
# Mendefinisikan suatu vektor
suatu_vektor <- c(1, 2, 3, 4, 5)
```

```
# Mengukur panjangnya
length(suatu_vektor)
```

```
## [1] 5
```

Dari kedua fungsi tersebut, kita dapat mengetahui bahwa terdapat perbedaan jumlah kolom dan juga perbedaan penamaan:

- `kepemilikan.sepedan` di data UIN vs `kepemilikan.sepeda` di data UBL.
- `biaya.dalam.ribu2` di data UIN vs `biaya.dalam.seminggu` di data UBL.

Perbedaan-perbedaan inilah yang membuat kita tidak bisa langsung menggabungkan data. Kita harus menyeragamkannya terlebih dahulu,

1.3.3 Menentukan Kolom-kolom yang Akan Diambil di Set Data Baru

Setelah mengidentifikasi nama-nama variabel dan jumlahnya, kita putuskan untuk mengambil variabel-variabel berikut: (1) `Nomor.urut`, (2) `Jenis.Kelamin`, (3) `Umur`, (4) `Fakultas`, (5) `Prodi`, (6) `Tingkat.Semester`, (7) `Uang.Saku`, (8) `kepemilikan.mobil`, (9) `kepemilikan.motor`, (10) `kepemilikan.sepeda`, (11) `kendaraan.utama`, (12) `jenis.tempat.tinggal`, (13) `jarak`, (14) `biaya.dalam.seminggu`, (15) `Jumlah.perjalanan.Senin`, (16) `Jumlah.perjalanan.Selasa`, (17) `Jumlah.perjalanan.Rabu`, (18) `Jumlah.perjalanan.Kamis`, (19) `Jumlah.perjalanan.Jumat`, (20) `Jumlah.perjalanan.Sabtu`, (21) `Jumlah.perjalanan.Ahad`.

Kita akan menyimpan nama-nama kolom yang sudah kita tentukan tersebut menjadi sebuah vektor.

```
# Membuat vektor berisi nama-nama kolom final yang kita inginkan
kolom_final <- c(
  "Nomor.urut", "Jenis.Kelamin", "Umur", "Fakultas", "Prodi",
  "Tingkat.Semester", "Uang.Saku",
  "kepemilikan.mobil",
  "kepemilikan.motor", "kepemilikan.sepeda", "kendaraan.utama",
  "jenis.tempat.tinggal", "jarak", "biaya.dalam.sepekan",
  "Jumlah.perjalanan.Senin", "Jumlah.perjalanan.Selasa",
  "Jumlah.perjalanan.Rabu", "Jumlah.perjalanan.Kamis",
  "Jumlah.perjalanan.Jumat", "Jumlah.perjalanan.Sabtu",
  "Jumlah.perjalanan.Ahad"
)
```

1.3.4 Menyeleksi Kolom-kolom Yang Akan Dipakai

Kita akan menggunakan fungsi `select()` dari `dplyr` (bagian dari `tidyverse`) untuk memilih hanya kolom-kolom yang relevan dari masing-masing set data. Perhatikan baik-baik, kita harus menggunakan **nama kolom asli** saat menyeleksi.

```
# 4. Menyeleksi kolom dari data UIN RIL
data_uin_selected <- select(data,
  Nomor.urut, Jenis.Kelamin, Umur, Fakultas, Prodi, Tingkat.Semester,
  Uang.Saku,
  kepemilikan.mobil, kepemilikan.motor,
  kepemilikan.sepedan, # Ingat, nama aslinya pakai 'n'
  kendaraan.utama, jenis.tempat.tinggal,
  `jarak.(km)`, # Gunakan backtick (`) karena ada karakter spesial, yakni tanda kurung.
  `biaya.dalam.ribu2`,
```

```

Jumlah.Perjalanan.Senin, Jumlah.Perjalanan.Selasa, Jumlah.Perjalanan.Rabu,
Jumlah.Perjalanan.Kamis, Jumlah.Perjalanan.Jumat, Jumlah.Perjalanan.Sabtu,
Jumlah.Perjalanan.Ahad
)

```

```

# 5. Menyeleksi kolom dari data UBL
data_ubl_selected <- select(data_ubl,
  Nomor.urut, Jenis.Kelamin, Umur, Fakultas, Prodi, Tingkat.Semester,
  Uang.Saku,
  kepemilikan.mobil, kepemilikan.motor,
  kepemilikan.sepeda, # Di sini namanya sudah benar
  kendaraan.utama, jenis.tempat.tinggal,
  `jarak.(km)`,
  biaya.dalam.seminggu,
  Jumlah.perjalanan.Senin, Jumlah.Perjalanan.Selasa, Jumlah.Perjalanan.Rabu,
  Jumlah.Perjalanan.Kamis, Jumlah.Perjalanan.Jumat, Jumlah.Perjalanan.Sabtu,
  Jumlah.Perjalanan.Ahad
)

```

Sekarang kita punya dua *data frame* baru (*data_uin_selected* dan *data_ubl_selected*) yang isinya hanya kolom-kolom yang kita butuhkan.

1.3.5 Mengubah Nama Kolom Menjadi Nama Standar

Ini adalah langkah kuncinya. Kita cukup menimpa nama-nama kolom di kedua *data frame* yang sudah kita seleksi tadi dengan nama-nama kolom dari vektor *kolom_final* yang kita buat di 3.3.3.

```

# Mengganti nama kolom di kedua data frame
colnames(data_uin_selected) <- kolom_final
colnames(data_ubl_selected) <- kolom_final

```

Untuk membuktikan bahwa namanya sudah berubah, kita bisa cek lagi.

```

# Cek nama kolom setelah diubah
colnames(data_uin_selected)

```

```

## [1] "Nomor.urut"          "Jenis.Kelamin"       "Umur"                "Fakultas"
## [6] "Tingkat.Semester"    "Uang.Saku"           "kepemilikan.mobil"   "kepemilikan.motor"
## [11] "kendaraan.utama"     "jenis.tempat.tinggal" "jarak"               "biaya.dalam.seminggu"
## [16] "Jumlah.perjalanan.Selasa" "Jumlah.perjalanan.Rabu" "Jumlah.perjalanan.Kamis" "Jumlah.perjalanan.Ahad"
## [21] "Jumlah.perjalanan.Ahad"

```

```

print("=====")

```

```

## [1] "=====

```

```

colnames(data_ubl_selected)

```

```

## [1] "Nomor.urut"          "Jenis.Kelamin"       "Umur"                "Fakultas"
## [6] "Tingkat.Semester"    "Uang.Saku"           "kepemilikan.mobil"   "kepemilikan.motor"
## [11] "kendaraan.utama"     "jenis.tempat.tinggal" "jarak"               "biaya.dalam.seminggu"
## [16] "Jumlah.perjalanan.Selasa" "Jumlah.perjalanan.Rabu" "Jumlah.perjalanan.Kamis" "Jumlah.perjalanan.Ahad"
## [21] "Jumlah.perjalanan.Ahad"

```

Hasil dari kedua perintah di atas seharusnya adalah daftar nama yang identik.

1.3.6 Menyatukan Kedua Set Data

Ini adalah langkah terakhir kita. Karena kedua set data kini memiliki struktur kolom yang sama persis, kita bisa menyatukannya menjadi satu *data frame* besar dengan fungsi `bind_rows()`.

```
# Menggabungkan kedua data frame menjadi satu
data_gabungan <- bind_rows(data_uin_selected, data_ubl_selected)
```

Hasil dari penggabungan tersebut menyebabkan *error* pada kolom `biaya.dalam.sepekan`. Ini disebabkan oleh `biaya.dalam.sepekan` dalam set data UINRIL (`data`) berbeda jenisnya dengan set data UBL (`data_ubl`). Begitu pula `Jumlah.perjalanan.Senin`

```
# Mengecek tipe data dalam set data UINRIL
glimpse(data_uin_selected)
```

```
## Rows: 400  
## Columns: 21  
## $ Nomor.urut <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, ...  
## $ Jenis.Kelamin <chr> "Laki-laki", "Laki-laki", "Perempuan", "Laki-laki", "Laki-laki", "La...  
## $ Umur <dbl> 22, 25, 24, 19, 23, 21, 22, 21, 21, 21, 19, 19, 19, 19, 19, 20, 19,...  
## $ Fakultas <chr> "Syariah", "Syariah", "Tarbiyah dan Keguruan", "Dakwah dan Ilmu Komu...  
## $ Prodi <chr> "Hukum Keluarga", "Hukum Keluarga", "Pendidikan Agama Islam", "Komun...  
## $ Tingkat.Semester <chr> "4 (Semester 7 - Semester 8)", "4 (Semester 7 - Semester 8)", "4 (S...  
## $ Uang.Saku <chr> "1Jt - 2 Jt", "< 1 Jt", "< 1 Jt", "< 1 Jt", "< 1 Jt", "< 1 Jt", "< 1...  
## $ kepemilikan.mobil <dbl> 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...  
## $ kepemilikan.motor <dbl> 1, 4, 0, 1, 1, 1, 1, 4, 4, 1, 0, 1, 3, 1, 0, 0, 1, 2, 2, 1, 0, 0, 1...  
## $ kepemilikan.sepeda <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0...  
## $ kendaraan.utama <chr> "Sepeda Motor Pribadi", "Sepeda Motor Pribadi", "Berjalan Kaki", "Se...  
## $ jenis.tempat.tinggal <chr> "Rumah Bersama Saudara", "Kos Sendiri", "Kos Sendiri", "Kos Sendiri...  
## $ jarak <dbl> 19.27, 0.58, 0.56, 1.05, 1.69, 7.91, 2.58, 2.32, 0.77, 6.56, 0.63, (...)  
## $ biaya.dalam.sepekan <dbl> 100, 80, 0, 50, 50, 50, 35, 50, 40, 50, 0, 50, 50, 40, 0, 0, 30, 50...  
## $ Jumlah.perjalanan.Senin <dbl> 3, 3, 3, 0, 4, 0, 3, 4, 2, 3, 4, 2, 3, 2, 3, 2, 2, 3, 2, 0, 2, 0, 0...  
## $ Jumlah.perjalanan.Selasa <dbl> 3, 3, 0, 2, 4, 2, 0, 4, 0, 2, 4, 2, 2, 2, 3, 2, 2, 2, 3, 0, 0, 0, 2...  
## $ Jumlah.perjalanan.Rabu <dbl> 0, 0, 3, 0, 0, 2, 3, 4, 3, 2, 4, 2, 2, 2, 2, 2, 2, 0, 3, 2, 2, 2, 2...  
## $ Jumlah.perjalanan.Kamis <dbl> 0, 3, 0, 2, 0, 0, 0, 4, 0, 2, 2, 2, 2, 2, 0, 3, 2, 2, 2, 2, 2, 2, 2...  
## $ Jumlah.perjalanan.Jumat <dbl> 0, 0, 0, 2, 3, 2, 2, 4, 0, 0, 2, 2, 2, 2, 2, 0, 2, 2, 0, 2, 2, 2, 2...  
## $ Jumlah.perjalanan.Sabtu <dbl> 0, 0, 0, 0, 2, 2, 2, 2, 2, 0, 2, 2, 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 0...  
## $ Jumlah.perjalanan.Ahad <dbl> 0, 0, 0, 2, 0, 2, 2, 2, 2, 0, 2, 2, 0, 2, 2, 0, 0, 1, 2, 0, 0, 2, 2
```

```
print("=====")
```

```
## [1] "=====
```

```
# Mengecek tipe data dalam set data UBL
glimpse(data_ubl_selected)
```

```
## Rows: 379
## Columns: 21
## $ Nomor.urut      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 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
```



```
## $ Jenis.Kelamin      <chr> "Perempuan", "Laki-Laki", "Laki-Laki", "Laki-Laki", "Laki-Laki", "P
## $ Umur                <dbl> 21, 20, 21, 22, 21, 21, 22, 22, 22, 22, 21, 22, 22, 22, 19, 22, 23,
## $ Fakultas           <chr> "Fakultas Ilmu Sosial dan Politik", "Fakultas Hukum", "Fakultas Ekon
## $ Prodi              <chr> "Ilmu Komunikasi", "Ilmu Hukum", "Manajemen", "Akuntansi", "Manajem
## $ Tingkat.Semester  <chr> "4 (Semester 7 - Semester 8)", "3 (Semester 5 - Semester 6)", "4 (S
## $ Uang.Saku          <chr> "< 1 jt", "1 jt - 2 jt", "1 jt - 2 jt", "1 jt -
2 jt", "2,1 jt - 3 jt", "1 jt - 2 jt", "1 jt - 2 jt", "< 1 ~
## $ kepemilikan.mobil  <dbl> 1, 2, 1, 1, 4, 1, 1, 1, 1, 1, NA, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 2,
## $ kepemilikan.motor  <dbl> 2, 1, 2, 4, 4, 2, 1, 1, 1, 1, 1, 2, 0, 1, 1, 1, 0, 1, 0, 2, 2, 1, 1
## $ kepemilikan.sepeda <dbl> 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, NA, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 1,
## $ kendaraan.utama    <chr> "Sepeda Motor Pribadi", "Mobil Pribadi", "Sepeda Motor Pribadi", "K
## $ jenis.tempat.tinggal <chr> "Rumah pribadi/rumah keluarga", "Rumah pribadi/rumah keluarga", "Ru
## $ jarak              <dbl> 4.2237967, 2.7033310, 3.2370722, 3.0023362, 5.9619250, 2.9253306, 2
## $ biaya.dalam.sepekan <chr> "25 ribu", "200", "50rb menggunakan kendaraan pribadi", "Sminggu 20
## $ Jumlah.perjalanan.Senin <chr> "2", "3", "1", "2", "1", "3", "2", "4", "4", "3", "Tidak Valid", "2
## $ Jumlah.perjalanan.Selasa <dbl> 3, 3, 1, 2, 1, 3, 2, 4, 4, 3, 3, 1, 1, 4, 1, 3, 3, 1, 1, 3, 1, 1, 1
## $ Jumlah.perjalanan.Rabu <dbl> 2, 3, 1, 2, 1, 1, 2, 4, 4, 3, 1, 2, 1, 4, 3, 2, 4, 1, 1, 3, 1, 1, 1
## $ Jumlah.perjalanan.Kamis <dbl> 3, 3, 1, 2, 1, 1, 2, 4, 4, 3, 1, 1, 1, 4, 3, 2, 2, 1, 1, 3, 1, 1, 1
## $ Jumlah.perjalanan.Jumat <dbl> 2, 3, 1, 2, 1, 1, 2, 4, 3, 3, 1, 2, 1, 4, 3, 2, 3, 1, 1, 3, 1, 1, 1
## $ Jumlah.perjalanan.Sabtu <dbl> 2, 3, 1, 2, 1, 1, 2, 3, 4, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 3, 1, 1, 1
## $ Jumlah.perjalanan.Ahad <dbl> 2, 3, 1, 2, 1, 1, 2, 3, 4, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 3, 1, 1, 1
```

Ini adalah kasus yang tepat untuk menggambarkan bahwa dalam data terstruktur, jenis nilai harus sama persis. Untuk kesederhanaan praktikum, kita akan melewati dulu variabel `biaya.dalam.sepekan` dan `Jumlah.perjalanan.Senin` tersebut.

1.3.7 Memperbaiki Set Data yang Akan Digabungkan

Kita akan menghapus `biaya.dalam.sepekan` dari kedua set data kita. Untuk menghapus suatu kolom, kita dapat menggunakan fungsi `select()` juga, akan tetapi kita tambahkan tanda negatif (-) di depan nama variabelnya.

```
# 'Menimpa' data_uin_selected dengan data_uin_selected yang variabel biaya.dalam.sepekan sudah dihapus
data_uin_selected <- select(data_uin_selected,
                             -`biaya.dalam.sepekan`, -`Jumlah.perjalanan.Senin`)

# Melihat hasilnya
glimpse(data_uin_selected)
```

```
## Rows: 400
## Columns: 19
## $ Nomor.urut        <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2
## $ Jenis.Kelamin      <chr> "Laki-laki", "Laki-laki", "Perempuan", "Laki-laki", "Laki-laki", "L
## $ Umur                <dbl> 22, 25, 24, 19, 23, 21, 22, 21, 21, 21, 19, 19, 19, 19, 20, 19,
## $ Fakultas           <chr> "Syariah", "Syariah", "Tarbiyah dan Keguruan", "Dakwah dan Ilmu Kom
## $ Prodi              <chr> "Hukum Keluarga", "Hukum Keluarga", "Pendidikan Agama Islam", "Komu
## $ Tingkat.Semester  <chr> "4 (Semester 7 - Semester 8)", "4 (Semester 7 - Semester 8)", "4 (S
## $ Uang.Saku          <chr> "1Jt - 2 Jt", "< 1 Jt", "< 1 Jt", "< 1 Jt", "< 1 Jt", "< 1 Jt", "<
## $ kepemilikan.mobil  <dbl> 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
## $ kepemilikan.motor  <dbl> 1, 4, 0, 1, 1, 1, 1, 4, 4, 1, 0, 1, 3, 1, 0, 0, 1, 2, 2, 1, 0, 0, 1
## $ kepemilikan.sepeda <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0
## $ kendaraan.utama    <chr> "Sepeda Motor Pribadi", "Sepeda Motor Pribadi", "Berjalan Kaki", "S
## $ jenis.tempat.tinggal <chr> "Rumah Bersama Saudara", "Kos Sendiri", "Kos Sendiri", "Kos Sendiri
```

```
colnames(data_uin_selected)
```

```
# 'Menimpa' data_ubl_selected dengan data_ubl_selected yang variabel biaya.dalam.sepekan sudah dihapus
data_ubl_selected <- select(data_ubl_selected,
                             -`biaya.dalam.sepekan`, -`Jumlah.perjalanan.Senin`)
```

```
glimpse(data_ubl_selected)
```

1.3.8 Menyatukan Kedua Set Data Setelah Diperbaiki

```
data_gabungan <- bind_rows(data_uin_selected, data_ubl_selected)
```

```
# Tampilkan struktur data gabungan yang sudah rapi
glimpse(data_gabungan)
```

1.3.9 Mengekspor Set Data Hasil Pengolahan ke Fail (*File*) Terpisah

```
Nama; umur
Ifna; 20
Salbina; 21
```

Nama	umur
Ifna	20
Salbina	21

```
write.csv2(variabel_dataset, "nama_file_ekspor.csv")  
write.xlsx(variabel_dataset, "nama_file_ekspor.xlsx")
```

27

1.3.9.1 Mengekspor set data ke fail .csv Terdapat dua jenis fungsi untuk mengekspor set data ke fail .csv: `write.csv()` dan `write.csv2()`. Perbedaan antara keduanya adalah **pemisah nilainya**.

Fungsi `write.csv()` digunakan untuk set data yang menggunakan **tanda titik** sebagai pemisah desimal, sehingga pemisahannya adalah **tanda koma (,)**. Ini adalah format yang biasa dipakai di Amerika Utara.

Sementara itu, `write.csv2()` menggunakan pemisah **titik-koma (;)** (*semicolon*). Ini digunakan untuk set data yang pemisah desimalnya adalah **tanda koma**.

```
# Mengekspor data_gabungan dalam format csv
write.csv2(data_gabungan, "Data UINRIL & UBL.csv")
```

1.3.9.2 Mengekspor set data ke fail .xlsx Sama seperti `write.csv2()`, fungsi `write.xlsx()` juga mengikuti format yang sama.

```
write.xlsx(data_gabungan, "Data UINRIL & UBL.xlsx")
```

1.4 Soal Latihan

1. Dari fail (*file*) “Data Praktikum 01.xlsx” terdapat dua *sheet* lagi yang merupakan hasil kuesioner mahasiswa ITERA dan UNILA. Untuk hasil kuesioner mahasiswa ITERA terletak pada *sheet* bernama `DataUtama_mhsITERA` dan hasil kuesioner mahasiswa UNILA pada `DataUtama_mhsUNILA`. Tentukan sebanyak mungkin kolom yang bisa digabungkan, yakni yang tipenya sama. Gabungkan dua set data tersebut seperti yang sudah kalian pelajari dan simpan menjadi fail dengan nama `Prak1_[NIM1]_[NIM2].xlsx`. STP-1.4.
2. Berapa variabel objek yang kalian hasilkan dalam data gabungan? Tunjukkan beserta buktinya STP-1.3.

2 Modul-2: Analisis Statistika Deskriptif

Setelah mempelajari modul ini, Anda diharapkan dapat mengoperasikan perhitungan persentase/proporsi, rasio, laju, ukuran pemusatan, dan ukuran penyebaran untuk suatu data kuantitatif dengan menggunakan perangkat lunak STP-2.4

Dalam praktikum ini kita akan melanjutkan analisis data terstruktur kita dengan analisis statistika deskriptif untuk variabel-variabel **kategoris** dan **numerik**. Kedua jenis variabel ini berkaitan juga dengan tingkat pengukurannya yang terdiri atas **nominal**, **ordinal**, dan **interval/rasio (metrik)**.

Analisis statistik deskriptif yang akan kita lakukan mencakup **persentase/proporsi**, **rata-rata (*mean*)**, **median**, dan **standar deviasi**. Ada beberapa ukuran statistik yang tidak kita pelajari di kelas tetapi dikenalkan pada praktikum ini, seperti **persentil** dan **kuartil**. Perhitungan IQV akan dilakukan dengan sekaligus mengenalkan mekanisme `function` di R.

2.1 Persiapan

Seperti biasa, kita perlu memuat *library* andalan kita dalam mengolah data, *tidyverse*, dan juga untuk membaca file Excel, *openxlsx*.

```
# Memuat library yang diperlukan
library(tidyverse)
library(openxlsx)
```

2.1.1 Mengimpor Dataset

Kita menggunakan data mahasiswa UBL yang sudah lengkap karena sudah diolah kembali.

```
# Mengeset variabel tersendiri untuk nama file nama sheet
file.dibaca <- "datasets/Data Praktikum 02.xlsx"
sheet.ubl <- "DataUtama_mhsUBL"

# Mengimpor file menjadi dataset memanfaatkan variabel nama file dan nama sheet yang kita buat tadi
data.ubl <- read.xlsx(file.dibaca, sheet = sheet.ubl)
```

Sebelumnya kita sudah mengenal *glimpse()* yang berasal dari *library dplyr*, sekarang kita akan mengenal perintah untuk menampilkan ringkasan dataset bawaan R (disebut juga ‘base R’) bernama *summary()*.

```
# Melihat ringkasan dataset (cara lain)
summary(data.ubl)
```

```
##      Timestamp      Kampus_PT      Nomor.urut      Jenis.Kelamin      Umur      Fakultas
##  Min.   :45414      Length:379      Min.    : 1.0      Length:379      Min.   :18.00      Length:379
##  1st Qu.:45416      Class :character      1st Qu.: 95.5      Class :character      1st Qu.:20.00      Class :charac
##  Median :45423      Mode  :character      Median :190.0      Mode  :character      Median :21.00      Mode  :charac
##  Mean   :45420                                     Mean   :190.0                                     Mean   :20.78
##  3rd Qu.:45423                                     3rd Qu.:284.5                                     3rd Qu.:22.00
##  Max.   :45424                                     Max.   :379.0                                     Max.   :23.00
##
##  Tingkat.Semester      Uang.Saku      kepemilikan.mobil      kepemilikan.motor      kepemilikan.sepeda      kendar
##  Length:379            Length:379            Min.   :0.0000      Min.   :0.0000      Min.   :0.0000      Length
##  Class :character      Class :character      1st Qu.:0.0000      1st Qu.:1.000      1st Qu.:0.0000      Class
##  Mode  :character      Mode  :character      Median :1.0000      Median :1.000      Median :0.0000      Mode
##                                     Mean   :0.9654      Mean   :1.517      Mean   :0.2857
##                                     3rd Qu.:2.0000      3rd Qu.:2.000      3rd Qu.:0.0000
##                                     Max.   :4.0000      Max.   :5.000      Max.   :3.0000
##                                     NA's   :3          NA's   :1
##  jenis.tempat.tinggal      nama.jalan.tempat.tinggal      jarak.(km)      alasan.pemilihan.lokasi.tempat.ti
##  Length:379            Length:379            Min.   : 0.06352      Length:379
##  Class :character      Class :character      1st Qu.: 1.31127      Class :character
##  Mode  :character      Mode  :character      Median : 2.69037      Mode  :character
##                                     Mean   : 3.37914
##                                     3rd Qu.: 4.69466
##                                     Max.   :43.54519
##
##  Jumlah.perjalanan.Senin      Jumlah.Perjalanan.Selasa      Jumlah.Perjalanan.Rabu      Jumlah.Perjalanan.Kamis      Jum
##  Min.   :1.000            Min.   :1.000            Min.   :1.000            Min.   :1.000            Min
```

```
## 1st Qu.:2.000      1st Qu.:1.000      1st Qu.:1.000      1st Qu.:1.000      1st
## Median :2.000      Median :2.000      Median :2.000      Median :2.000      Med
## Mean   :2.679      Mean   :2.264      Mean   :2.092      Mean   :2.092      Mean
## 3rd Qu.:4.000      3rd Qu.:3.000      3rd Qu.:3.000      3rd Qu.:3.000      3rd
## Max.    :8.000      Max.    :6.000      Max.    :6.000      Max.    :7.000      Max
## NA's    :43
## Jumlah.Perjalanan.Sabtu Jumlah.Perjalanan.Ahad
## Min.     :1.000      Min.     :1.00
## 1st Qu.:1.000      1st Qu.:1.00
## Median :2.000      Median :1.00
## Mean   :1.749      Mean   :1.52
## 3rd Qu.:2.000      3rd Qu.:2.00
## Max.    :5.000      Max.    :4.00
##
```

2.1.2 Merapikan Dataset (*Data Wrangling*)

Sering dikatakan bahwa 50% pekerjaan ilmuwan data adalah **merapikan dataset**. Ini diistilahkan dengan *wrangling*. *Data wrangling* mencakup penggantian nama variabel, pengecekan nilai yang hilang (*missing values*), mengubah/manipulasi variabel, dan masih banyak lagi.

2.1.2.1 Mengganti Nama Variabel Pada praktikum sebelumnya kita mengubah nama kolom/variabel dalam *dataset* secara **keseluruhan**, yakni dengan menimpakan vektor nama-nama baru variabel kita dengan fungsi `colnames()`.

Kita pastinya tidak akan selalu mengubah semua kolom secara sekaligus, karena itu kita harus belajar bagaimana cara mengganti nama beberapa kolom saja, yakni dengan perintah `rename()`.

Perintah `rename()` memiliki sintaks seperti berikut

```
rename(dataset_kita, nama_kolom_baru = nama_kolom_lama)
```

```
# Membaca ulang dataset supaya menjadi kondisi semula
data.ubl <- read.xlsx(file.dibaca, sheet = sheet.ubl)
colnames(data.ubl)
```

```
## [1] "Timestamp"      "Kampus_PT"      "Nomor.urut"
## [4] "Jenis.Kelamin"  "Umur"           "Fakultas"
## [7] "Prodi"          "Tingkat.Semester" "Uang.Saku"
## [10] "kepemilikan.mobil" "kepemilikan.motor" "kepemilikan.kendaraan"
## [13] "kendaraan.utama" "kelurahan"       "jenis.tempat.tinggal"
## [16] "nama.jalan.tempat.tinggal" "jarak.(km)"      "alasan.pemilihan"
## [19] "biaya.dalam.seminggu" "Jumlah.perjalanan.Senin" "Jumlah.Perjalanan.Senin"
## [22] "Jumlah.Perjalanan.Rabu" "Jumlah.Perjalanan.Kamis" "Jumlah.Perjalanan.Kamis"
## [25] "Jumlah.Perjalanan.Sabtu" "Jumlah.Perjalanan.Ahad" "Jumlah.Perjalanan.Ahad"
```

```
# Mengganti nama variabel-variabel
data.ubl <- rename(data.ubl, biaya.dalam.sepekan = biaya.dalam.seminggu)
data.ubl <- rename(data.ubl, Jumlah.Perjalanan.Senin = Jumlah.perjalanan.Senin)
data.ubl <- rename(data.ubl, jarak = `jarak.(km)`) # tanda backtick (`) digunakan di antara "jarak.(km)"
# karena ada tanda kurung yang dianggap oleh R sebagai komentar
```

Pengenalan Teknik Pengolahan dalam R: *Pipe Operator* (`|>`)

Dalam pengolahan data dengan *tidyverse* penting untuk kita kuasai penggunaan operator yang satu ini. Perhatikan dua cara berikut untuk mengganti nama variabel-variabel kita.

CARA-1: CARA KONVENSIONAL

```
# Membaca ulang dataset supaya menjadi kondisi semula
data.ubl <- read.xlsx(file.dibaca, sheet = sheet.ubl)
colnames(data.ubl)
```

```
## [1] "Timestamp"          "Kampus_PT"          "Nomor.urut"
## [4] "Jenis.Kelamin"      "Umur"               "Fakultas"
## [7] "Prodi"              "Tingkat.Semester"   "Uang.Saku"
## [10] "kepemilikan.mobil"  "kepemilikan.motor"  "kepemilikan."
## [13] "kendaraan.utama"    "kelurahan"          "jenis.tempat"
## [16] "nama.jalan.tempat.tinggal" "jarak.(km)"         "alasan.pemil."
## [19] "biaya.dalam.seminggu" "Jumlah.perjalanan.Senin" "Jumlah.Perja."
## [22] "Jumlah.Perjalanan.Rabu" "Jumlah.Perjalanan.Kamis" "Jumlah.Perja."
## [25] "Jumlah.Perjalanan.Sabtu" "Jumlah.Perjalanan.Ahad"
```

Mengganti nama variabel-variabel

```
data.ubl <- rename(data.ubl, biaya.dalam.sepekan = biaya.dalam.seminggu)
data.ubl <- rename(data.ubl, Jumlah.Perjalanan.Senin = Jumlah.perjalanan.Senin)
data.ubl <- rename(data.ubl, jarak = `jarak.(km)`)
```

CARA-2: MENGGUNAKAN PIPE OPERATOR

```
# Membaca ulang dataset supaya menjadi kondisi semula
data.ubl <- read.xlsx(file.dibaca, sheet = sheet.ubl)
```

Mengganti nama variabel-variabel

```
data.ubl <- data.ubl |>
  rename(biaya.dalam.sepekan = biaya.dalam.seminggu) |>
  # setiap baris yang masih ada proses selanjutnya harus diisi dengan pipe operator
  rename(Jumlah.Perjalanan.Senin = Jumlah.perjalanan.Senin) |>
  rename(jarak = `jarak.(km)`)
```

Dalam cara 1, kita harus menulis `data.ubl` berkali-kali untuk setiap perintah. Ini mungkin tidak akan terasa jika jumlah kolom yang kita miliki sedikit. Tetapi, akan lain ceritanya jika jumlahnya banyak. Selain itu, data kita tidak bisa secara intuitif seperti biasanya kita membaca suatu teks dari kiri ke kanan.

Operator pipa (`|>`) memungkinkan hal tersebut. Selain kita tidak perlu menuliskan dataset kita berkali-kali, kita dengan lebih mudah memahami baris-baris kode kita tersebut sebagai proses dari atas ke bawah, dari kiri ke kanan, seperti pipa.

Berikut adalah penjelasan cara-2:

- `data.ubl |>`: perintah “Ambil dataset `data.ubl`.”
- Kirimkan dataset itu ke fungsi berikutnya (`rename`).
- Hasil dari fungsi `rename` pertama dikirimkan lagi ke fungsi `rename` kedua, dan seterusnya.

Penting

Pipe operator dapat dimasukkan dengan kombinasi tombol `Ctrl + Shift + M` di *keyboard*.

2.1.2.2 Mengecek Missing Values Pengecekan *missing values* dilakukan dengan perintah `is.na()` yang menghasilkan tabel berisi nilai *boolean* FALSE dan TRUE yang berarti ‘tidak kosong’ dan ‘kosong’ secara berturut-turut.

`colSums()` menjumlahkan nilai-nilai TRUE pada setiap kolom, menandakan berapa jumlah objek yang kosong nilainya.

```
checkMV_ubl <- is.na(data_ubl)
```

```
checkMV_ubl
```

```
##      Timestamp Kampus_PT Nomor.urut Jenis.Kelamin  Umur Fakultas Prodi Tingkat.Semester Uang.Saku kep
## 1      FALSE      FALSE      FALSE      FALSE FALSE      FALSE FALSE      FALSE      FALSE      FALSE
## 2      FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 3      FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 4      FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 5      FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 6      FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 7      FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 8      FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 9      FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 10     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 11     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 12     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 13     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 14     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 15     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 16     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 17     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 18     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 19     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 20     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 21     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 22     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 23     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 24     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 25     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 26     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 27     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 28     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 29     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 30     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 31     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 32     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 33     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 34     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 35     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 36     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 37     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
## 38     FALSE      FALSE      FALSE      FALSE FALSE FALSE      FALSE FALSE      FALSE      FALSE
##      kepemilikan.sepeda kendaraan.utama kelurahan jenis.tempat.tinggal nama.jalan.tempat.tinggal jarak
## 1      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE
## 2      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE
## 3      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE
```


## 4	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 5	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 6	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 7	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 8	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 9	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 10	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 11	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
## 12	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 13	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 14	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 15	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 16	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 17	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 18	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 19	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 20	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 21	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 22	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 23	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 24	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 25	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 26	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 27	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 28	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 29	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 30	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 31	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 32	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 33	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 34	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 35	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 36	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 37	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 38	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
##	biaya.dalam.sepekan	Jumlah.Perjalanan.Senin	Jumlah.Perjalanan.Selasa	Jumlah.Perjalanan.Rabu	Jumlah.Perjalanan.Kamis	Jumlah.Perjalanan.Jumat
## 1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 2	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 3	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 4	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 5	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
## 6	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 7	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 8	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 9	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 10	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 11	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
## 12	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 13	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
## 14	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 15	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 16	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 17	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
## 18	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

## 19	FALSE	FALSE	FALSE	FALSE
## 20	FALSE	FALSE	FALSE	FALSE
## 21	FALSE	FALSE	FALSE	FALSE
## 22	FALSE	TRUE	FALSE	FALSE
## 23	FALSE	TRUE	FALSE	FALSE
## 24	FALSE	FALSE	FALSE	FALSE
## 25	FALSE	FALSE	FALSE	FALSE
## 26	FALSE	TRUE	FALSE	FALSE
## 27	FALSE	FALSE	FALSE	FALSE
## 28	FALSE	TRUE	FALSE	FALSE
## 29	TRUE	TRUE	FALSE	FALSE
## 30	FALSE	FALSE	FALSE	FALSE
## 31	FALSE	FALSE	FALSE	FALSE
## 32	FALSE	FALSE	FALSE	FALSE
## 33	FALSE	FALSE	FALSE	FALSE
## 34	FALSE	FALSE	FALSE	FALSE
## 35	FALSE	FALSE	FALSE	FALSE
## 36	FALSE	FALSE	FALSE	FALSE
## 37	FALSE	FALSE	FALSE	FALSE
## 38	FALSE	FALSE	FALSE	FALSE
##	Jumlah.Perjalanan.Jumat	Jumlah.Perjalanan.Sabtu	Jumlah.Perjalanan.Ahad	
## 1	FALSE	FALSE	FALSE	
## 2	FALSE	FALSE	FALSE	
## 3	FALSE	FALSE	FALSE	
## 4	FALSE	FALSE	FALSE	
## 5	FALSE	FALSE	FALSE	
## 6	FALSE	FALSE	FALSE	
## 7	FALSE	FALSE	FALSE	
## 8	FALSE	FALSE	FALSE	
## 9	FALSE	FALSE	FALSE	
## 10	FALSE	FALSE	FALSE	
## 11	FALSE	FALSE	FALSE	
## 12	FALSE	FALSE	FALSE	
## 13	FALSE	FALSE	FALSE	
## 14	FALSE	FALSE	FALSE	
## 15	FALSE	FALSE	FALSE	
## 16	FALSE	FALSE	FALSE	
## 17	FALSE	FALSE	FALSE	
## 18	FALSE	FALSE	FALSE	
## 19	FALSE	FALSE	FALSE	
## 20	FALSE	FALSE	FALSE	
## 21	FALSE	FALSE	FALSE	
## 22	FALSE	FALSE	FALSE	
## 23	FALSE	FALSE	FALSE	
## 24	FALSE	FALSE	FALSE	
## 25	FALSE	FALSE	FALSE	
## 26	FALSE	FALSE	FALSE	
## 27	FALSE	FALSE	FALSE	
## 28	FALSE	FALSE	FALSE	
## 29	FALSE	FALSE	FALSE	
## 30	FALSE	FALSE	FALSE	
## 31	FALSE	FALSE	FALSE	
## 32	FALSE	FALSE	FALSE	
## 33	FALSE	FALSE	FALSE	

```
## 34          FALSE          FALSE          FALSE
## 35          FALSE          FALSE          FALSE
## 36          FALSE          FALSE          FALSE
## 37          FALSE          FALSE          FALSE
## 38          FALSE          FALSE          FALSE
## [ reached 'max' / getOption("max.print") -- omitted 341 rows ]
```

```
colSums(checkMV_ubl)
```

```
##          Timestamp          Kampus_PT
##          0          0
##          Jenis.Kelamin          Umur
##          0          0
##          Prodi          Tingkat.Semester
##          0          0
##          kepemilikan.mobil          kepemilikan.motor          ke
##          3          0
##          kendaraan.utama          kelurahan          jeni
##          0          0
##          nama.jalan.tempat.tinggal          jarak alasan.pemilihan.lokas
##          4          0
##          biaya.dalam.sepekan          Jumlah.Perjalanan.Senin          Jumlah.P
##          2          43
##          Jumlah.Perjalanan.Rabu          Jumlah.Perjalanan.Kamis          Jumlah.L
##          0          0
##          Jumlah.Perjalanan.Sabtu          Jumlah.Perjalanan.Ahad
##          0          0
```

2.1.2.3 Menghapus Missing Values Penghapusan *missing values* juga dapat digunakan dengan *pipe operator*. Adapun perintah yang dapat kita gunakan adalah `drop_na()` dari paket `tidyr`.

```
# Menghapus observasi yang memiliki Missing Values
data_ubl <- data_ubl |>
  drop_na()

# Mengecek kembali Missing Values
checkMV_ubl <- is.na(data_ubl)
colSums(checkMV_ubl)
```

```
##          Timestamp          Kampus_PT
##          0          0
##          Jenis.Kelamin          Umur
##          0          0
##          Prodi          Tingkat.Semester
##          0          0
##          kepemilikan.mobil          kepemilikan.motor          ke
##          0          0
##          kendaraan.utama          kelurahan          jeni
##          0          0
##          nama.jalan.tempat.tinggal          jarak alasan.pemilihan.lokas
##          0          0
##          biaya.dalam.sepekan          Jumlah.Perjalanan.Senin          Jumlah.P
```

```
##                                0                                0
##                Jumlah.Perjalanan.Rabu                Jumlah.Perjalanan.Kamis                Jumlah.L
##                                0                                0
##                Jumlah.Perjalanan.Sabtu                Jumlah.Perjalanan.Ahad
##                                0                                0
```

2.1.2.4 Menyesuaikan Variabel Nominal dan Ordinal dengan factor() Apabila kita mengimpor dataset dengan variabel nonangka, bentuknya akan terbaca sebagai *character* (chr). Perintah `factor()` dari paket `forcats` berfungsi untuk mendefinisikan nilai-nilai yang ada dalam variabel kategoris tersebut sehingga kita mendefinisikan urutannya dan menetapkan sebagai variabel ordinal.

```
# Mengidentifikasi variabel-variabel yang bisa diubah menjadi factor
glimpse(data.ubl)
```

```
## Rows: 330
## Columns: 26
## $ Timestamp                <dbl> 45413.83, 45413.83, 45413.85, 45413.85, 45413.92, 454
## $ Kampus_PT                <chr> "UBL", "UBL", "UBL", "UBL", "UBL", "UBL", "UBL", "UBL
## $ Nomor.urut                <dbl> 1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 14, 15, 16, 17, 18, 1
## $ Jenis.Kelamin            <chr> "Perempuan", "Laki-Laki", "Laki-Laki", "Laki-Laki", "L
## $ Umur                      <dbl> 21, 20, 21, 22, 21, 22, 22, 22, 22, 22, 19, 22, 2
## $ Fakultas                 <chr> "Fakultas Ilmu Sosial dan Politik", "Fakultas Hukum",
## $ Prodi                    <chr> "Ilmu Komunikasi", "Ilmu Hukum", "Manajemen", "Akuntan
## $ Tingkat.Semester        <chr> "4 (Semester 7 - Semester 8)", "3 (Semester 5 - Semes
## $ Uang.Saku                 <chr> "< 1 jt", "1 jt - 2 jt", "1 jt - 2 jt", "1 jt -
2 jt", "1 jt - 2 jt", "1 jt - 2 jt", "< 1 jt"~
## $ kepemilikan.mobil        <dbl> 1, 2, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1,
## $ kepemilikan.motor        <dbl> 2, 1, 2, 4, 2, 1, 1, 1, 1, 2, 1, 1, 0, 1, 0, 2, 2,
## $ kepemilikan.sepeda       <dbl> 1, 1, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 3,
## $ kendaraan.utama          <chr> "Sepeda Motor Pribadi", "Mobil Pribadi", "Sepeda Moto
## $ kelurahan                 <chr> "Kalibalau Kencana", "Gunung sulah", "langkapura", "B
## $ jenis.tempat.tinggal      <chr> "Rumah pribadi/rumah keluarga", "Rumah pribadi/rumah l
## $ nama.jalan.tempat.tinggal <chr> "eLBe Laundry", "Jl.Urip Sumoharjo no 88", "JL DARUSS
## $ jarak                     <dbl> 4.2237967, 2.7033310, 3.2370722, 3.0023362, 2.9253306
## $ alasan.pemilihan.lokasi.tempat.tinggal <chr> "Bersama keluarga/saudara/teman", "Mudahnya akses ber
## $ biaya.dalam.sepekan       <dbl> 25, 200, 50, 20, 40, 70, 35, 50, 350, 50, 30, 35, 50,
## $ Jumlah.Perjalanan.Senin   <dbl> 2, 3, 1, 2, 3, 2, 4, 4, 3, 2, 4, 2, 2, 1, 1, 4, 4,
## $ Jumlah.Perjalanan.Selasa  <dbl> 3, 3, 1, 2, 3, 2, 4, 4, 3, 1, 4, 1, 3, 3, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Rabu     <dbl> 2, 3, 1, 2, 1, 2, 4, 4, 3, 2, 4, 3, 2, 4, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Kamis    <dbl> 3, 3, 1, 2, 1, 2, 4, 4, 3, 1, 4, 3, 2, 2, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Jumat    <dbl> 2, 3, 1, 2, 1, 2, 4, 3, 3, 2, 4, 3, 2, 3, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Sabtu    <dbl> 2, 3, 1, 2, 1, 2, 3, 4, 2, 1, 1, 1, 2, 1, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Ahad     <dbl> 2, 3, 1, 2, 1, 2, 3, 4, 1, 1, 1, 1, 1, 2, 1, 1, 3, 1,
```

Contoh pertama kita adalah variabel yang paling mudah: `Jenis.Kelamin`.

```
# Mengubah variabel Jenis.Kelamin dari chr menjadi factor (fct)
jk <- c("Laki-laki", "Perempuan") # Membuat vektor rincian nilai nominal

# Mengubah jenis variabel "Jenis.Kelamin" dari chr ke fct dengan 'mutate()'
data.ubl <- data.ubl |>
  mutate(Jenis.Kelamin = factor(Jenis.Kelamin,jk))
```

```
# Mengecek hasil
glimpse(data.ubl)
```

```
## Rows: 330
## Columns: 26
## $ Timestamp          <dbl> 45413.83, 45413.83, 45413.85, 45413.85, 45413.92, 454
## $ Kampus_PT          <chr> "UBL", "UBL", "UBL", "UBL", "UBL", "UBL", "UBL", "UBL
## $ Nomor.urut         <dbl> 1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 14, 15, 16, 17, 18, 1
## $ Jenis.Kelamin      <fct> Perempuan, NA, NA, NA, Perempuan, Perempuan, Perempuan
## $ Umur               <dbl> 21, 20, 21, 22, 21, 22, 22, 22, 22, 22, 19, 22, 2
## $ Fakultas           <chr> "Fakultas Ilmu Sosial dan Politik", "Fakultas Hukum",
## $ Prodi              <chr> "Ilmu Komunikasi", "Ilmu Hukum", "Manajemen", "Akuntan
## $ Tingkat.Semester  <chr> "4 (Semester 7 - Semester 8)", "3 (Semester 5 - Semes
## $ Uang.Saku          <chr> "< 1 jt", "1 jt - 2 jt", "1 jt - 2 jt", "1 jt -
2 jt", "1 jt - 2 jt", "1 jt - 2 jt", "< 1 jt"~
## $ kepemilikan.mobil  <dbl> 1, 2, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1,
## $ kepemilikan.motor  <dbl> 2, 1, 2, 4, 2, 1, 1, 1, 1, 2, 1, 1, 1, 0, 1, 0, 2, 2,
## $ kepemilikan.sepeda <dbl> 1, 1, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 3,
## $ kendaraan.utama    <chr> "Sepeda Motor Pribadi", "Mobil Pribadi", "Sepeda Moto
## $ kelurahan          <chr> "Kalibalau Kencana", "Gunung sulah", "langkapura", "B
## $ jenis.tempat.tinggal <chr> "Rumah pribadi/rumah keluarga", "Rumah pribadi/rumah l
## $ nama.jalan.tempat.tinggal <chr> "eLBe Laundry", "Jl.Urip Sumoharjo no 88", "JL DARUSS
## $ jarak              <dbl> 4.2237967, 2.7033310, 3.2370722, 3.0023362, 2.9253306
## $ alasan.pemilihan.lokasi.tempat.tinggal <chr> "Bersama keluarga/saudara/teman", "Mudahnya akses ber
## $ biaya.dalam.sepekan <dbl> 25, 200, 50, 20, 40, 70, 35, 50, 350, 50, 30, 35, 50,
## $ Jumlah.Perjalanan.Senin <dbl> 2, 3, 1, 2, 3, 2, 4, 4, 3, 2, 4, 2, 2, 2, 1, 1, 4, 4,
## $ Jumlah.Perjalanan.Selasa <dbl> 3, 3, 1, 2, 3, 2, 4, 4, 3, 1, 4, 1, 3, 3, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Rabu <dbl> 2, 3, 1, 2, 1, 2, 4, 4, 3, 2, 4, 3, 2, 4, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Kamis <dbl> 3, 3, 1, 2, 1, 2, 4, 4, 3, 1, 4, 3, 2, 2, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Jumat <dbl> 2, 3, 1, 2, 1, 2, 4, 3, 3, 2, 4, 3, 2, 3, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Sabtu <dbl> 2, 3, 1, 2, 1, 2, 3, 4, 2, 1, 1, 1, 2, 1, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Ahad <dbl> 2, 3, 1, 2, 1, 2, 3, 4, 1, 1, 1, 1, 1, 2, 1, 1, 3, 1,
```

Ternyata, hasil pendefinisian **factor** ke dalam data kita mengandung NA yang berarti data kosong atau *missing value*. Hal tersebut terjadi karena kita mendefinisikan vektor **jk** berisi `c("Laki-laki", "Perempuan")`.

Sementara itu, jika kita cermat, sebelumnya nilai jenis kelamin ditulis dengan "**Laki-Laki**" . Maka, kita harus memperbaiki vektor kita yang berisi nilai-nilai yang sesuai dengan dataset terlebih dahulu.

```
# Mengurangi pembacaan file sekaligus merapikan nama variabel-variabelnya serta menghilangkan Missing Value
data.ubl <- read.xlsx(file.dibaca, sheet = sheet.ubl) |>
  rename(biaya.dalam.sepekan = biaya.dalam.seminggu) |>
  rename(Jumlah.Perjalanan.Senin = Jumlah.perjalanan.Senin) |>
  rename(jarak = `jarak.(km)` ) |>
  drop_na()

# Memperbaiki vektor nilai nominal sesuai dengan yang ada di data dengan mengganti penulisan "Laki-laki"
jk <- c("Laki-Laki", "Perempuan")

# Mengubah lagi jenis variabel "Jenis.Kelamin" dari chr ke fct dengan 'mutate()'
data.ubl <- data.ubl |>
  mutate(Jenis.Kelamin = factor(Jenis.Kelamin,jk))
```

```
# Mengecek hasil
glimpse(data.ubl)
```

```
## Rows: 330
## Columns: 26
## $ Timestamp                <dbl> 45413.83, 45413.83, 45413.85, 45413.85, 45413.92, 454
## $ Kampus_PT                <chr> "UBL", "UBL", "UBL", "UBL", "UBL", "UBL", "UBL", "UBL
## $ Nomor.urut               <dbl> 1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 14, 15, 16, 17, 18, 1
## $ Jenis.Kelamin            <fct> Perempuan, Laki-Laki, Laki-Laki, Laki-Laki, Perempuan
## $ Umur                     <dbl> 21, 20, 21, 22, 21, 22, 22, 22, 22, 22, 19, 22, 2
## $ Fakultas                 <chr> "Fakultas Ilmu Sosial dan Politik", "Fakultas Hukum",
## $ Prodi                    <chr> "Ilmu Komunikasi", "Ilmu Hukum", "Manajemen", "Akuntan
## $ Tingkat.Semester        <chr> "4 (Semester 7 - Semester 8)", "3 (Semester 5 - Semes
## $ Uang.Saku                 <chr> "< 1 jt", "1 jt - 2 jt", "1 jt - 2 jt", "1 jt -
2 jt", "1 jt - 2 jt", "1 jt - 2 jt", "< 1 jt"~
## $ kepemilikan.mobil        <dbl> 1, 2, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1,
## $ kepemilikan.motor        <dbl> 2, 1, 2, 4, 2, 1, 1, 1, 1, 2, 1, 1, 1, 0, 1, 0, 2, 2,
## $ kepemilikan.sepeda       <dbl> 1, 1, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 3,
## $ kendaraan.utama          <chr> "Sepeda Motor Pribadi", "Mobil Pribadi", "Sepeda Moto
## $ kelurahan                 <chr> "Kalibalau Kencana", "Gunung sulah", "langkapura", "B
## $ jenis.tempat.tinggal      <chr> "Rumah pribadi/rumah keluarga", "Rumah pribadi/rumah l
## $ nama.jalan.tempat.tinggal <chr> "eLBe Laundry", "Jl.Urip Sumoharjo no 88", "JL DARUSS
## $ jarak                     <dbl> 4.2237967, 2.7033310, 3.2370722, 3.0023362, 2.9253306
## $ alasan.pemilihan.lokasi.tempat.tinggal <chr> "Bersama keluarga/saudara/teman", "Mudahnya akses ber
## $ biaya.dalam.sepekan       <dbl> 25, 200, 50, 20, 40, 70, 35, 50, 350, 50, 30, 35, 50,
## $ Jumlah.Perjalanan.Senin   <dbl> 2, 3, 1, 2, 3, 2, 4, 4, 3, 2, 4, 2, 2, 2, 1, 1, 4, 4,
## $ Jumlah.Perjalanan.Selasa  <dbl> 3, 3, 1, 2, 3, 2, 4, 4, 3, 1, 4, 1, 3, 3, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Rabu    <dbl> 2, 3, 1, 2, 1, 2, 4, 4, 3, 2, 4, 3, 2, 4, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Kamis   <dbl> 3, 3, 1, 2, 1, 2, 4, 4, 3, 1, 4, 3, 2, 2, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Jumat   <dbl> 2, 3, 1, 2, 1, 2, 4, 3, 3, 2, 4, 3, 2, 3, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Sabtu   <dbl> 2, 3, 1, 2, 1, 2, 3, 4, 2, 1, 1, 1, 2, 1, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Ahad    <dbl> 2, 3, 1, 2, 1, 2, 3, 4, 1, 1, 1, 1, 1, 2, 1, 1, 3, 1,
```

Kita sudah berhasil menjadikan variabel `Jenis.Kelamin` sebagai factor. Selanjutnya, kita akan mengubah variabel-variabel kategoris lain menjadi factor juga. Variabel-variabel tersebut di antaranya adalah `Prodi`, `Tingkat.Semester`, `Uang.Saku`, dan `jenis.tempat.tinggal`.

Sekarang kita akan mengecek nilai variabel-variabel kategoris tersebut dengan dua cara berikut:

```
# Mengecek rincian nilai-nilai variabel lain secara bersamaan
table(data.ubl$Fakultas) # cara 1
```

```
##
##          Fakultas Ekonomi dan Bisnis          Fakultas Hukum          Fakultas
##                               104                               98
##          Fakultas Ilmu Sosial dan Politik Fakultas Keguruan dan Ilmu Pendidikan          Fal
##                               52                               6
```

```
data.ubl |> # cara 2
count(Prodi)
```

```
##          Prodi  n
```

```
## 1      Administrasi Bisnis 15
## 2      Administrasi Publik 21
## 3              Akuntansi 23
## 4              Arsitektur  8
## 5              Ilmu Hukum  98
## 6              Ilmu Komunikasi 20
## 7              Informatika 18
## 8              Manajemen 77
## 9 Pendidikan Bahasa Inggris 6
## 10             Sistem Informasi 13
## 11             Teknik Mesin 7
## 12             Teknik Sipil 24
```

```
data.ubl |>
  count(Tingkat.Semester)
```

```
##           Tingkat.Semester  n
## 1 1 (Semester 1 - Semester 2) 34
## 2 2 (Semester 3 - Semester 4) 50
## 3 3 (Semester 5 - Semester 6) 63
## 4 4 (Semester 7 - Semester 8) 183
```

```
data.ubl |>
  count(Uang.Saku)
```

```
##           Uang.Saku  n
## 1  1 jt - 2 jt 167
## 2 2,1 jt - 3 jt 116
## 3 3,1 jt - 4 jt  18
## 4    < 1 jt  28
## 5    > 4 jt   1
```

```
table(data.ubl$jenis.tempat.tinggal)
```

```
##
##           Kos bersama-sama           Kos Sendiri           Rumah bersama saudara Rumah meng
##                   14                   52                   23
##           Rumah mengontrak pribadi  Rumah pribadi/rumah keluarga
##                   3                   233
```

Kemudian kita buat vektor-vektor yang menyimpan nilai-nilai yang mungkin.

Perhatikan bahwa untuk **variabel ordinal** kita harus menuliskan vektor nilainya dengan urutan yang sesuai dengan tingkatannya.

```
# Membuat vektor nilai-nilai kategoris (untuk factor) berdasarkan nilai-nilai dari tampilan rincian
fakultas <- c("Fakultas Ekonomi dan Bisnis", "Fakultas Hukum",
             "Fakultas Ilmu Komputer", "Fakultas Ilmu Sosial dan Politik",
             "Fakultas Keguruan dan Ilmu Pendidikan", "Fakultas Teknik")
# nilai-nilai dalam vektor dapat di-Enter setelah tanda koma agar script lebih rapi dan enak dibaca
prodi <- c("Administrasi Bisnis", "Administrasi Publik",
```

```

      "Akuntansi","Arsitektur","Ilmu Hukum","Ilmu Komunikasi",
      "Informatika","Manajemen",
      "Pendidikan Bahasa Inggris", "Sistem Informasi")

# Variabel ordinal harus ditulis sesuai dengan urutan yang benar
tingkat <- c("1 (Semester 1 - Semester 2)",
            "2 (Semester 3 - Semester 4)",
            "3 (Semester 5 - Semester 6)",
            "4 (Semester 7 - Semester 8)")

uang_saku <- c("< 1 jt", "1 jt - 2 jt", "2,1 jt - 3 jt",
              "3,1 jt - 4 jt", "> 4 jt")

```

Sekarang kita sudah bisa mengubah nilai-nilai di variabel kategoris tersebut dari *character* menjadi *factor*. Variabel ordinal menggunakan atribut `ordered` yang bernilai `TRUE`, artinya urutan dipentingkan dalam levels.

```

# Mengubah variabel-variabel tersebut menjadi factor
data.ubl <- data.ubl |>
  mutate(Jenis.Kelamin = factor(Jenis.Kelamin, levels = jk),
         Fakultas = factor(Fakultas, levels = fakultas),
         Prodi = factor(Prodi, levels = prodi),
         # jangan lupa atribut 'ordered = TRUE' untuk var. ordinal
         Tingkat.Semester = factor(Tingkat.Semester,
                                    levels = tingkat,
                                    ordered = TRUE),
         Uang.Saku = factor(Uang.Saku, levels = uang_saku, ordered = TRUE))

# Mengecek hasil
glimpse(data.ubl)

```

```

## Rows: 330
## Columns: 26
## $ Timestamp                <dbl> 45413.83, 45413.83, 45413.85, 45413.85, 45413.92, 454
## $ Kampus_PT                <chr> "UBL", "UBL", "UBL", "UBL", "UBL", "UBL", "UBL", "UBL
## $ Nomor.urut               <dbl> 1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 14, 15, 16, 17, 18, 1
## $ Jenis.Kelamin            <fct> Perempuan, Laki-Laki, Laki-Laki, Laki-Laki, Perempuan
## $ Umur                     <dbl> 21, 20, 21, 22, 21, 22, 22, 22, 22, 22, 22, 22, 2
## $ Fakultas                 <fct> Fakultas Ilmu Sosial dan Politik, Fakultas Hukum, Fak
## $ Prodi                    <fct> Ilmu Komunikasi, Ilmu Hukum, Manajemen, Akuntansi, Ak
## $ Tingkat.Semester        <ord> 4 (Semester 7 - Semester 8), 3 (Semester 5 - Semester
## $ Uang.Saku                 <ord> "< 1 jt", "1 jt - 2 jt", "1 jt - 2 jt", "1 jt -
2 jt", "1 jt - 2 jt", "1 jt - 2 jt", "< 1 jt"~
## $ kepemilikan.mobil         <dbl> 1, 2, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1,
## $ kepemilikan.motor         <dbl> 2, 1, 2, 4, 2, 1, 1, 1, 1, 2, 1, 1, 1, 0, 1, 0, 2, 2,
## $ kepemilikan.sepeda        <dbl> 1, 1, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 3,
## $ kendaraan.utama           <chr> "Sepeda Motor Pribadi", "Mobil Pribadi", "Sepeda Moto
## $ kelurahan                 <chr> "Kalibatalau Kencana", "Gunung sulah", "langkapura", "B
## $ jenis.tempat.tinggal       <chr> "Rumah pribadi/rumah keluarga", "Rumah pribadi/rumah l
## $ nama.jalan.tempat.tinggal  <chr> "eLBe Laundry", "Jl.Urip Sumoharjo no 88", "JL DARUSS
## $ jarak                     <dbl> 4.2237967, 2.7033310, 3.2370722, 3.0023362, 2.9253306
## $ alasan.pemilihan.lokasi.tempat.tinggal <chr> "Bersama keluarga/saudara/teman", "Mudahnya akses ber
## $ biaya.dalam.sepekan        <dbl> 25, 200, 50, 20, 40, 70, 35, 50, 350, 50, 30, 35, 50,

```



```
## $ Jumlah.Perjalanan.Senin      <dbl> 2, 3, 1, 2, 3, 2, 4, 4, 3, 2, 4, 2, 2, 2, 1, 1, 4, 4,
## $ Jumlah.Perjalanan.Selasa     <dbl> 3, 3, 1, 2, 3, 2, 4, 4, 3, 1, 4, 1, 3, 3, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Rabu       <dbl> 2, 3, 1, 2, 1, 2, 4, 4, 3, 2, 4, 3, 2, 4, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Kamis      <dbl> 3, 3, 1, 2, 1, 2, 4, 4, 3, 1, 4, 3, 2, 2, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Jumat      <dbl> 2, 3, 1, 2, 1, 2, 4, 3, 3, 2, 4, 3, 2, 3, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Sabtu      <dbl> 2, 3, 1, 2, 1, 2, 3, 4, 2, 1, 1, 1, 2, 1, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Ahad       <dbl> 2, 3, 1, 2, 1, 2, 3, 4, 1, 1, 1, 1, 1, 2, 1, 1, 3, 1,
```

Sekarang, tipe data untuk variabel-variabel kategoris kita tadi sudah menampilkan `<fct>` yang berarti “factor”. Variabel ordinal diidentifikasi dengan tipe `<ord>`.

Kita dapat mengecek urutan factor kita dengan perintah `levels()` berikut.

```
# Mengecek hasil dengan mengidentifikasi level factor untuk data kategoris-ordinal
levels(data.ubl$Tingkat.Semester)
```

```
## [1] "1 (Semester 1 - Semester 2)" "2 (Semester 3 - Semester 4)" "3 (Semester 5 - Semester 6)" "4 (Semester 7 - Semester 8)"
```

```
levels(data.ubl$Uang.Saku)
```

```
## [1] "< 1 jt"      "1 jt - 2 jt"   "2,1 jt - 3 jt" "3,1 jt - 4 jt" "> 4 jt"
```

2.2 Analisis Statistika Deskriptif

Kita akan menggunakan paket `gtsummary` dan `flextable` untuk melakukan analisis statistik deskriptif dengan ringkas. Paket ini memiliki perintah `tbl_summary()` yang dapat mengeluarkan analisis statistika deskriptif secara intuitif berdasarkan jenis data yang kita masukkan.

Untuk meng-install kedua paket tersebut, hapus tanda pagar di depan baris pertama dalam *chunk* di bawah dan muat paket `gtsummary` saja.

```
# install.packages(c("gtsummary", "flextable"))
library(gtsummary)
```

2.2.1 Variabel Kategoris

Kita masih akan menggunakan *pipe operator* dalam mengoperasikan analisis ini. Perintah yang kita gunakan adalah `tbl_summary()` yang kita teruskan ke `as_flex_table()` sebagai keluarannya.

Untuk menampilkan persentase/proporsi, kita cukup memasukkan variabel-variabel kategoris ke dalam tabel dengan atribut `include`. Atribut tersebut menerima masukan berupa vektor nama-nama variabel kategoris dalam dataset kita. Fungsi `tbl_summary` ini akan menampilkan secara otomatis tabel distribusi frekuensinya disertai dengan nilai persentasenya di samping nilai frekuensi tersebut.

```
data.ubl |>
  tbl_summary(include = c(Jenis.Kelamin,
                          Fakultas,
                          Tingkat.Semester,
                          Uang.Saku,
                          jenis.tempat.tinggal)) |>
  as_flex_table()
```

Characteristic	N = 330 ¹
Jenis.Kelamin	
Laki-Laki	165 (50%)
Perempuan	165 (50%)
Fakultas	
Fakultas Ekonomi dan Bisnis	104 (32%)
Fakultas Hukum	98 (30%)
Fakultas Ilmu Komputer	29 (8.8%)
Fakultas Ilmu Sosial dan Politik	52 (16%)
Fakultas Keguruan dan Ilmu Pendidikan	6 (1.8%)
Fakultas Teknik	41 (12%)
Tingkat.Semester	
1 (Semester 1 – Semester 2)	34 (10%)
2 (Semester 3 – Semester 4)	50 (15%)
3 (Semester 5 - Semester 6)	63 (19%)
4 (Semester 7 - Semester 8)	183 (55%)
Uang.Saku	
< 1 jt	28 (8.5%)
1 jt – 2 jt	167 (51%)
2,1 jt – 3 jt	116 (35%)
3,1 jt – 4 jt	18 (5.5%)
> 4 jt	1 (0.3%)
jenis.tempat.tinggal	
Kos bersama-sama	14 (4.2%)
Kos Sendiri	52 (16%)
Rumah bersama saudara	23 (7.0%)
Rumah mengontrak bersama-sama	5 (1.5%)
Rumah mengontrak pribadi	3 (0.9%)
Rumah pribadi/rumah keluarga	233 (71%)
¹ n (%)	

Untuk menghitung IQV, kita bisa memanfaatkan perintah `function` yang mengambil masukan nilai-nilai yang ada di tanda kurungnya dan memprosesnya di dalam kurung kurawal `{}`. Perintah `function` selalu diakhiri dengan `return()` di dalamnya untuk menunjukkan hasil yang akan menjadi keluaran fungsi tersebut.

```
iqv <- function(x) {  
  # 1. Membersihkan data dari nilai yang hilang (NA)
```

```

x_clean <- x[!is.na(x)]

# 2. Membuat tabel frekuensi dari data yang bersih
counts <- table(x_clean)

# 3. Menghitung jumlah kategori (K)
K <- length(counts)

# 4. Kasus khusus: Jika hanya ada 1 kategori atau tidak ada data,
# maka tidak ada variasi, sehingga IQV = 0.
if (K <= 1) {
  return(0)
}

# 5. Menghitung jumlah total observasi (n)
n <- sum(counts)

# 6. Menghitung jumlah kuadrat dari proporsi setiap kategori ( $\sum p^2$ )
sum_p_sq <- sum((counts / n)^2)

# 7. Menerapkan formula IQV
#  $IQV = [K / (K - 1)] * [1 - \sum p^2]$ 
iqv_value <- (K / (K - 1)) * (1 - sum_p_sq)

# 8. Mengembalikan hasil perhitungan IQV
return(iqv_value)
}

```

Untuk menghasilkan nilai IQV suatu variabel, kita perlu memasukkan vektor yang berisi nilai-nilai dalam variabel dataset kita. Ini dapat dilakukan dengan sintaks `dataset$nama_variabel`. Tanda `$` berfungsi memberi tahu R untuk memilih variabel yang ada dalam dataset yang digunakan.

```

# Menampilkan vektor jenis.tempat.tinggal (opsional. Hapus komentar pada baris
# berikut jika ingin mencobanya)
# data.ubl$jenis.tempat.tinggal

# Menghitung IQV variabel jenis.tempat.tinggal
iqv(data.ubl$jenis.tempat.tinggal)

```

```
## [1] 0.5636143
```

2.2.2 Variabel Numerik

Perintah `tbl_summary` akan menentukan jenis statistik deskriptif yang ditampilkan secara intuitif berdasarkan dataset yang dibacanya. Perhatikan hasil dari `tbl_summary` berikut untuk variabel kepemilikan.mobil dan jarak.

```

data.ubl |>
  tbl_summary(include = c(kepemilikan.mobil,
                          jarak)) |>
  as_flex_table()

```

Characteristic	N = 330 ¹
kepemilikan.mobil	
0	107 (32%)
1	126 (38%)
2	80 (24%)
3	15 (4.5%)
4	2 (0.6%)
jarak	2.71 (1.31, 4.74)
¹ n (%); Median (Q1, Q3)	

Seperti yang kalian lihat, `tbl_summary` langsung menampilkan persentase tiap nilai yang ada di kepemilikan.mobil seolah-olah kepemilikan.mobil adalah variabel diskret. Hal ini terjadi karena nilai variabel kepemilikan.mobil adalah nilai numerik diskret, sehingga masing-masing nilai dikenali sebagai kategori.

Hal ini berbeda lagi dengan variabel jarak yang menampilkan statistik deskriptif untuk variabel numerik, yakni median, kuartil bawah dan kuartil atas (Q1 & Q3).

Untuk meminta `tbl_summary` menampilkan analisis statistik deskriptif untuk variabel numerik seperti mean, median, kuartil, dan standar deviasi (simpangan baku), kita dapat menyatakannya dalam atribut-atribut berikut.

```
data.ubl |>
  tbl_summary(include = c(kepemilikan.mobil,
                          kepemilikan.motor,
                          jarak),
              type = list(kepemilikan.mobil ~ "continuous", # untuk mengarahkan tbl_summary membaca tip
                          kepemilikan.motor ~ "continuous"),
              digits = list(kepemilikan.mobil ~ 0, # untuk menentukan jumlah desimal nilai yang ditampilkan
                           kepemilikan.motor ~ 0),
              statistic = list(kepemilikan.mobil ~ "{median} ({p25}, {p75})", # untuk menentukan jenis statistik
                              kepemilikan.motor ~ "{median}",
                              jarak ~ "{mean} ({sd})")) |>
  as_flex_table()
```

Characteristic	N = 330 ¹
kepemilikan.mobil	1 (0, 2)
kepemilikan.motor	1
jarak	3.35 (3.54)
¹ Median (Q1, Q3); Median; Mean (SD)	

Ketika meng-enter script, jangan salah memperhatikan urutan dan kelengkapan tanda koma pada fungsi

Untuk mengatur jenis statistik deskriptif yang ditampilkan, ubah bagian kanan tanda ~ pada atribut `statistic` sehingga apa yang ditulis di antara tanda petik menjadi template untuk ditampilkan di tabel. Adapun teknik statistik yang kita pakai harus kita tuliskan dalam kurung kurawalnya. Teknik-teknik statistik yang bisa kita gunakan di antaranya:

- `{mean}` : rata-rata
- `{median}` : median
- `{min}`, `{max}`: nilai minimum, maksimum
- `{p##}` : persentil `##`. Persentil adalah nilai yang membagi data menjadi seratus bagian. Karena kuartil membagi data menjadi empat bagian, maka untuk menampilkan kuartil bawah (Q1), yang berarti nilai yang membagi data menjadi 1/4 terbawah (25%), kita menuliskan `{p25}`. Begitu juga dengan Q3 yang membagi data menjadi 3/4 terbawah (75%), kita menuliskan `{p75}`
- `{sd}` : *standard deviation*/simpangan baku
- `{n}` : frekuensi kategori
- `{N}` : jumlah seluruhnya
- `{p}` : persentase

Pengenalan Tipe Data Baru R: List (`list`)

List pada dasarnya adalah tipe data nontunggal seperti vektor, hanya saja perbedaannya terletak pada tipe nilai yang dikandung. Vektor mewajibkan setiap elemen bernilai sama: string saja, angka saja, atau *boolean* saja misalnya.

Di sisi lain, *list* bisa memuat lebih dari satu jenis nilai. Ini sangat berguna ketika kita harus mendefinisikan sesuatu yang membutuhkan lebih dari satu nilai, seperti cara menampilkan statistik deskriptif untuk tiap-tiap variabel kita di fungsi `tbl_summary` di atas:

```
type = list(kepemilikan.motor ~ "continuous", kepemilikan.mobil ~ "continuous")
digits = list(kepemilikan.motor ~ 0, kepemilikan.mobil ~ 0)
```

- *list* dalam pengaturan atribut `type` berarti “anggap tipe variabel `kepemilikan.motor` sebagai *continuous*, begitu juga untuk variabel `kepemilikan.mobil`”
- *list* dalam pengaturan atribut `digits` berarti “atur agar nilai desimal untuk `kepemilikan.motor` adalah 0, begitu juga untuk variabel `kepemilikan.mobil`”

2.3 Soal Latihan

Impor dataset UIN RIL dan lakukan analisis statistik deskriptif untuk data-data yang ada di dalamnya STP-2.4:

- Persentase mahasiswa berdasarkan uang saku
- Median, kuartil atas dan kuartil bawah jumlah perjalanan per hari
- Mean dan simpangan baku biaya perjalanan per pekan
- IQV alasan memilih tempat tinggal

3 Modul-3: Visualisasi Data Kuantitatif

Setelah mempelajari modul ini, Anda diharapkan dapat:

1. memilih visualiasi yang tepat sesuai dengan variabel yang akan disajikan dan informasi yang ingin disampaikan **STP-3.1**
2. menginterpretasikan suatu visualiasi data kuantitatif secara mendalam **STP-3.2**
3. menjelaskan pentingnya menentukan tingkat pengukuran untuk sebuah variabel dari kaitannya dengan analisis statistik deskriptif dan diagram yang dipilih untuk menyajikan informasi **STP-3.4**
4. menghasilkan grafik yang tepat sesuai variabel yang akan disajikan **STP-3.3**

3.1 Visualisasi Data dengan ggplot2

ggplot2 adalah sebuah paket R yang dibuat oleh Hadley Wickham untuk membuat grafik dan visualisasi data. Paket ini didasarkan pada “Grammar of Graphics”, sebuah kerangka kerja yang memecah visualisasi menjadi komponen-komponen terpisah seperti data, sistem koordinat, dan elemen-elemen visual (geometries). Dengan pendekatan ini, Anda dapat membangun grafik lapis demi lapis (*layer by layer*).

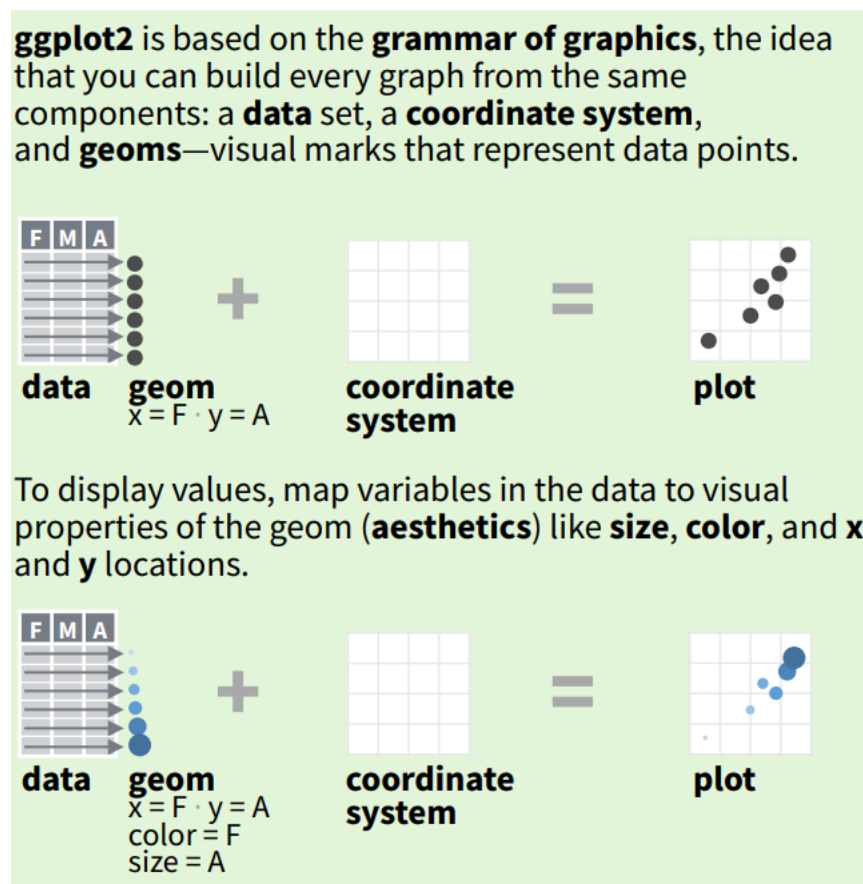


Figure 1: Konsep ggplot2

3.1.1 Mengimpor *Library* & Mengatur Dataset

Pertama, kita perlu memuat paket `tidyverse` yang sudah mencakup `ggplot2` untuk visualisasi dan `dplyr` serta `readr` untuk manipulasi data.

Tak lupa, kita juga akan menyertakan `openxlsx` dan `gtsummary` untuk mengolah data secara tabular jika sekiranya diperlukan dalam menunjang alur kerja kita.

```
library(tidyverse)
library(openxlsx)
library(gtsummary)
```

Selanjutnya, kita akan mengimpor *dataset* kita, yakni hasil kuesioner kepada mahasiswa UBL., seperti halnya praktikum-praktikum sebelumnya.

```
# Mengeset variabel tersendiri untuk nama file nama sheet}
file.dibaca <- "datasets/Data Praktikum 03.xlsx"
sheet.ubl <- "DataUtama_mhsUBL"

# Mengimpor file menjadi dataset memanfaatkan variabel nama file dan nama sheet yang kita buat tadi
data.ubl <- read.xlsx(file.dibaca, sheet = sheet.ubl)

# Pengenalan fungsi baru: 'mengintip' sejumlah baris pertama dari dataset kita
head(data.ubl)
```

```
##   Timestamp Kampus_PT Nomor.urut Jenis.Kelamin Umur          Fakultas          Prodi
## 1  45413.83      UBL           1    Perempuan  21 Fakultas Ilmu Sosial dan Politik Ilmu Komunikasi
## 2  45413.83      UBL           2     Laki-Laki  20          Fakultas Hukum      Ilmu Hukum
## 3  45413.85      UBL           3     Laki-Laki  21 Fakultas Ekonomi dan Bisnis      Manajemen
## 4  45413.85      UBL           4     Laki-Laki  22 Fakultas Ekonomi dan Bisnis      Akuntansi
## 5  45413.88      UBL           5     Laki-Laki  21 Fakultas Ekonomi dan Bisnis      Manajemen
## 6  45413.92      UBL           6    Perempuan  21 Fakultas Ekonomi dan Bisnis      Akuntansi
##      Uang.Saku kepemilikan.mobil kepemilikan.motor kepemilikan.sepeda
## 1      < 1 jt              1              2              1
## 2    1 jt - 2 jt              2              1              1
## 3    1 jt - 2 jt              1              2              2
## 4    1 jt - 2 jt              1              4              1 Kendaraan Bermotor (menumpang
## 5 2,1 jt - 3 jt              4              4              1
## 6    1 jt - 2 jt              1              2              1
##      kelurahan      jenis.tempat.tinggal      nama.jalan.tempat.tinggal
## 1    Kalibalau Kencana Rumah pribadi/rumah keluarga      eLBe Lound
## 2      Gunung sulah Rumah pribadi/rumah keluarga      Jl.Urip Sumoharjo no 8
## 3      langkapura Rumah pribadi/rumah keluarga      JL DARUSSALAM GG LANGGAR LK 1
## 4 Bilabong JAYA JAYA JAYA Rumah pribadi/rumah keluarga Jl Darussalam bilabong bila bolong di jait dor
## 5      sukarama Rumah pribadi/rumah keluarga      jl.cendrawasih sukarama bandar lampun
## 6      Way halim Rumah pribadi/rumah keluarga      Jl P tabuan nomor 2
##      alasan.pemilihan.lokasi.tempat.tinggal biaya.dalam.seminggu Jumlah.perjalanan.Senin Jumlah
## 1      Bersama keluarga/saudara/teman      25      2
## 2 Mudahnya akses berpergian dari tempat tinggal      200      3
## 3      Bersama keluarga/saudara/teman      50      1
## 4    Lingkungan nyaman karna aman dari kejahatan      20      2
## 5      Dekat dengan fasilitas umum      NA      1
## 6      Bersama keluarga/saudara/teman      40      3
##      Jumlah.Perjalanan.Kamis Jumlah.Perjalanan.Jumat Jumlah.Perjalanan.Sabtu Jumlah.Perjalanan.Ahad
```

## 1	3	2	2	2
## 2	3	3	3	3
## 3	1	1	1	1
## 4	2	2	2	2
## 5	1	1	1	1
## 6	1	1	1	1

Kemudian kita perlu menetapkan **factor** untuk variabel-variabel kategoris kita agar data kita lebih ‘bersih.’

```
# Menetapkan vektor untuk factor variabel kategoris
jk <- c("Laki-Laki", "Perempuan")

fakultas <- c("Fakultas Ekonomi dan Bisnis", "Fakultas Hukum",
              "Fakultas Ilmu Komputer", "Fakultas Ilmu Sosial dan Politik",
              "Fakultas Keguruan dan Ilmu Pendidikan", "Fakultas Teknik")

prodi <- c("Administrasi Bisnis", "Administrasi Publik", "Akuntansi", "Arsitektur",
           "Ilmu Hukum", "Ilmu Komunikasi", "Informatika", "Manajemen",
           "Pendidikan Bahasa Inggris", "Sistem Informasi")

tingkat <- c("1 (Semester 1 - Semester 2)",
             "2 (Semester 3 - Semester 4)",
             "3 (Semester 5 - Semester 6)",
             "4 (Semester 7 - Semester 8)")

uang_saku <- c("< 1 jt", "1 jt - 2 jt", "2,1 jt - 3 jt", "3,1 jt - 4 jt", "> 4 jt")

# Merapikan dataset
data.ubl <- data.ubl |>
  # mengubah nama variabel
  rename(biaya.dalam.sepekan = biaya.dalam.seminggu) |>
  rename(Jumlah.Perjalanan.Senin = Jumlah.perjalanan.Senin) |>
  rename(jarak = `jarak.(km)`) |>
  # menghapus data dengan missing values
  drop_na() |>
  # menambahkan factor ke variabel kategoris
  mutate(Jenis.Kelamin = factor(Jenis.Kelamin, levels = jk),
         Fakultas = factor(Fakultas, levels = fakultas),
         Prodi = factor(Prodi, levels = prodi),
         Tingkat.Semester = factor(Tingkat.Semester,
                                    levels = tingkat,
                                    ordered = TRUE),
         Uang.Saku = factor(Uang.Saku, levels = uang_saku, ordered = TRUE))
```

Kita akan mengubah kategori **factor** pada variabel **Tingkat.Semester** dan **Uang.Saku** agar lebih mudah dibaca saat divisualisasikan. Kita akan menggunakan fungsi dari **dplyr** (bagian dari **tidyverse**) untuk membersihkan dan mengubah data.

```
# Membersihkan dan mengubah nama level pada variabel Tingkat.Semester dan Uang.Saku
data.ubl.cleaned <- data.ubl %>%
  mutate(
    Tingkat.Semester = fct_recode(Tingkat.Semester,
```



```
## Warning: There were 2 warnings in `mutate()`.
## The first warning was:
## i In argument: `Tingkat.Semester = fct_recode(...)`.
```

```
## Rows: 330
## Columns: 26
## $ Timestamp      <dbl> 45413.83, 45413.83, 45413.85, 45413.85, 45413.92, 454
## $ Kampus_PT      <chr> "UBL", "UBL", "UBL", "UBL", "UBL", "UBL", "UBL", "UBL
## $ Nomor.urut      <dbl> 1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 14, 15, 16, 17, 18, 1
## $ Jenis.Kelamin   <fct> Perempuan, Laki-Laki, Laki-Laki, Laki-Laki, Perempuan
## $ Umur            <dbl> 21, 20, 21, 22, 21, 22, 22, 22, 22, 22, 19, 22, 2
## $ Fakultas        <fct> Fakultas Ilmu Sosial dan Politik, Fakultas Hukum, Faku
## $ Prodi           <fct> Ilmu Komunikasi, Ilmu Hukum, Manajemen, Akuntansi, Aku
## $ Tingkat.Semester <ord> Semester 7 & 8, Semester 5 & 6, Semester 7 & 8, Semes
## $ Uang.Saku       <ord> "< 1 jt", "1 jt - 2 jt", "1 jt - 2 jt", "1 jt -
2 jt", "1 jt - 2 jt", "1 jt - 2 jt", "< 1 jt"~
## $ kepemilikan.mobil <dbl> 1, 2, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1,
## $ kepemilikan.motor <dbl> 2, 1, 2, 4, 2, 1, 1, 1, 1, 2, 1, 1, 1, 0, 1, 0, 2, 2,
## $ kepemilikan.sepeda <dbl> 1, 1, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 3,
## $ kendaraan.utama  <chr> "Sepeda Motor Pribadi", "Mobil Pribadi", "Sepeda Moto
## $ kelurahan        <chr> "Kalibalau Kencana", "Gunung sulah", "langkapura", "B
## $ jenis.tempat.tinggal <chr> "Rumah pribadi/rumah keluarga", "Rumah pribadi/rumah l
## $ nama.jalan.tempat.tinggal <chr> "eLBe Laundry", "Jl.Urip Sumoharjo no 88", "JL DARUSSA
## $ jarak            <dbl> 4.2237967, 2.7033310, 3.2370722, 3.0023362, 2.9253306
## $ alasan.pemilihan.lokasi.tempat.tinggal <chr> "Bersama keluarga/saudara/teman", "Mudahnya akses berj
## $ biaya.dalam.sepekan <dbl> 25, 200, 50, 20, 40, 70, 35, 50, 350, 50, 30, 35, 50,
## $ Jumlah.Perjalanan.Senin <dbl> 2, 3, 1, 2, 3, 2, 4, 4, 3, 2, 4, 2, 2, 2, 1, 1, 4, 4,
## $ Jumlah.Perjalanan.Selasa <dbl> 3, 3, 1, 2, 3, 2, 4, 4, 3, 1, 4, 1, 3, 3, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Rabu <dbl> 2, 3, 1, 2, 1, 2, 4, 4, 3, 2, 4, 3, 2, 4, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Kamis <dbl> 3, 3, 1, 2, 1, 2, 4, 4, 3, 1, 4, 3, 2, 2, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Jumat <dbl> 2, 3, 1, 2, 1, 2, 4, 3, 3, 2, 4, 3, 2, 3, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Sabtu <dbl> 2, 3, 1, 2, 1, 2, 3, 4, 2, 1, 1, 1, 2, 1, 1, 1, 3, 1,
## $ Jumlah.Perjalanan.Ahad <dbl> 2, 3, 1, 2, 1, 2, 3, 4, 1, 1, 1, 1, 1, 2, 1, 1, 3, 1,
```

49

3.1.2 Tata Tulis Grafik (*Grammar of Graphics*)

Setiap grafik `ggplot2` terdiri dari beberapa komponen kunci:

- **DATA**: Dataset yang ingin Anda visualisasikan.
- **MAPPING**: `aes()` (aesthetics), yang menghubungkan variabel dari data Anda ke properti visual dari grafik (misalnya, sumbu x, sumbu y, warna, ukuran).
- **GEOM_FUNCTION**: Objek geometris yang merepresentasikan data (misalnya, `geom_point()` untuk scatter plot, `geom_bar()` untuk diagram batang).
- **STAT**: Transformasi statistik. Setiap `geom` memiliki statistik default (misalnya, `geom_bar` secara default menggunakan `stat_count`), tetapi Anda bisa menentukannya secara manual.
- **POSITION**: Penyesuaian posisi untuk `geom` yang tumpang tindih (misalnya, `position_dodge()` atau `position_stack()`).
- **COORDINATE_FUNCTION**: Sistem koordinat yang digunakan (`coord_cartesian`, `coord_flip`, dll.).
- **FACET_FUNCTION**: Membagi plot menjadi beberapa sub-plot berdasarkan variabel kategori (`facet_wrap` atau `facet_grid`).

```
ggplot(<DATA>) +  
  <GEOM_FUNCTION>(mapping = aes(<MAPPING>),  
    stat = <STAT>,  
    position = <POSITION>) +  
  <COORDINATE_FUNCTION> +  
  <FACET_FUNCTION> +  
  <SCALE_FUNCTION> + # optional  
  <THEME_FUNCTION> # optional
```

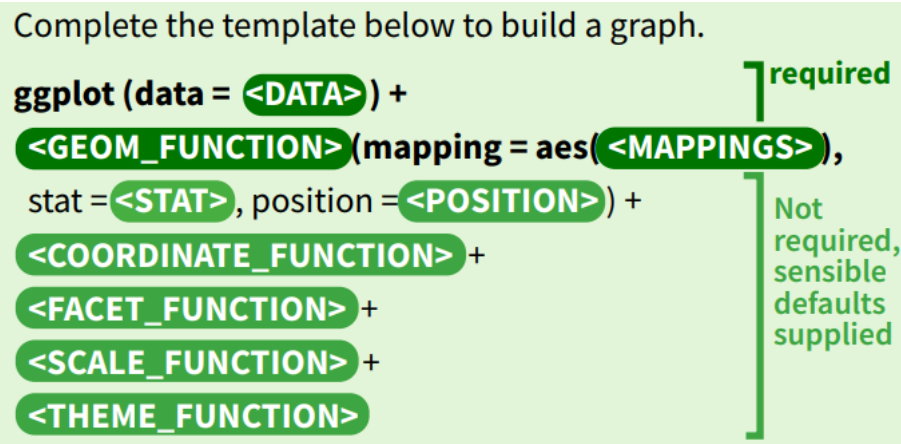


Figure 2: “Grammar of Graphics”

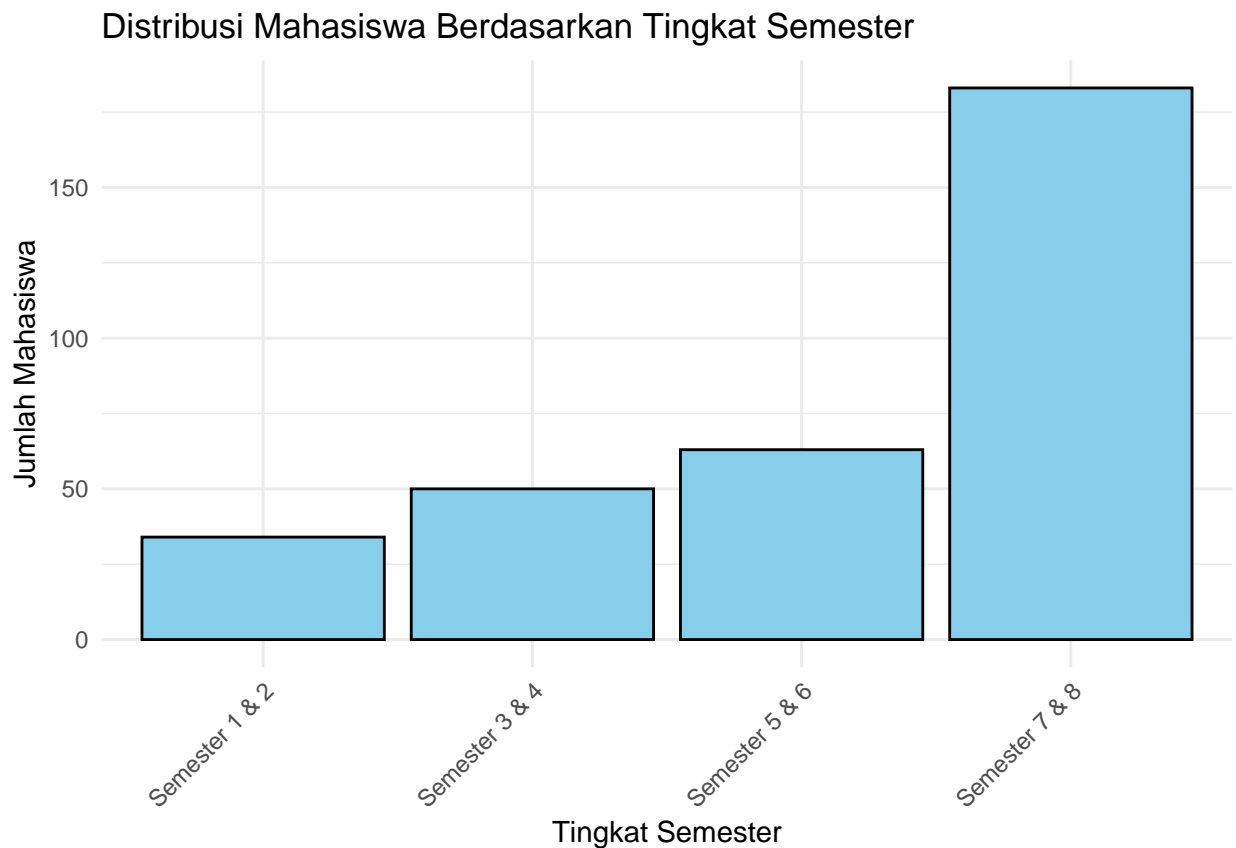
3.1.3 Praktik Visualisasi Data

3.1.3.1 Diagram Batang (*Column/Bar Chart*) Diagram batang sangat baik untuk menampilkan distribusi atau perbandingan data **kategoris**.

3.1.3.1.1 Diagram Batang Tunggal Mari kita lihat distribusi mahasiswa berdasarkan tingkat semester. `geom_bar()` secara otomatis menghitung jumlah observasi untuk setiap kategori di sumbu x.

```
diagram_batang <- ggplot(data.ubl.cleaned) +
  geom_bar(mapping = aes(x = Tingkat.Semester), fill = "skyblue",
    color = "black") +
  labs(
    title = "Distribusi Mahasiswa Berdasarkan Tingkat Semester",
    x = "Tingkat Semester",
    y = "Jumlah Mahasiswa"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotasi label x agar tidak tumpang tindih

diagram_batang
```



Interpretasi: Grafik di atas menunjukkan bahwa mayoritas responden mahasiswa berasal dari tingkat semester 5 & 6, diikuti oleh semester 7 & 8.

Penjelasan Sintaks (Grammar of Graphics):

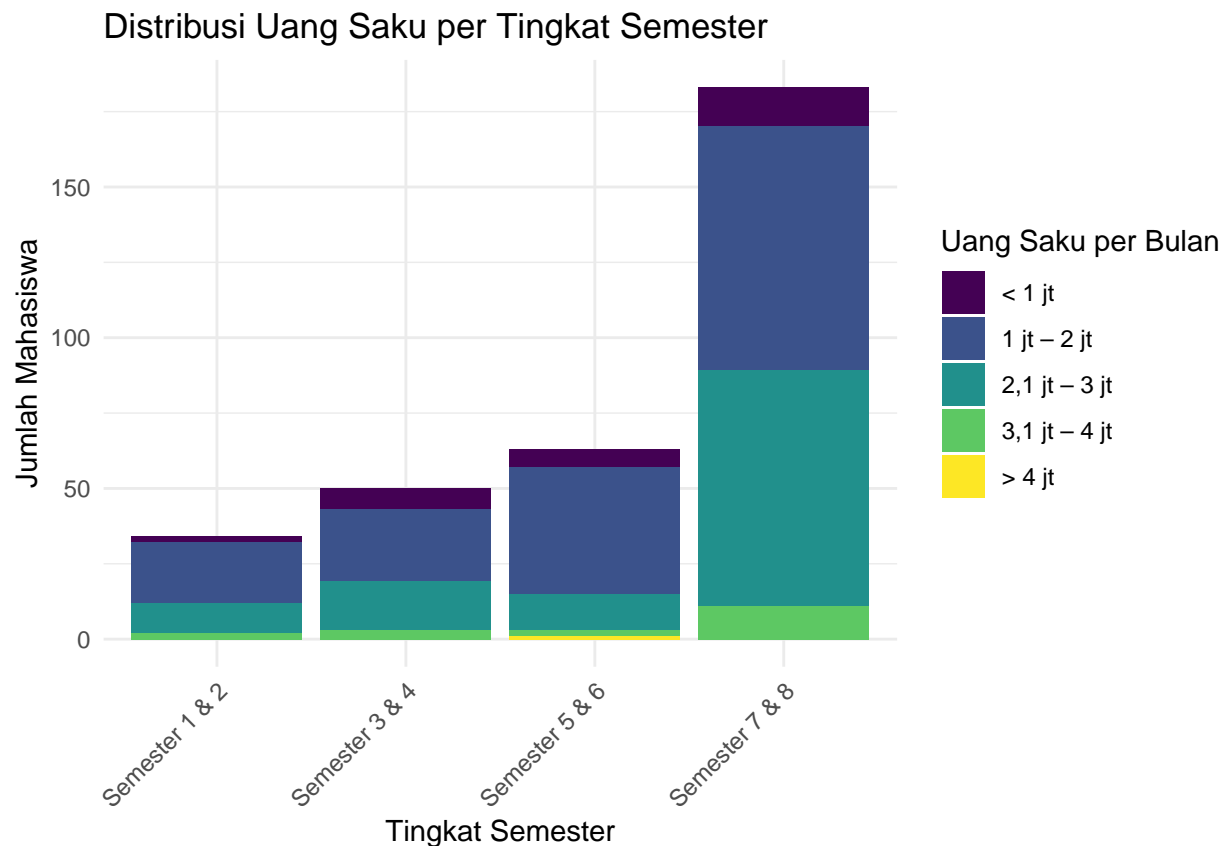
- **DATA:** `ggplot(data.ubl.cleaned)` mendefinisikan dataset yang digunakan.
- **GEOM:** `geom_bar(...)` menentukan bentuk geometris yang digunakan, yaitu batang.
- **MAPPING:** `mapping = aes(x = Tingkat.Semester)` memetakan variabel `Tingkat.Semester` dari data ke sumbu x pada grafik.
- **STAT:** `geom_bar()` secara default menggunakan `stat = "count"`, yang berarti ia secara otomatis melakukan transformasi statistik dengan menghitung jumlah baris untuk setiap kategori `Tingkat.Semester` dan menampilkannya sebagai ketinggian batang di sumbu y.

- `fill = "skyblue", color = "black"`: Ini adalah pengaturan properti visual, bukan pemetaan. Kita mengatur semua batang agar memiliki warna isian “skyblue” dan garis tepi “black”.
- `labs(...), theme_minimal(), theme(...)`: Ini adalah lapisan tambahan untuk kustomisasi label dan tema, bukan bagian inti dari “grammar”.

3.1.3.1.2 Diagram Batang Bertumpuk (*Stacked*) Kita bisa menambahkan variabel lain, misalnya `Uang.Saku`, ke dalam `aes()` dengan properti `fill` untuk membuat diagram batang bertumpuk. Ini menunjukkan proporsi uang saku di setiap tingkat semester.

```
diagram_batangTumpuk <- ggplot(data.ubl.cleaned) +
  geom_bar(mapping = aes(x = Tingkat.Semester, fill = Uang.Saku)) +
  labs(
    title = "Distribusi Uang Saku per Tingkat Semester",
    x = "Tingkat Semester",
    y = "Jumlah Mahasiswa",
    fill = "Uang Saku per Bulan"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

diagram_batangTumpuk
```



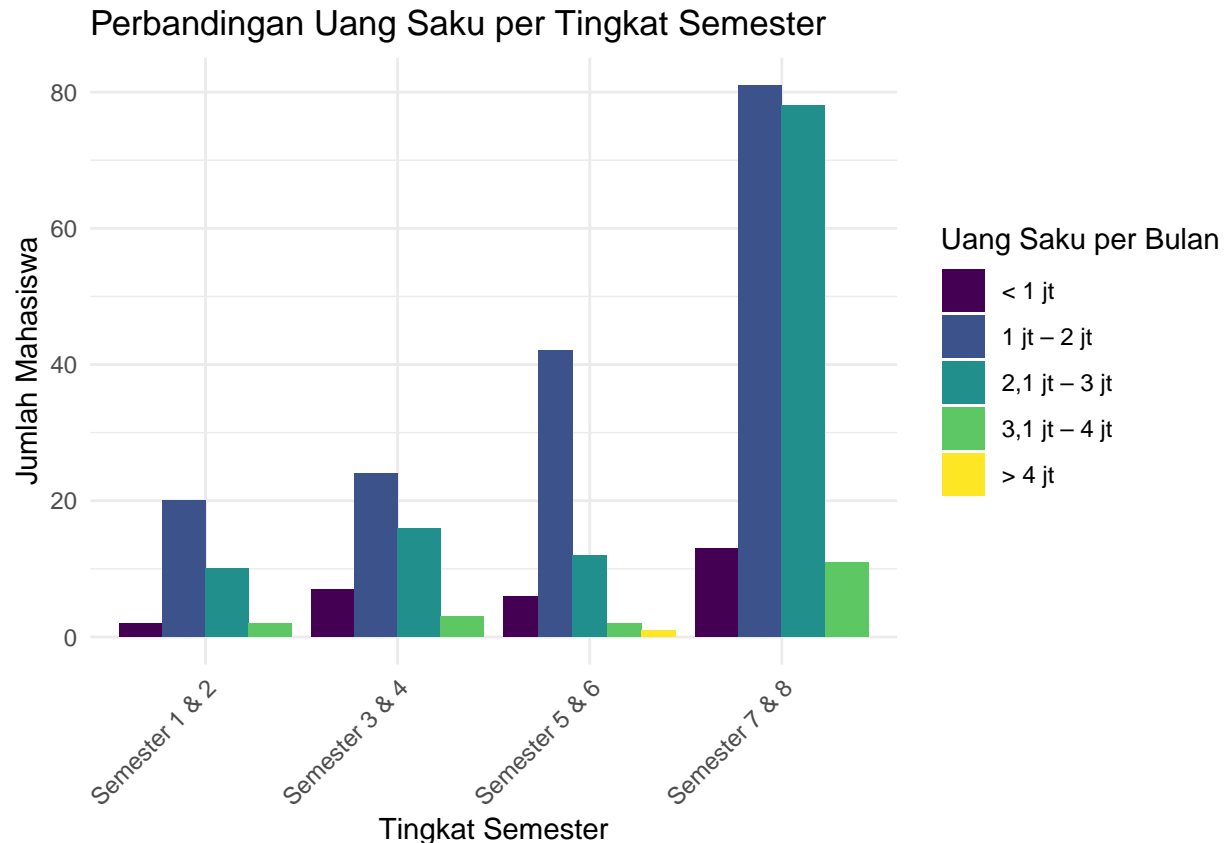
Interpretasi: Dari grafik ini, kita bisa melihat komposisi uang saku di setiap angkatan. Misalnya, pada tingkat “Semester 5 & 6”, sebagian besar mahasiswa memiliki uang saku antara 1 juta hingga 2 juta. Hal ini kita ketahui dari perbandingan relatif tinggi porsi warna-warna dalam masing-masing batang.

Penjelasan Sintaks (Grammar of Graphics):

- **MAPPING:** `mapping = aes(x = Tingkat.Semester, fill = Uang.Saku)` kini memiliki pemetaan tambahan. Selain sumbu x, kita juga memetakan variabel `Uang.Saku` ke properti visual `fill` (warna isian). `ggplot` akan membuat segmen berwarna berbeda di dalam setiap batang sesuai kategori uang saku.
- **POSITION:** Secara default, `geom_bar()` menggunakan `position = "stack"` ketika `fill` dipetakan ke sebuah variabel. Inilah yang menyebabkan segmen-segmen tersebut ditumpuk di atas satu sama lain.
- **labs(fill = "Uang Saku per Bulan"):** Argumen `fill` di dalam `labs()` berfungsi untuk mengubah judul dari legenda yang secara otomatis dibuat dari pemetaan `fill`.

Untuk membandingkan jumlah absolut antar kategori uang saku, diagram batang berkelompok lebih efektif. Kita gunakan `position = "dodge"`.

```
diagram_batangSebar <- ggplot(data.ubl.cleaned) +  
  geom_bar(mapping = aes(x = Tingkat.Semester, fill = Uang.Saku),  
            position = "dodge") +  
  labs(  
    title = "Perbandingan Uang Saku per Tingkat Semester",  
    x = "Tingkat Semester",  
    y = "Jumlah Mahasiswa",  
    fill = "Uang Saku per Bulan"  
  ) +  
  theme_minimal() +  
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  
  
diagram_batangSebar
```



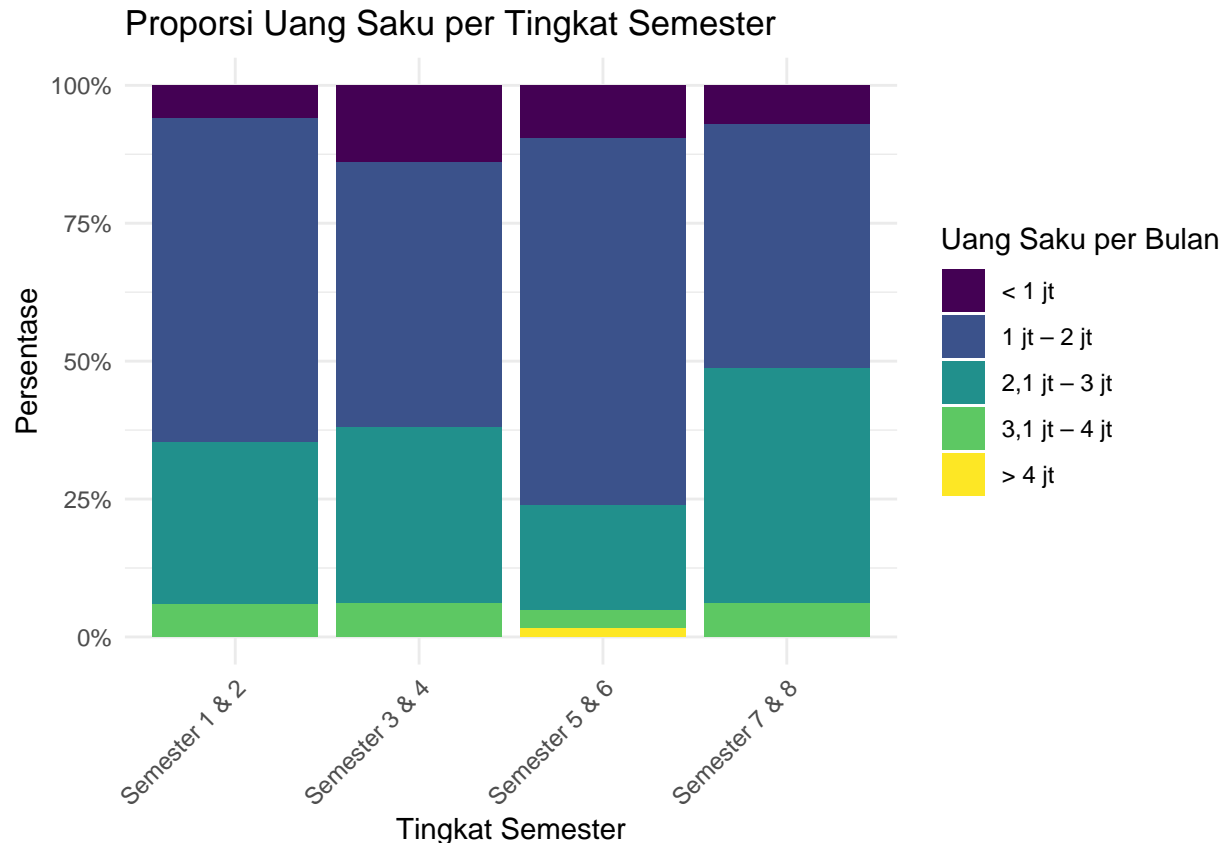
Interpretasi: Grafik ini mempermudah perbandingan langsung. Terlihat jelas bahwa kategori uang saku “1 jt – 2 jt” mendominasi di hampir semua tingkat semester.

- **POSITION:** Komponen `position` diubah secara eksplisit menjadi `position = "dodge"`. Ini menginstruksikan `ggplot` untuk menempatkan batang-batang yang memiliki kategori x yang sama (misalnya, “Semester 5 & 6”) bersebelahan, bukan menumpuknya. Ini memungkinkan perbandingan langsung antar kategori `Uang.Saku`.

Terkadang kita lebih tertarik pada perbandingan proporsi antar grup daripada jumlah absolutnya. Dengan mengubah `position` menjadi `"fill"`, kita dapat membuat setiap batang memiliki tinggi yang sama (100%) dan menunjukkan persentase relatif dari setiap subgroup.

```
diagram_btTumpuk100 <- ggplot(data.ubl.cleaned) +
  geom_bar(mapping = aes(x = Tingkat.Semester, fill = Uang.Saku),
           position = "fill") +
  scale_y_continuous(labels = scales::percent) + # untuk mengubah satuan sumbu Y menjadi '%'
  labs(
    title = "Proporsi Uang Saku per Tingkat Semester",
    x = "Tingkat Semester",
    y = "Persentase",
    fill = "Uang Saku per Bulan"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

diagram_btTumpuk100
```



Interpretasi: Grafik ini menunjukkan bahwa secara proporsional, mahasiswa dengan kategori uang saku tertinggi (“>4jt”) yang paling dominan terdapat pada tingkat 3 (“Semester 5 & 6”). Sementara itu, mahasiswa kategori uang saku terendah paling banyak porsinya pada tingkat 2 (“Semester 3 & 4”). Ini adalah wawasan yang mungkin tidak terlihat jelas pada grafik jumlah absolut.

Penjelasan Sintaks (Grammar of Graphics):

- **POSITION:** Komponen `position` diubah menjadi `position = "fill"`. Pengaturan ini secara otomatis melakukan transformasi **STAT** yang berbeda: ia menghitung proporsi dari setiap subgrup (Uang.Saku) dalam setiap grup (Tingkat.Semester). Hasilnya adalah setiap batang dinormalisasi menjadi setinggi 1 (atau 100%).
- **SCALE:** `scale_y_continuous(labels = scales::percent)` adalah lapisan tambahan yang mengontrol **SKALA** pada sumbu y. Fungsi `scales::percent` digunakan untuk memformat label sumbu dari angka desimal (misal: 0.5) menjadi format persentase (misal: 50%) agar lebih mudah dibaca.

3.1.4 Diagram Lollipop

Diagram lolipop adalah alternatif dari diagram batang yang dapat mengurangi tinta visual dan memberikan penekanan lebih pada nilai data. Grafik ini menggunakan segmen garis dan titik untuk merepresentasikan nilai. Ini sangat efektif untuk menampilkan data kategoris yang banyak kategorinya.

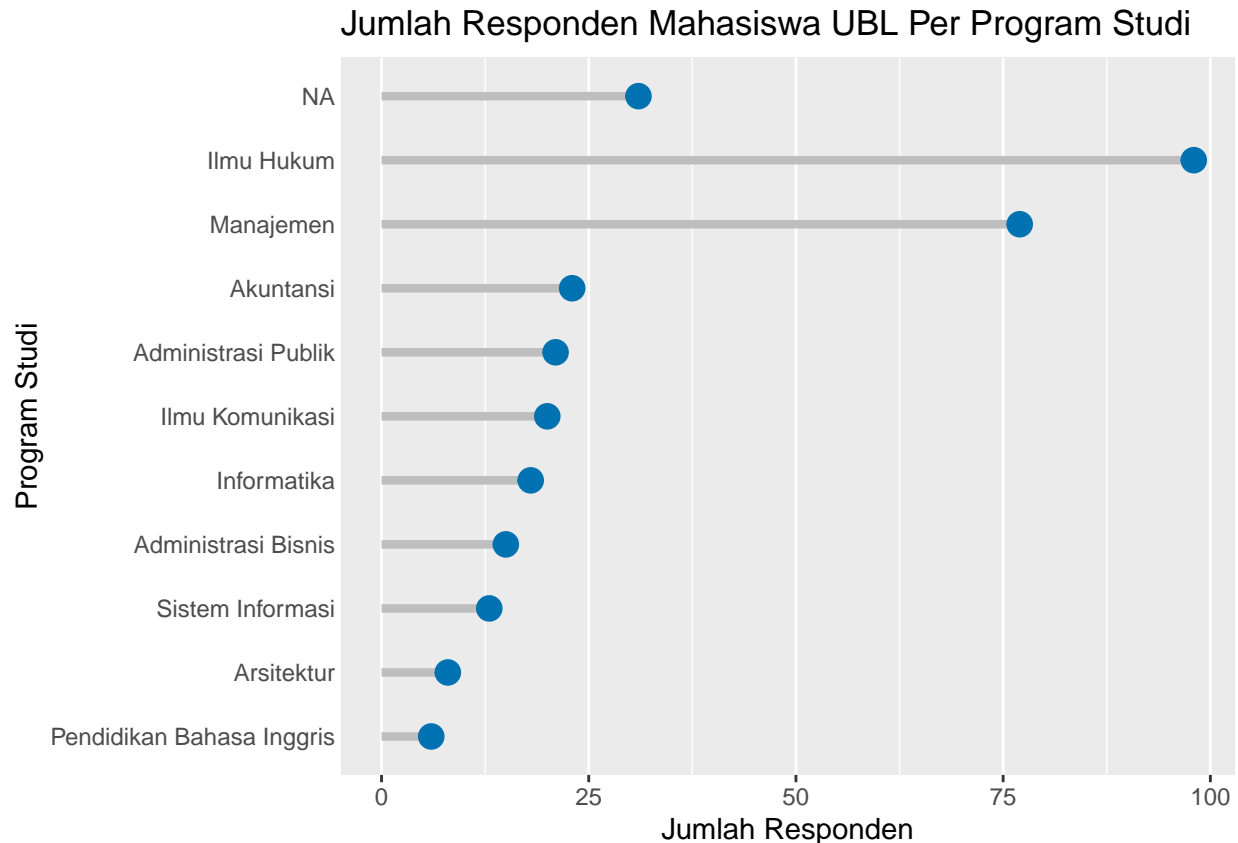
```
# Pertama, kita perlu membuat tabel baru yang menampilkan jumlah mahasiswa per program studi
prodi_count <- data.ubl.cleaned |>
  count(Prodi, name = "jumlah") |>
  arrange(jumlah) |>
  mutate(Prodi = fct_inorder(Prodi)) # mengubah jadi faktor terurut
```

```

# Kedua, kita baru bisa membuat diagram lolipop-nya
diagram_lollipop <- ggplot(prodi_count) +
  # diagram lolipop terdiri atas 2 geometri: geom_segment yang bertindak sebagai
  # batang dan geom_point yang bertindak sebagai permennya
  geom_segment(
    mapping = aes(x = Prodi,
                  y = 0, yend = jumlah),
    color = "grey",
    size = 1.5) +
  geom_point(
    mapping = aes(x = Prodi, y = jumlah),
    color = "#0072B2", size = 4
  ) +
  coord_flip() + # Membalik sumbu agar mudah dibaca
  labs(
    title = "Jumlah Responden Mahasiswa UBL Per Program Studi",
    x = "Program Studi",
    y = "Jumlah Responden"
  ) +
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
    axis.ticks.y = element_blank()
  )

diagram_lollipop

```

Interpretasi: Grafik ini menunjukkan bahwa kebanyakan responden berasal dari program studi Ilmu Hukum dan Manajemen, dengan perbandingan yang cukup timpang dengan prodi-prodi lain. Selain itu, ternyata cukup dominan mahasiswa yang tidak memberikan data prodi mereka (NA) di antara prodi-prodi selain dua yang tertinggi tadi.

Penjelasan Sintaks (Grammar of Graphics):

- **DATA & STAT:** Sama seperti contoh sebelumnya, kita melakukan pra-pemrosesan data menggunakan `dplyr`. Kita mengelompokkan data berdasarkan **Prodi**, menghitung (`count`) jumlahnya, mengurutkan (`arrange`), dan yang terpenting, mengubah **Prodi** menjadi variabel faktor yang terurut (`fct_inorder`) agar plot ditampilkan sesuai urutan yang kita inginkan.
- **GEOM & MAPPING (Layering):** Di sinilah keunikan diagram lolipop. Kita menggunakan **dua** lapisan **geom**:
 - `geom_segment()`: Digunakan untuk membuat “batang” atau segmen garis. **Mapping**-nya membutuhkan empat estetika: `x` dan `xend` (yang sama untuk garis vertikal) serta `y` (titik awal, yaitu 0) dan `yend` (titik akhir, yaitu `jumlah`).
 - `geom_point()`: Digunakan untuk membuat “permen” atau titik di ujung segmen. **Mapping**-nya lebih sederhana, hanya membutuhkan `x` dan `y`.
- **KOORDINAT:** `coord_flip()` digunakan untuk membalik sumbu, membuat diagram lolipop horizontal yang seringkali lebih mudah dibaca label kategorinya.
- **THEME:** Lapisan tema digunakan untuk membersihkan tampilan, seperti menghilangkan beberapa garis grid dan batas panel untuk menonjolkan data itu sendiri.

3.1.5 Diagram Pai/Donat (*Pie/Donut Chart*)

Diagram pai (dan variasinya, diagram donat) digunakan untuk menunjukkan proporsi dari sebuah keseluruhan. Meskipun populer, diagram pai seringkali sulit untuk dibaca secara akurat, terutama ketika ada banyak irisan atau ukurannya mirip. Diagram donat sedikit lebih baik karena mengurangi penekanan pada sudut dan lebih fokus pada panjang busur.

Di `ggplot2`, diagram pai dibuat dengan memulai dari diagram batang bertumpuk, lalu mengubah sistem **KOORDINAT**-nya menjadi koordinat polar. Akan tetapi, dalam `ggplot2` kita tidak bisa membuat diagram pai langsung dari datasetnya, tetapi kita harus membentuk **tabel distribusi frekuensinya** terlebih dahulu.

```
fakultas_count <- data.ubl.cleaned |>
  count(Fakultas, name = "jumlah") |> # Membuat kolom jumlah responden per fakultas
  mutate(
    persen = jumlah / sum(jumlah) # Membuat kolom persentase dari 'jumlah'
  )

fakultas_count
```

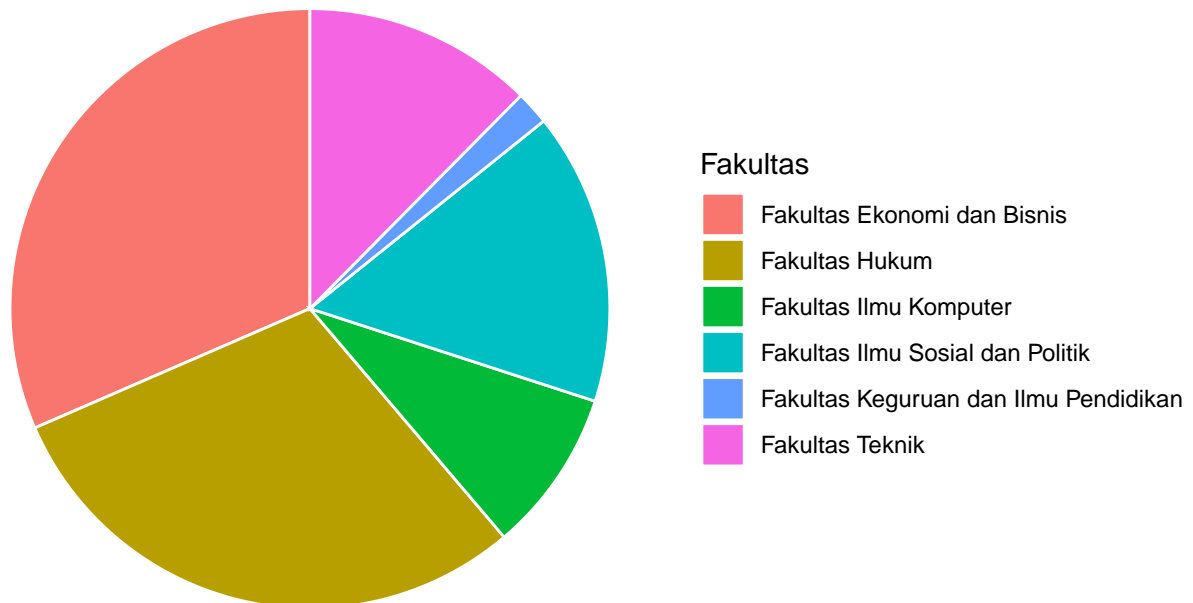
	Fakultas	jumlah	persen
## 1	Fakultas Ekonomi dan Bisnis	104	0.31515152
## 2	Fakultas Hukum	98	0.29696970
## 3	Fakultas Ilmu Komputer	29	0.08787879
## 4	Fakultas Ilmu Sosial dan Politik	52	0.15757576
## 5	Fakultas Keguruan dan Ilmu Pendidikan	6	0.01818182
## 6	Fakultas Teknik	41	0.12424242

Baru kemudian kita bisa menghasilkan perintah `ggplot` untuk

```
diagram_pai <- ggplot(fakultas_count, aes(x = 2, y = persen, fill = Fakultas)) +
  geom_bar(stat = "identity", color = "white") +
  coord_polar(theta = "y", start = 0) +
  labs(
    title = "Sebaran Fakultas Responden Mahasiswa UBL",
    fill = "Fakultas"
  ) +
  # Membersihkan tema
  theme_void() +
  theme(legend.position = "right")

diagram_pai
```

Sebaran Fakultas Responden Mahasiswa UBL



Interpretasi: Diagram donat ini menunjukkan proporsi dari setiap fakultas. Terlihat jelas bahwa irisan “Fakultas Ekonomi dan Bisnis” dan “Fakultas Hukum” mendominasi porsi responden.

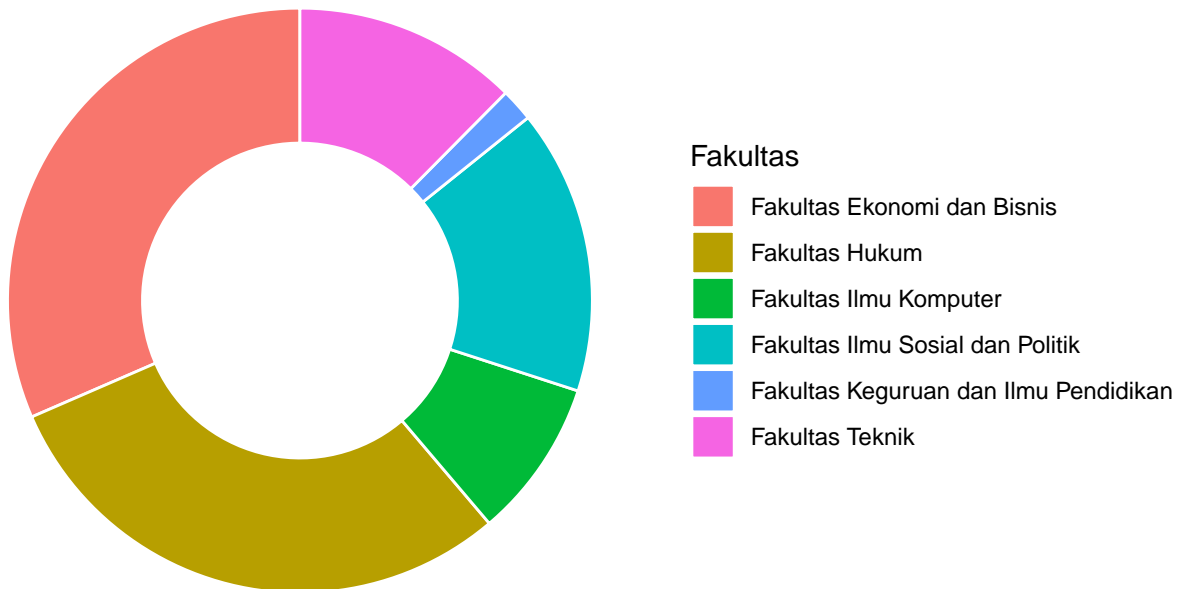
Penjelasan Sintaks (Grammar of Graphics):

- **DATA & STAT:** Kita menggunakan data `alasan_counts` yang sudah diagregasi, lalu menambahkan kolom baru untuk persentase (`persen`) dan posisi vertikal untuk label (`posisi_y_label`).
- **GEOM & MAPPING:** Kita mulai dengan `geom_bar(stat = "identity")` yang membuat diagram batang di mana tinggi batang (`y`) adalah nilai persentase itu sendiri. `x=2` adalah trik untuk membuat satu batang tunggal yang akan kita “lilit”.
- **KOORDINAT:** `coord_polar(theta = "y")` adalah komponen kunci. Ini mengambil diagram batang dan mengubah sistem koordinatnya dari Kartesius (`x,y`) menjadi Polar. Sumbu `y` “dibengkokkan” menjadi sebuah lingkaran.
- `theme_void()`: Menghilangkan semua elemen tema seperti sumbu, label sumbu, dan latar belakang, yang tidak relevan untuk diagram pai/donat.
- kita bisa menambahkan lubang di tengah pai dengan menambahkan `xlim(0.5, 2.5)` yang merupakan batas dalam dan batas luar dari radius si diagram seperti berikut.

```
diagram_donat <- diagram_pai + xlim(0.5, 2.5)

diagram_donat
```

Sebaran Fakultas Responden Mahasiswa UBL



Penting

Komunitas perupa data pada dasarnya menganjurkan kita untuk ‘menghindari’ diagram lingkaran. Hal ini bisa kalian baca di laman ini. Alternatifnya, mereka lebih menyarankan kita untuk menggunakan diagram batang atau diagram lollipop saja

3.1.6 Diagram *Treemap*

Treemap adalah alternatif lain untuk diagram pai, terutama efektif ketika Anda memiliki banyak kategori. Treemap menampilkan data hierarkis atau bagian-ke-keseluruhan sebagai satu set persegi panjang bersarang. Ukuran setiap persegi panjang sebanding dengan nilainya.

Untuk membuat treemap, kita perlu paket tambahan yaitu `treemapify`.

```
# Pastikan paket sudah terinstall: install.packages("treemapify")
# install.packages("treemapify")
library(treemapify)
```

Kita akan membuat *treemap* dari alasan mahasiswa memilih tempat tinggal mereka

```
# Pertama, kita siapkan data dengan menghitung jumlah dan mengurutkannya
alasan_counts <- data.ubl.cleaned |>
# Mengganti nama yang terlalu panjang agar muat di plot
mutate(alasan_singkat = fct_recode(`alasan.pemilihan.lokasi.tempat.tinggal`,
  "Bersama Keluarga" = "Bersama keluarga/saudara/teman",
```

```

"Dekat Kampus/lokasi lain" = "Dekat dengan kampus",
"Dekat Kampus/lokasi lain" = "Dekat dengan fasilitas umum",
"Dekat Kampus/lokasi lain" = "Mudahnya akses berpergian dari tempat tinggal ",
"Fasilitas Lengkap" = "Fasilitas tempat tinggal lengkap",
"Murah" = "Biaya tempat tinggal murah",
"Aman" = "Lingkungan nyaman karna aman dari kejahatan"
)) |>
count(alasan_singkat, name = "jumlah")

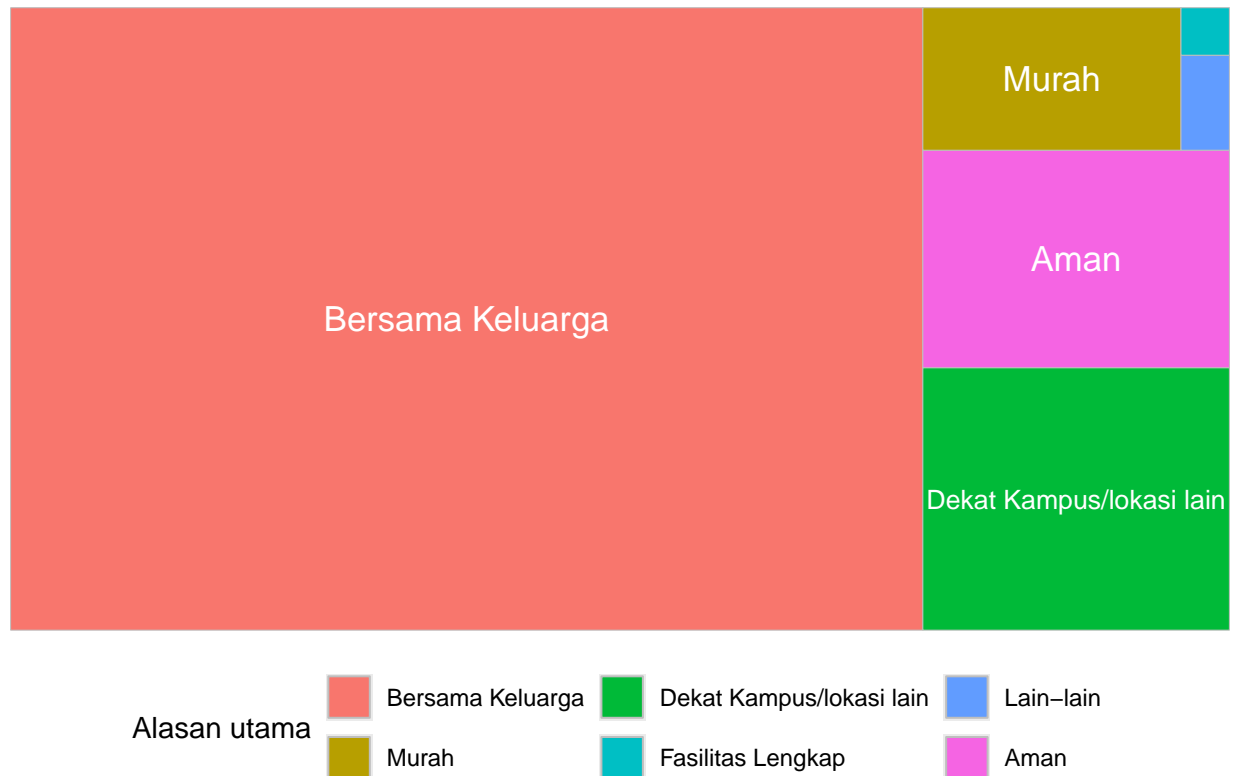
alasan_counts

##           alasan_singkat jumlah
## 1      Bersama Keluarga    247
## 2                Murah     16
## 3 Dekat Kampus/lokasi lain    35
## 4      Fasilitas Lengkap      1
## 5                Lain-lain      2
## 6                Aman      29

ggplot(alasan_counts, aes(area = jumlah, fill = alasan_singkat, label = alasan_singkat)) +
  geom_treemap() +
  geom_treemap_text(
    colour = "white",
    place = "centre",
    size = 13
  ) +
  labs(
    title = "Proporsi Alasan Mahasiswa Memilih Tempat Tinggal (Treemap)",
    fill = "Alasan utama"
  ) +
  theme(legend.position = "bottom") # Untuk menampilkan area yang tidak ada labelnya

```

Proporsi Alasan Mahasiswa Memilih Tempat Tinggal (Treemap)



Penjelasan Sintaks (Grammar of Graphics):

- **DATA:** Kita menggunakan data `alasan_counts` yang sudah diagregasi.
- **GEOM & MAPPING:** Paket `treemapify` menyediakan `geom` baru yang terintegrasi dengan `ggplot2`.
 1. `geom_treemap()`: Ini adalah `geom` utama. Alih-alih `x` dan `y`, **MAPPING** utamanya adalah `aes(area = jumlah)`. `ggplot` akan secara otomatis menghitung tata letak persegi panjang berdasarkan nilai `jumlah`. Kita juga memetakan `alasan_singkat` ke `fill` untuk warna dan `label` untuk teks.
 2. `geom_treemap_text()`: Ini adalah `geom` tambahan khusus untuk menempatkan teks di dalam setiap area treemap.
- **THEME:** Kita bisa menyembunyikan legenda karena setiap area sudah diberi label secara langsung, sehingga legenda menjadi berlebihan. Caranya adalah mengatur nilai `bottom` = menjadi `"none"`. Akan tetapi, untuk kasus kita, kita punya area yang terlalu kecil untuk diberi label, sehingga kita tetap tampilkan legenda.

3.1.7 Histogram

Histogram digunakan untuk melihat distribusi dari variabel numerik/kontinu, seperti `Umur`.

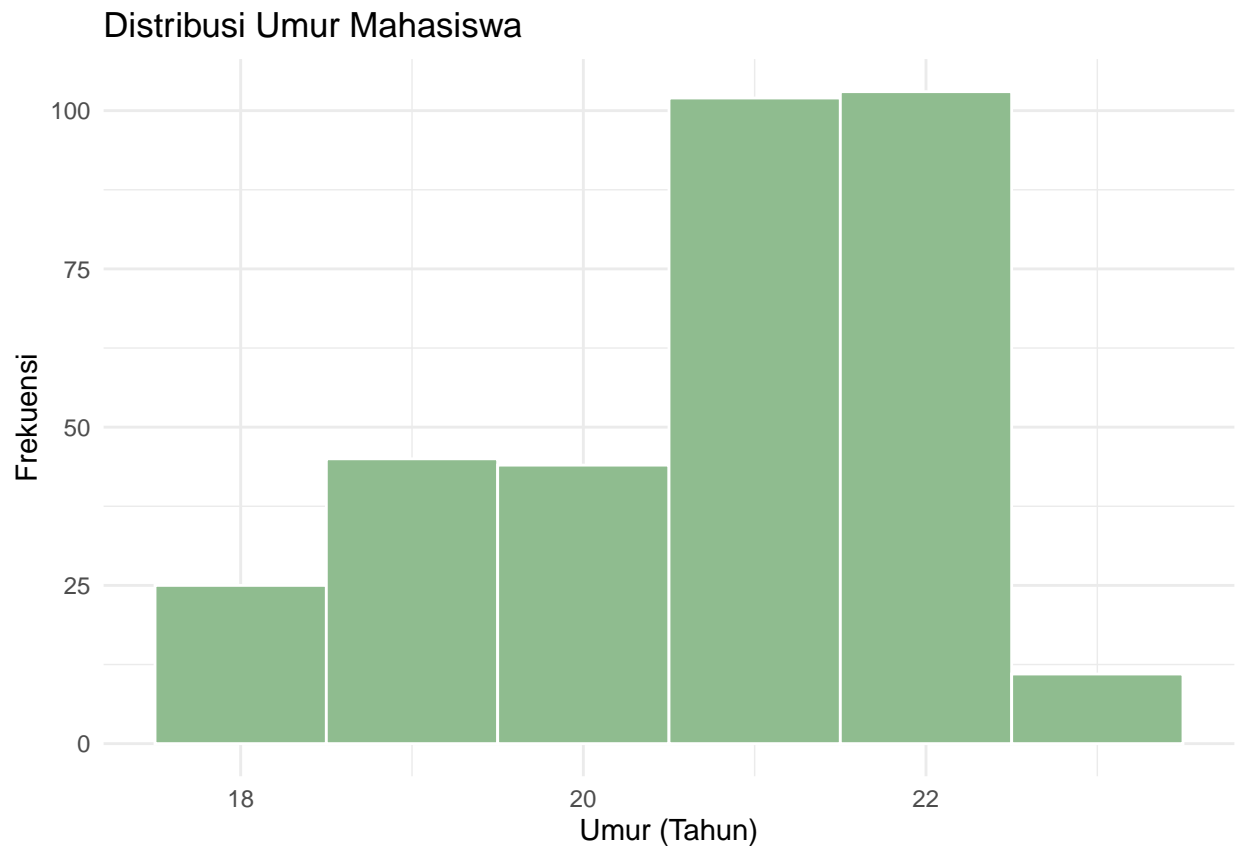
```
histogram <- ggplot(data.ubl.cleaned) +  
  geom_histogram(mapping = aes(x = Umur), binwidth = 1, fill = "darkseagreen", color = "white") +  
  labs(  
    title = "Distribusi Umur Mahasiswa",
```

```

x = "Umur (Tahun)",
y = "Frekuensi"
) +
theme_minimal()

```

histogram



Penjelasan Sintaks (Grammar of Graphics):

- **DATA:** `ggplot(data.ubl.cleaned)` mendefinisikan dataset.
- **GEOM:** `geom_histogram(...)` menentukan bentuk geometris berupa histogram.
- **MAPPING:** `mapping = aes(x = Umur)` memetakan variabel numerik `Umur` ke sumbu x.
- **STAT:** `geom_histogram` memiliki `stat = "bin"` sebagai defaultnya. Transformasi statistik ini akan membagi data `Umur` ke dalam beberapa rentang (bins) yang lebarnya diatur oleh `binwidth = 1`, lalu menghitung frekuensi data di setiap rentang tersebut untuk ditampilkan di sumbu y.

3.1.8 *Boxplot*

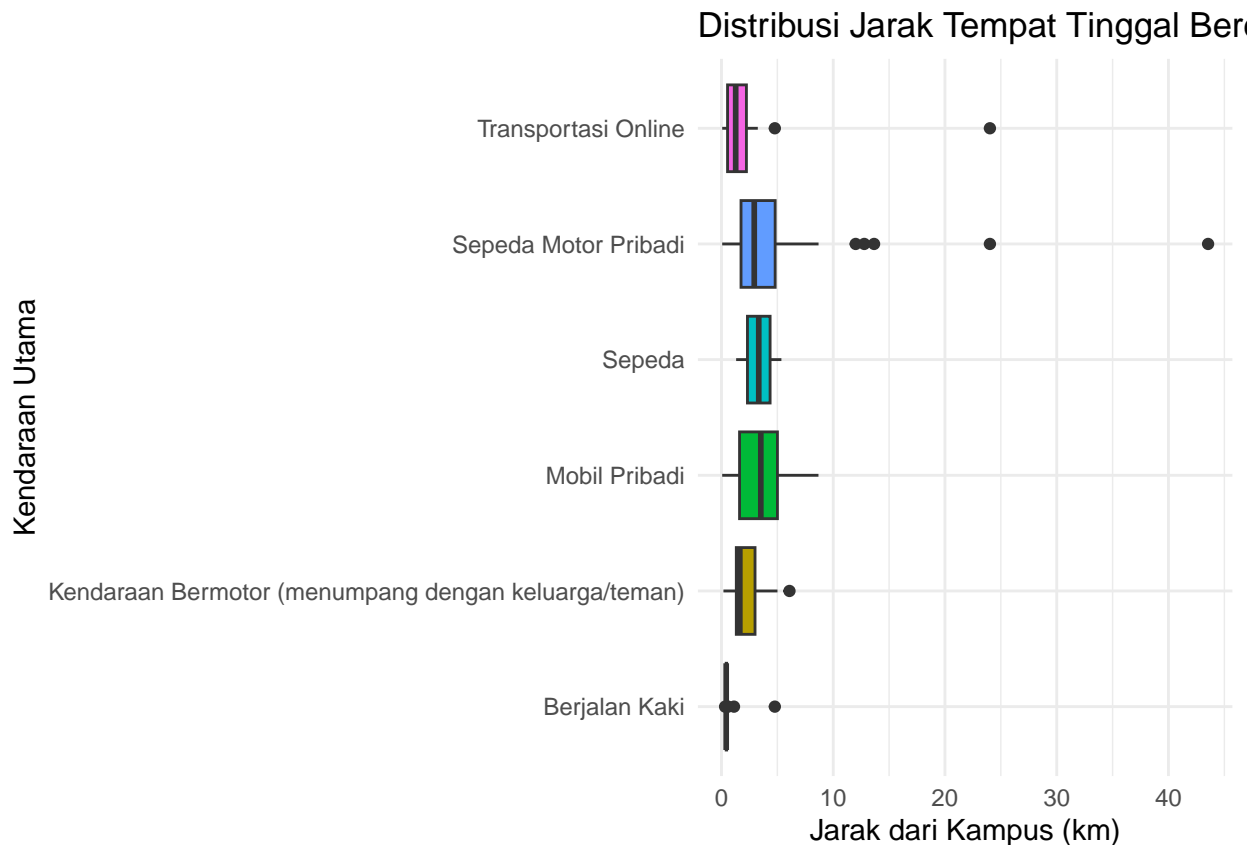
Box plot berguna untuk membandingkan distribusi variabel numerik di antara beberapa grup/kategori. Mari kita bandingkan distribusi jarak tempat tinggal (`jarak`) untuk setiap jenis kendaraan utama.

```

boxplot <- ggplot(data.ubl.cleaned) +
  geom_boxplot(mapping = aes(x = kendaraan.utama, y = jarak, fill = kendaraan.utama)) +
  coord_flip() + # Membalik sumbu agar label mudah dibaca

```

```
labs(
  title = "Distribusi Jarak Tempat Tinggal Berdasarkan Kendaraan Utama",
  x = "Kendaraan Utama",
  y = "Jarak dari Kampus (km)"
) +
theme_minimal() +
theme(legend.position = "none") # Menghilangkan legenda karena sudah ada di sumbu
boxplot
```



Interpretasi: Box plot ini menunjukkan bahwa mahasiswa yang menggunakan mobil pribadi cenderung memiliki rentang jarak tempat tinggal yang lebih bervariasi dan median yang sedikit lebih tinggi dibandingkan pengguna sepeda motor. Pengguna ojek online memiliki median jarak yang paling rendah di antara kategori lainnya.

Penjelasan Sintaks (Grammar of Graphics):

- **DATA:** `ggplot(data.ubl.cleaned)` mendefinisikan dataset.
- **GEOM:** `geom_boxplot(...)` menentukan bentuk geometris berupa diagram kotak.
- **MAPPING:** `mapping = aes(x = kendaraan.utama, y = \jarak.(km)', fill = kendaraan.utama)` memetakan tiga hal: variabel kategorikendaraan.utamake sumbu x, variabel numerikjarak.(km)ke sumbu y, dankendaraan.utamake warna isianfill'.
- **STAT:** `geom_boxplot` secara default menggunakan `stat_boxplot`, yang menghitung ringkasan lima angka (minimum, Q1, median, Q3, maksimum) untuk setiap grup di sumbu x.

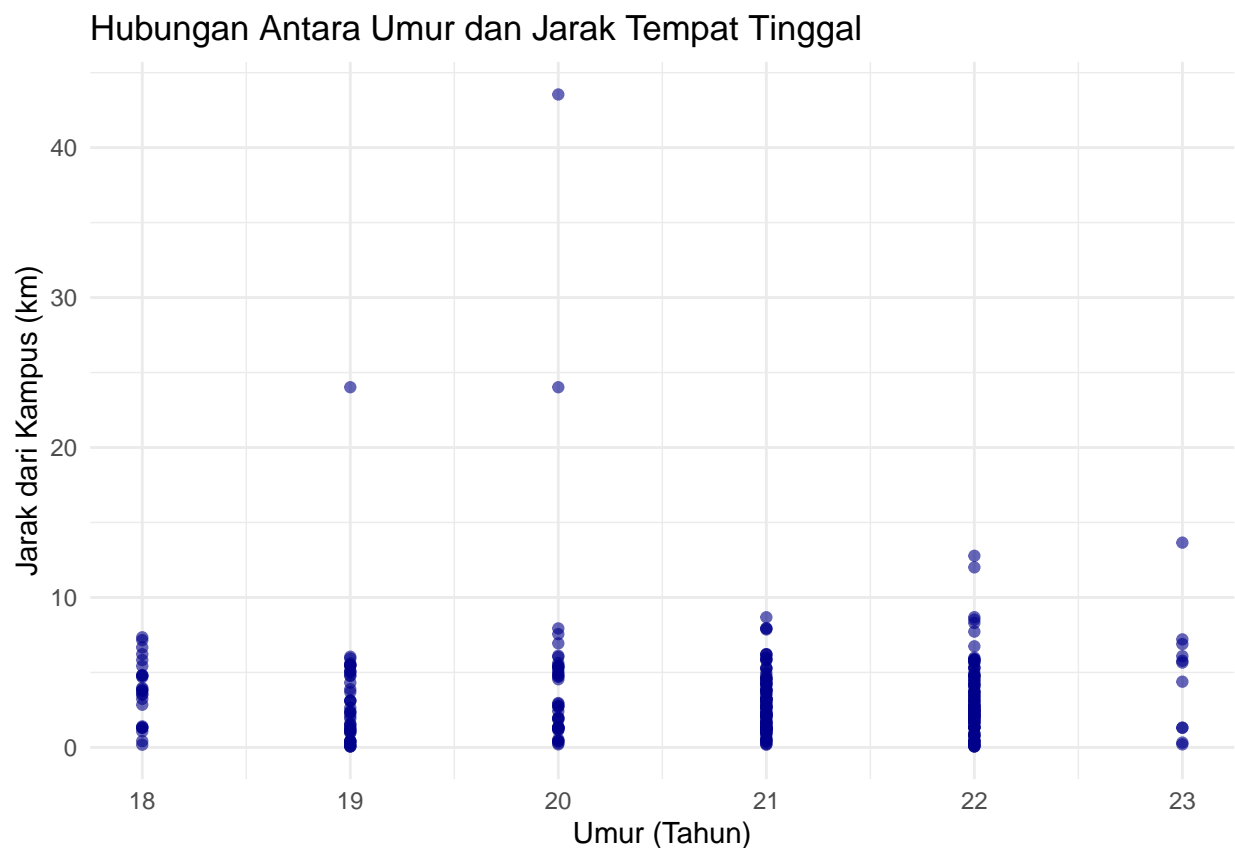
- **KOORDINAT:** `coord_flip()` secara eksplisit mengubah sistem koordinat dengan membalik sumbu x dan y. Ini adalah komponen terpisah yang diterapkan setelah komponen lainnya.
- `theme(legend.position = "none")`: Kustomisasi lapisan tema untuk menyembunyikan legenda.

3.1.9 Grafik Pencar (*Scatter Plot*)

Scatter plot ideal untuk melihat hubungan antara dua variabel numerik. Mari kita lihat hubungan antara Umur dan jarak tempat tinggal.

```
scatter_plot <- ggplot(data.ubl.cleaned) +
  geom_point(mapping = aes(x = Umur, y = jarak), alpha = 0.6, color = "darkblue") + # alpha untuk trans
  labs(
    title = "Hubungan Antara Umur dan Jarak Tempat Tinggal",
    x = "Umur (Tahun)",
    y = "Jarak dari Kampus (km)"
  ) +
  theme_minimal()

scatter_plot
```



Interpretasi: Grafik ini tidak menunjukkan adanya pola atau hubungan yang jelas antara umur mahasiswa dan jarak tempat tinggal mereka dari kampus. Titik-titik tersebar secara acak.

Penjelasan Sintaks (Grammar of Graphics):

- **DATA:** `ggplot(data.ubl.cleaned)` mendefinisikan dataset.

- **GEOM:** `geom_point(...)` menentukan bentuk geometris berupa titik.
- **MAPPING:** `mapping = aes(x = Umur, y = \jarak.(km))` memetakan dua variabel, Umur ke sumbu x dan `jarak.(km)` ke sumbu y. Setiap baris data akan menjadi satu titik pada plot.
- `alpha = 0.6` dan `color = "darkblue"` adalah pengaturan properti visual untuk semua titik.

3.2 Soal Praktikum

1. Kumpulkan kembali fail modul ini yang sudah lengkap dengan hasil pengerjaannya. **STP-3.3**
2. Tuliskan interpretasi kalian terhadap setiap diagram yang dihasilkan dalam modul ini. **STP-3.2**
3. Buatlah diagram yang tepat untuk variabel `biaya.dalam.sepekan`. **STP-3.1**
4. Mengapa Anda memilih diagram seperti yang Anda lakukan di nomor 3? Perhatikan tingkat pengukuran variabel dan jenis nilai dari variabel tersebut! **STP-3.4**

4 Modul-4: Distribusi Sampling dan Interval Kepercayaan

Setelah mempelajari modul ini, Anda diharapkan dapat:

1. menghasilkan distribusi statistik sampel dan menghitung standard error-nya **STP-4.3**
2. menghasilkan interval kepercayaan dengan menggunakan perangkat lunak komputer **STP-5.2**

4.1 Pendahuluan

Dalam statistika inferensial, kita seringkali ingin mengetahui karakteristik dari sebuah populasi (misalnya, rata-rata uang saku seluruh mahasiswa di Indonesia). Namun, mengumpulkan data dari seluruh populasi seringkali tidak memungkinkan. Sebagai solusinya, kita mengambil sampel acak dari populasi tersebut dan menggunakan statistik dari sampel (seperti rata-rata sampel) untuk menduga parameter populasi.

Modul ini akan membahas dua konsep fundamental dalam statistika inferensial: **Distribusi Sampling** dan **Interval Kepercayaan** untuk rata-rata dan proporsi. Kita akan menggunakan studi kasus data jarak tempuh mahasiswa dari empat universitas di Bandar Lampung dan sekitarnya untuk memahami bagaimana konsep ini diterapkan dalam praktik.

4.2 Perangkat Lunak dan Pustaka (Libraries)

Pastikan Anda telah menginstal pustaka `tidyverse` yang akan sangat membantu dalam proses manipulasi dan visualisasi data.

```
library(tidyverse)
library(readr)
```

4.3 Memuat dan Mempersiapkan Data

Langkah pertama adalah memuat data survei mahasiswa dari empat universitas. Kita akan menggabungkan dan membersihkan data tersebut untuk analisis. Variabel yang akan menjadi fokus kita adalah **jarak tempuh (km)** dan **jenis tempat tinggal**.

```
# Membaca 4 file CSV
```

```
df_uinril <- read_csv2("datasets/DataUtama_mhsUINRIL.csv")
```

```
## i Using "','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
## Rows: 400 Columns: 25
```

```
## -- Column specification -----
```

```
## Delimiter: ";"
```

```
## chr (11): Kampus_PT, Jenis Kelamin, Fakultas, Prodi, Tingkat Semester, Uang Saku, kendaraan utama, k
```

```
## dbl (14): Nomor urut, Umur, kepemilikan mobil, kepemilikan motor, kepemilikan sepeda, jarak (km), b
```

```
##
```

```
## 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.
```

```
df_ubl <- read_csv2("datasets/DataUtama_mhsUBL.csv")
```

```
## i Using "','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
## Rows: 379 Columns: 26-- Column specification -----
```

```
## Delimiter: ";"
```

```
## chr (14): Timestamp, Kampus_PT, Jenis Kelamin, Fakultas, Prodi, Tingkat Semester, Uang Saku, kendaraan
```

```
## dbl (12): Nomor urut, Umur, kepemilikan mobil, kepemilikan motor, kepemilikan sepeda, jarak (km), Jum
```

```
## 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.
```

```
df_unila <- read_csv2("datasets/DataUtama_mhsUNILA.csv")
```

```
## i Using "','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
## Rows: 393 Columns: 25-- Column specification -----
```

```
## Delimiter: ";"
```

```
## chr (11): Timestamp, Kampus_PT, Jenis Kelamin, Fakultas, Prodi, Tingkat Semester, Uang Saku, endaraan
```

```
## dbl (13): Nomor urut, Umur, kepemilikan mobil, Jumlah motor, Jumlah Sepeda, Biaya perjalanan, Jumlah
```

```
## lgl (1): jara
```

```
## 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.
```

```
df_itera <- read_csv2("datasets/DataUtama_mhsITERA.csv")
```

```
## i Using "','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
## Rows: 428 Columns: 26-- Column specification -----
```

```
## Delimiter: ";"
```

```
## chr (11): Timestamp, Kampus_PT, Jenis Kelamin, Fakultas, Prodi, Uang Saku, kendaraan utama, kelurahan
```

```
## dbl (15): Nomor urut, Umur, Tingkat Semester, kepemilikan mobil, kepemilikan motor, kepemilikan sepe
```

```
## 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.
```

```
# Membersihkan dan menggabungkan data
```

```
data_mahasiswa <- bind_rows(
```

```
  df_uinril |>
```

```
  select(
```

```
    kampus = Kampus_PT,
```

```

    jarak_km = `jarak (km)`,
    jenis_tinggal = `jenis tempat tinggal`
  ),
df_ubl |>
  select(
    kampus = Kampus_PT,
    jarak_km = `jarak (km)`,
    jenis_tinggal = `jenis tempat tinggal`
  ),
# Perhatikan nama kolom 'jarak' dan 'jenis tempat tinggal' berbeda di file UNILA
df_unila |>
  select(
    kampus = Kampus_PT,
    jarak_km = jara,
    jenis_tinggal = `Jenis Tempat Tinggal`
  ),
df_itera |>
  select(
    kampus = Kampus_PT,
    jarak_km = `jarak (km)`,
    jenis_tinggal = `jenis tempat tinggal`
  )
) |>
# Mengubah tipe data 'jarak_km' menjadi numerik
mutate(jarak_km = as.numeric(jarak_km)) |>
# Menghapus baris dengan data yang kosong (NA)
drop_na(jarak_km, jenis_tinggal) |>
# Melakukan filter sederhana untuk data jarak yang lebih masuk akal
filter(jarak_km > 0 & jarak_km < 50) |>
# Membuat variabel baru yang disederhanakan untuk jenis tempat tinggal
mutate(
  tipe_tinggal_baku = case_when(
    # Menggunakan str_detect untuk mencari kata kunci "kos", "asrama", atau "rusunawa" (tidak case-sensitive)
    str_detect(tolower(jenis_tinggal), "kos|asrama|rusunawa") ~ "Kos/Asrama",
    # Selain itu, dikategorikan sebagai "Rumah Keluarga/Pribadi"
    TRUE ~ "Rumah Keluarga/Pribadi"
  )
)

# Menampilkan beberapa baris pertama dari data gabungan
head(data_mahasiswa)

```

```

## # A tibble: 6 x 4
##   kampus jarak_km jenis_tinggal tipe_tinggal_baku
##   <chr>     <dbl> <chr>                <chr>
## 1 UINRIL    19.3  Rumah Bersama Saudara Rumah Keluarga/Pribadi
## 2 UINRIL     0.58 Kos Sendiri          Kos/Asrama
## 3 UINRIL     0.56 Kos Sendiri          Kos/Asrama
## 4 UINRIL     1.05 Kos Sendiri          Kos/Asrama
## 5 UINRIL     1.69 Rumah Mengontrak Bersama-sama Rumah Keluarga/Pribadi
## 6 UINRIL     7.91 Rumah Pribadi/Rumah Keluarga Rumah Keluarga/Pribadi

```

```
# Menampilkan ringkasan data
summary(data_mahasiswa)
```

```
##      kampus      jarak_km      jenis_tinggal      tipe_tinggal_baku
## Length:1206      Min.      : 0.06352      Length:1206      Length:1206
## Class :character      1st Qu.: 1.65799      Class :character      Class :character
## Mode  :character      Median : 3.27882      Mode  :character      Mode  :character
##                               Mean  : 3.90177
##                               3rd Qu.: 5.28562
##                               Max.   :43.54519
```

```
# Menampilkan frekuensi jenis tempat tinggal yang sudah dibakukan
table(data_mahasiswa$tipe_tinggal_baku)
```

```
##
##      Kos/Asrama Rumah Keluarga/Pribadi
##      504      702
```

Penjelasan case_when():

Fungsi `case_when()` (bagian dari `dplyr`) sangat berguna untuk membuat variabel baru berdasarkan serangkaian aturan kondisional, mirip seperti pernyataan IF-ELSE IF-ELSE. Di dalam `mutate()`, kita membuat kolom baru bernama `tipe_tinggal_baku`.

- **Aturan pertama:** `str_detect(tolower(jenis_tinggal), "kos|asrama|rusunawa") ~ "Kos/Asrama"`
 - `tolower(jenis_tinggal)`: Mengubah semua teks di kolom `jenis_tinggal` menjadi huruf kecil agar pencarian tidak sensitif terhadap huruf besar/kecil.
 - `str_detect(...)`: Fungsi ini memeriksa apakah sebuah teks mengandung pola tertentu.
 - `"kos|asrama|rusunawa"`: Ini adalah polanya. Tanda `|` berarti “ATAU”. Jadi, fungsi ini mencari kata “kos” ATAU “asrama” ATAU “rusunawa”.
 - `~ "Kos/Asrama"`: Jika salah satu kata kunci ditemukan, maka nilai untuk kolom `tipe_tinggal_baku` adalah “Kos/Asrama”.
- **Aturan kedua:** `TRUE ~ "Rumah Keluarga/Pribadi"`
 - `TRUE`: Ini adalah kondisi “penampung” atau *default*. Jika tidak ada kondisi sebelumnya yang terpenuhi, aturan ini akan dijalankan.
 - `~ "Rumah Keluarga/Pribadi"`: Memberi nilai “Rumah Keluarga/Pribadi” untuk semua baris yang tidak cocok dengan aturan pertama.

Singkatnya, kode ini membakukan data `jenis_tinggal` yang bervariasi menjadi dua kategori yang bersih dan konsisten untuk analisis proporsi.

Pada tahap ini, kita menganggap data gabungan dari 1.206 mahasiswa sebagai “populasi” kita untuk tujuan simulasi.

4.4 Distribusi *Sampling* dan *Standard Error*

Distribusi *sampling* adalah distribusi dari suatu statistik (misalnya, rata-rata) yang dihitung dari semua kemungkinan sampel dengan ukuran yang sama yang diambil dari sebuah populasi. Ini adalah istilah lain dari **distribusi statistik sampel**.

Teorema Limit Pusat (*Central Limit Theorem*) menyatakan bahwa jika kita mengambil sampel yang cukup besar, distribusi *sampling* dari rata-rata akan mendekati distribusi normal.

4.4.1 Mendefinisikan Parameter Populasi

Pertama, mari kita hitung rata-rata dan standar deviasi “populasi” kita sebagai acuan.

```
# Menghitung rata-rata jarak "populasi"
pop_mean <- mean(data_mahasiswa$jarak_km)
# Menghitung standar deviasi jarak "populasi"
pop_sd <- sd(data_mahasiswa$jarak_km)

paste("Rata-rata Jarak Populasi ( ):", round(pop_mean, 2), "km")
```

```
## [1] "Rata-rata Jarak Populasi ( ): 3.9 km"
```

```
paste("Standar Deviasi Populasi ( ):", round(pop_sd, 2), "km")
```

```
## [1] "Standar Deviasi Populasi ( ): 3.53 km"
```

4.4.2 Simulasi Pengambilan Sampel

Sekarang, kita akan mensimulasikan proses pengambilan sampel secara berulang. Kita akan mengambil 1000 sampel acak, masing-masing berukuran 50 mahasiswa ($n=50$), lalu menghitung rata-rata jarak untuk setiap sampel.

```
# Menetapkan parameter simulasi
ukuran_sampel <- 50
jumlah_simulasi <- 1000

# Menjalankan simulasi
set.seed(123) # Untuk hasil yang dapat direproduksi. Silakan ganti seed sesuai keinginan Anda.
# 'replicate' akan menjalankan ekspresi kedua sebanyak 'jumlah_simulasi' kali
rataan_sampel <- replicate(jumlah_simulasi, {
  sampel_jarak <- sample(data_mahasiswa$jarak_km, ukuran_sampel)
  mean(sampel_jarak)
})

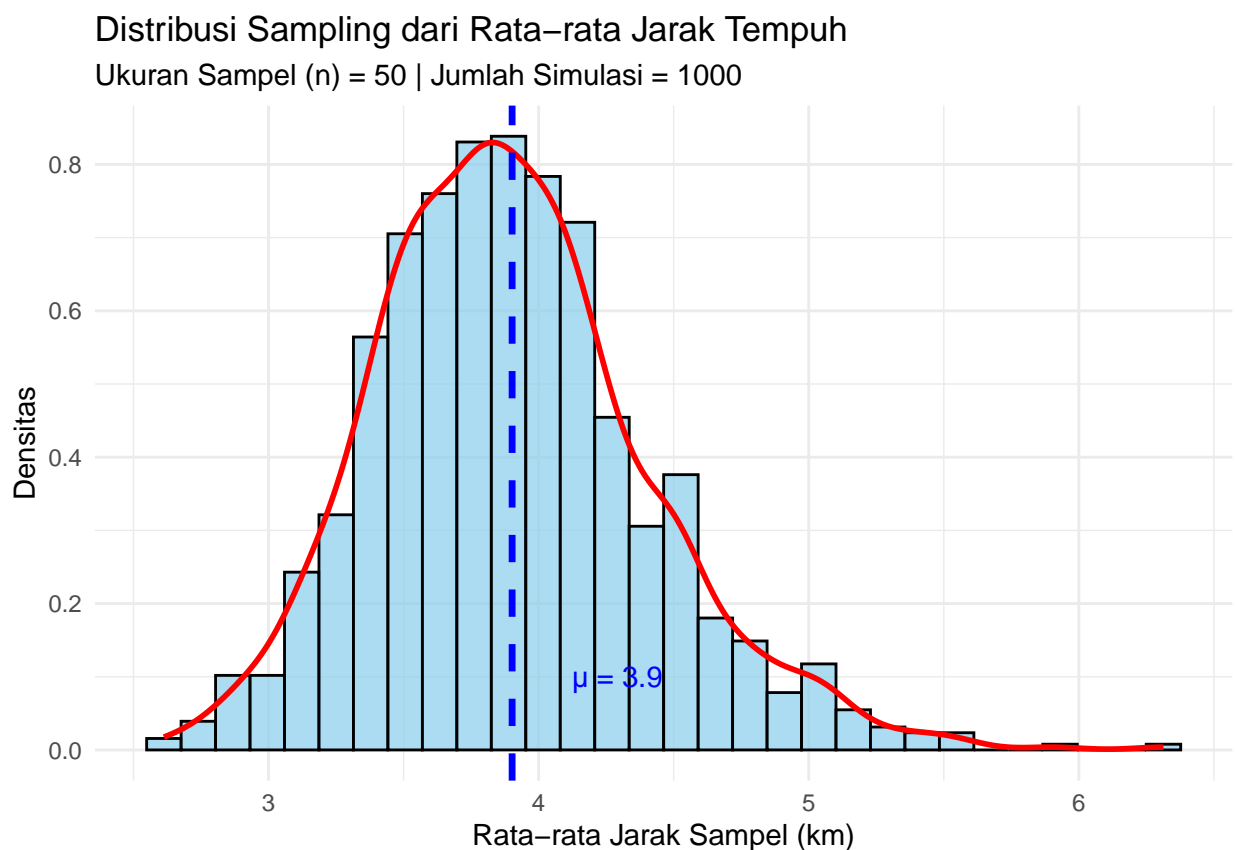
# Membuat dataframe dari hasil simulasi
df_sampling <- data.frame(rataan_sampel)
head(df_sampling)
```

```
##   rataan_sampel
## 1      3.619188
## 2      3.934443
## 3      4.089476
## 4      4.732993
## 5      3.701094
## 6      3.178470
```

4.4.3 Visualisasi Distribusi Sampling

Kumpulan dari 1000 rata-rata sampel inilah yang membentuk **distribusi sampling**. Mari kita visualisasikan dalam bentuk histogram.

```
# Visualisasi Distribusi Sampling dengan Histogram
ggplot(df_sampling, aes(x = rata_sampel)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "skyblue", color = "black", alpha = 0.7) +
  geom_density(color = "red", size = 1) +
  geom_vline(xintercept = pop_mean, color = "blue", linetype = "dashed", size = 1.2) +
  labs(
    title = "Distribusi Sampling dari Rata-rata Jarak Tempuh",
    subtitle = paste("Ukuran Sampel (n) =", ukuran_sampel, "| Jumlah Simulasi =", jumlah_simulasi),
    x = "Rata-rata Jarak Sampel (km)",
    y = "Densitas"
  ) +
  annotate("text", x = pop_mean * 1.1, y = 0.1, label = paste("μ =", round(pop_mean, 2)), color = "blue") +
  theme_minimal()
```



Perhatikan bagaimana distribusi dari rata-rata sampel berbentuk seperti lonceng (mendekati normal) dan berpusat di sekitar rata-rata populasi (garis biru putus-putus).

4.4.4 Menghitung *Standard Error* (SE)

Standard Error (SE) adalah standar deviasi dari distribusi sampling. Ini mengukur seberapa besar variasi rata-rata sampel di sekitar rata-rata populasi. SE dapat dipahami dari dua perspektif: satu dari sisi teori dalam simulasi, dan satu lagi dari sisi praktik ketika kita hanya memiliki satu sampel.

1. **Empiris (dari simulasi):** Menghitung standar deviasi dari 1000 rata-rata sampel yang kita hasilkan.

2. **Teoritis (dari populasi):** Menggunakan rumus $SE = \frac{\sigma}{\sqrt{n}}$, di mana σ adalah standar deviasi populasi dan n adalah ukuran sampel.
3. **Estimasi (dari sampel):** Dalam praktik statistika inferensial, kita hampir tidak pernah mengetahui σ . Oleh karena itu, kita mengestimasi menggunakan standar deviasi dari sampel kita sendiri (s). Rumusnya menjadi $SE = \frac{s}{\sqrt{n}}$. **Inilah nilai yang sebenarnya digunakan saat kita membuat interval kepercayaan atau melakukan uji hipotesis dari data sampel nyata.**

```
# 1. Standard Error Empiris (dari 1000 sampel)
se_empiris <- sd(df_sampling$rataan_sampel)

# 2. Standard Error Teoritis (menggunakan info populasi)
se_teoritis <- pop_sd / sqrt(ukuran_sampel)

paste("Standard Error (Empiris, dari simulasi):", round(se_empiris, 3))

## [1] "Standard Error (Empiris, dari simulasi): 0.505"

paste("Standard Error (Teoritis, dari populasi):", round(se_teoritis, 3))

## [1] "Standard Error (Teoritis, dari populasi): 0.5"
```

Mari kita hitung estimasi standard error dari satu sampel acak yang akan kita gunakan nanti di Bagian 4.5.

```
# Mengambil satu sampel acak (sama seperti di Bagian 4.1)
set.seed(42)
sampel_tunggal_untuk_se <- sample(data_mahasiswa$jarak_km, ukuran_sampel)
sd_sampel_tunggal <- sd(sampel_tunggal_untuk_se)

# 3. Estimasi Standard Error (dari satu sampel, kasus nyata)
se_estimasi <- sd_sampel_tunggal / sqrt(ukuran_sampel)
paste("Estimasi Standard Error (dari satu sampel):", round(se_estimasi, 3))

## [1] "Estimasi Standard Error (dari satu sampel): 0.283"
```

Perhatikan bahwa nilai SE Empiris dan Teoritis sangat dekat karena berasal dari simulasi di mana parameter populasi diketahui. Nilai SE Estimasi akan bervariasi tergantung sampel mana yang kita ambil, namun nilai inilah yang paling realistis untuk digunakan dalam analisis nyata.

4.5 Estimasi Parameter-1: Interval Kepercayaan Rata-rata

Untuk bagian ini kita akan menggunakan *library* bernama *MKinfer* yang berguna untuk menghasilkan interval kepercayaan sebagai estimasi rentang.

Lakukan instalasi *library* ini dengan perintah berikut.

```
install.packages("MKinfer")

## Warning: package 'MKinfer' is in use and will not be installed
```

Kemudian muat paket *MKinfer* tersebut.


```
library(MKinfer)
```

4.5.1 Mengambil Satu Sampel

```
# Kita gunakan sampel yang sudah dibuat sebelumnya
sampel_tunggal <- sampel_tunggal_untuk_se
mean_sampel_tunggal <- mean(sampel_tunggal)

cat(paste("Rata-rata Sampel Tunggal:", round(mean_sampel_tunggal, 2), "km\n"))
```

```
## Rata-rata Sampel Tunggal: 3.38 km
```

```
cat(paste("Standar Deviasi Sampel Tunggal:", round(sd_sampel_tunggal, 2), "km\n"))
```

```
## Standar Deviasi Sampel Tunggal: 2 km
```

4.5.2 Menghasilkan Interval Kepercayaan

```
# Menghitung interval kepercayaan 95% menggunakan MKinfer
hasil_ci_mean <- meanCI(sampel_tunggal, conf.level = 0.95)

# Menampilkan hasil
hasil_ci_mean
```

```
##
## Exact confidence interval(s)
##
## 95 percent confidence interval:
##      2.5 %    97.5 %
## mean 2.809335 3.946775
##
## sample estimates:
##      mean      sd
## 3.378055 2.001147
##
## additional information:
## SE of mean
## 0.2830049
```

Hasil dari fungsi `meanCI` tersebut langsung menunjukkan rentang kepercayaan yang limit bawahnya ditunjukkan oleh angka di bawah 2.5% dan limit atasnya oleh angka di bawah 97.5%. Nilai-nilai ini adalah nilai $\alpha/2$ yang dibagi dua ke kiri dan kanan grafik.

Pertanyaan

Berapakah rentang kepercayaan untuk rata-rata jarak tempat tinggal dari kampus dari sampel kita? Tuliskan interpretasinya yang tepat. Bandingkan dengan rata-rata parameter

```

# Jawablah pertanyaan di atas dengan menuliskannya sebagai komentar di chunk ini
# Rentang kepercayaan adalah 2,81 - 3,95 km

# Apa interpretasi dari hasil ini?
# Rata-rata jarak dari kampus untuk tempat tinggal seluruh mahasiswa berada di rentang 2,81 - 3,95 km.
pop_mean

## [1] 3.90177

```

4.6 Estimasi Parameter-2: Interval Kepercayaan Proporsi

Selain rata-rata, kita juga sering tertarik untuk mengestimasi proporsi dari suatu kategori dalam populasi. Contohnya, berapa persentase mahasiswa yang tinggal di kos/asrama? Prosesnya mirip: kita mengambil sampel, menghitung proporsi sampel, lalu membangun interval kepercayaan di sekitar proporsi tersebut.

4.6.1 Parameter Proporsi

Pertama, mari kita hitung proporsi “populasi” yang sebenarnya dari data kita. Kita akan mencari proporsi mahasiswa yang tinggal di “Kos/Asrama”.

```

# Menghitung proporsi populasi
tabel_populasi <- table(data_mahasiswa$tipe_tinggal_baku)
pop_prop <- prop.table(tabel_populasi)["Kos/Asrama"]

cat("Tabel Frekuensi Jenis Tinggal di Populasi:\n")

## Tabel Frekuensi Jenis Tinggal di Populasi:

print(tabel_populasi)

##
##           Kos/Asrama Rumah Keluarga/Pribadi
##           504                      702

cat(paste("\nProporsi Mahasiswa di Kos/Asrama (p):", round(pop_prop, 4)))

##
## Proporsi Mahasiswa di Kos/Asrama (p): 0.4179

```

4.6.2 Mengambil Satu Sampel dan Menghitung Proporsinya

Sekarang, kita ambil satu sampel acak (misalnya, $n=100$) dan hitung proporsi sampel \hat{p} mahasiswa yang tinggal di kos/asrama.

```

# Menetapkan ukuran sampel
ukuran_sampel_prop <- 100

# Mengambil sampel acak dari kolom tipe tinggal

```

```

set.seed(101) # Menggunakan seed baru
sampel_prop <- sample(data_mahasiswa$tipe_tinggal_baku, ukuran_sampel_prop)

# Menghitung frekuensi di dalam sampel
tabel_sampel <- table(sampel_prop)
# Menghitung proporsi sampel
prop_sampel <- prop.table(tabel_sampel)["Kos/Asrama"]

cat("Tabel Frekuensi Jenis Tinggal di Sampel:\n")

```

```
## Tabel Frekuensi Jenis Tinggal di Sampel:
```

```
print(tabel_sampel)
```

```
## sampel_prop
##           Kos/Asrama Rumah Keluarga/Pribadi
##                46                54
```

```
cat(paste("\nProporsi Sampel Mahasiswa di Kos/Asrama (p-hat):", round(prop_sampel, 4)))
```

```
##
## Proporsi Sampel Mahasiswa di Kos/Asrama (p-hat): 0.46
```

4.6.3 Menghasilkan Interval Kepercayaan untuk Proporsi

Kita akan menggunakan fungsi `binomCI()` di R. Fungsi ini memerlukan jumlah “sukses”, yakni jumlah mahasiswa di kos/asrama, dan total ukuran sampel.

```

# Mendapatkan jumlah "sukses" dari tabel sampel
jumlah_sukses <- tabel_sampel["Kos/Asrama"]
total_sampel <- ukuran_sampel_prop

# Menghitung interval kepercayaan 95% untuk proporsi
hasil_prop_test <- binomCI(jumlah_sukses, total_sampel, conf.level = 0.95)
# Menampilkan hasil
print(hasil_prop_test)

```

```
##
## wilson confidence interval
##
## 95 percent confidence interval:
##      2.5 %      97.5 %
## prob 0.3656081 0.5573514
##
## sample estimate:
## prob
## 0.46
##
## additional information:
## standard error of prob
##      0.04891501
```

Pertanyaan

Berapakah rentang kepercayaan untuk proporsi mahasiswa yang tinggal di kos/asrama? Tuliskan interpretasinya yang tepat. Bandingkan dengan proporsi parameter.

```
# Jawablah pertanyaan di atas dengan menuliskannya sebagai komentar di chunk ini # Rentang kepercayaan
# Tuliskan interpretasi dari rentang tersebut
# Tidak banyak mahasiswa yang tinggal di kos/asrama, karena rentang kepercayaan parameter proporsi maha.
```

4.7 Soal Latihan

Hasilkan rentang kepercayaan untuk rata-rata jarak tempuh dan proporsi penghuni kos/asrama dengan tingkat kepercayaan **97%**.

1. Berapakah rentang kepercayaan untuk rata-rata jarak?
2. Berapakah rentang kepercayaan untuk proporsi penghuni kos/asrama?

5 Modul-5: Uji Hipotesis

Setelah mempelajari modul ini, Anda diharapkan dapat:

1. mampu menghasilkan uji hipotesis satu populasi dengan menggunakan perangkat lunak komputer **STP-6.2**
2. mampu menghasilkan uji hipotesis dua atau lebih populasi pada suatu kasus menggunakan perangkat lunak komputer **STP-7.2**

5.1 Pendahuluan

Uji hipotesis adalah salah satu teknik dalam analisis statistik inferensial yang memperkirakan parameter melalui pernyataan-pernyataan dugaan atau **hipotesis**. Hipotesis di sini adalah pernyataan yang mengandung dugaan bahwa nilai parameter adalah sama dengan suatu nilai atau berbeda. Hipotesis yang sama dengan suatu nilai kita sebut dengan **hipotesis kosong** (*null hypothesis*), sementara yang berbeda disebut **hipotesis alternatif** (*alternative hypothesis*)

5.2 Perangkat Lunak dan Pustaka (Libraries)

Seperti biasa, kita perlu memuat pustaka (*libraries*) yang diperlukan dalam pengolahan data kita.

Dalam modul ini kita akan menggunakan sebuah pustaka bernama **stats** yang merupakan pustaka khusus untuk perhitungan-perhitungan statistik dan juga penghasil angka acak.

```
library(tidyverse)
library(readr)
library(stats)
```

Biasanya pustaka **stats** sudah termuat secara bawaan (*default*) saat kita menjalankan R melalui RStudio. Untuk mengecek apakah suatu pustaka sudah termuat ketika kita menjalankan R, tulis perintah berikut.

```
"package:{nama pustaka}" %in% search() # ganti {nama pustaka} dengan nama pustaka yang ingin dicari
"package: stats" %in% search() # untuk mengecek apakah 'stats' sudah dimuat
```

5.3 Memuat Data

```
data_mahasiswa <- read_csv2("data_mahasiswa.csv")
# read_csv2() → untuk file CSV dgn pemisah titik koma (;)
# read_csv() → untuk file CSV dgn pemisah koma (,).
```

```
data_mahasiswa <- read_csv2("datasets/data_mahasiswa.csv")
```

```
## i Using "','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
## Rows: 1206 Columns: 4
## -- Column specification -----
## Delimiter: ";"
## chr (3): kampus, jenis_tinggal, tipe_tinggal_baku
## dbl (1): jarak_km
##
## 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.
```

5.4 Perkenalan Markdown: LaTeX

Dalam file Markdown terdapat fitur bahasa LaTeX. Bahasa ini digunakan untuk menulis simbol matematis.

Untuk menyisipkan persamaan matematis menggunakan LaTeX, kita dapat menggunakan perintah Insert > LaTeX Math > Inline Math/Display Math. Inline Math akan menyisipkan simbol matematis di dalam paragraf, sementara Display Math membuat persamaan di bagian terpisah dari paragraf.

Dalam mode Source, Inline Math disisipkan dengan mengetikkan $$$$ terlebih dahulu dan menuliskan simbolnya di antara dua $$$ tersebut. Sementara itu, untuk Display Math kita perlu mengetikkan empat buah tanda $$$ sehingga terlihat seperti berikut.

```
$$
H_0 : \mu_0 = 4
\newline H_1 : \mu_0 \neq 4
$$
```

Hasilnya adalah seperti berikut:

$$H_0 : \mu_0 = 4 \quad H_1 : \mu_0 \neq 4$$

Anda dapat mempelajari penulisan LaTeX selengkapnya di sini.

5.5 Uji Hipotesis Satu Populasi

Pengujian hipotesis satu populasi bermakna pengujian hipotesis untuk suatu parameter yang berasal dari satu buah populasi saja. Misalnya, dalam kasus modul ini, kita memiliki populasi berupa seluruh mahasiswa yang berkuliah di Kota Bandar Lampung dan sekitarnya.

Dalam uji hipotesis satu populasi, kita hanya berfokus pada satu kelompok data, dan ingin mengetahui apakah parameter populasi sama dengan nilai tertentu atau tidak. Nilai acuan ini biasanya berasal dari teori, standar, atau dugaan awal. Ada dua jenis parameter yang bisa diuji:

Rata-rata (μ) → misalnya, apakah rata-rata jarak mahasiswa ke kampus = 4 km?

Proporsi (p) → misalnya, apakah proporsi mahasiswa yang tinggal di kos/asrama = 50%?

Prinsipnya sama: kita rumuskan hipotesis nol (H_0), tentukan hipotesis alternatif (H_1), hitung statistik uji (Z atau t), tentukan daerah kritis, lalu disimpulkan.

Penjelasan lebih lanjut akan dibahas dalam subbab-subbab berikut.

5.5.1 Rata-rata

Dalam kasus ini, kita akan menguji hipotesis parameter rata-rata berupa jarak mahasiswa yang berkuliah di universitas-universitas di Kota Bandar Lampung dan sekitarnya. Berdasarkan `data_mahasiswa`, kita dapat menghitung rata-rata statistik jarak mahasiswa yang menjadi responden. Rata-rata statistik jarak tersebut kita gunakan untuk menguji hipotesis parameter rata-rata dengan suatu nilai.

Jika dimisalkan suatu nilai tersebut adalah 4 km, maka hipotesis yang diujinya menjadi “rata-rata jarak tempat tinggal mahasiswa ke kampusnya masing-masing di Kota Bandar Lampung dan sekitarnya adalah 4 km.” Ini disebut **hipotesis kosong**, karena memuat kesamaan terhadap suatu nilai. Sementara itu, lawannya, yakni **hipotesis alternatif**, akan dibahas di masing-masing bagian.

Membuat fungsi perhitungan statistik uji

Pertama, mari kita buat sebuah fungsi di R untuk menghitung nilai statistik uji. Dengan adanya fungsi ini, kita tidak perlu lagi melakukan perhitungan manual setiap kali menguji hipotesis. Fungsi ini akan menerima masukan berupa rata-rata sampel, rata-rata hipotesis, simpangan baku, dan ukuran sampel, kemudian menghasilkan nilai statistik uji Z .

```
# Membuat fungsi uji hipotesis (hypothesis testing, ht) untuk rata-rata (mean)
# dengan 1 populasi (1pop)
# Keterangan input:
#   - xbar : statistik rata-rata
#   - mu   : parameter rata-rata
#   - sd   : statistik simpangan baku
#   - n    : ukuran sampel
ht_mean_1pop <- function(xbar, mu, sd, n) {
  se <- sd/sqrt(n) # menghitung standard error untuk rata-rata
  Z_hitung <- (xbar - mu)/se # menghitung nilai Z dari statistik
  return(Z_hitung)
}
```

Mendeklarasikan variabel uji

Setelah fungsi dibuat, kita deklarasikan variabel-variabel yang akan digunakan. Variabel-variabel ini akan dipakai langsung sebagai input fungsi uji hipotesis. Dengan mendeklarasikannya lebih dulu, kita memastikan semua nilai yang diperlukan untuk perhitungan sudah tersedia dengan jelas dan proses penghitungan menjadi lebih rapi.

```

# Mendeklarasikan variabel-variabel yang akan diuji
sample_mean <- mean(data_mahasiswa$jarak_km) # artinya menghitung rata-rata dari kolom jarak_km pada da
pop_mean <- 4.0 # ini nilai hipotesis yang ingin diuji.
sample_sd <- sd(data_mahasiswa$jarak_km)
# Menghasilkan data frame ringkasan dataset dengan menghitung jumlah mahasiswa sebagai ukuran sampel
ukuran_sampel <- data_mahasiswa |> # "/" disebut pipe operator, fungsinya untuk meneruskan hasil dari
  summarize(ukuran_sampel = n())
# Menyimpan angka jumlah mahasiswa sebagai variabel n
ukuran_sampel <- ukuran_sampel$ukuran_sampel # mengekstrak angka dari tabel menjadi nilai tunggal (scal

# Menampilkan variabel-variabel yang akan diuji
cat("Rata-rata jarak sampel adalah", sample_mean, "km.",
    "\nParameter rata-rata jarak yang diuji adalah", pop_mean, "km.",
    "\nSimpangan baku jarak sampel adalah", sample_sd, "km.",
    "\nUkuran sampel adalah", ukuran_sampel, "orang.")

```

```

## Rata-rata jarak sampel adalah 3.90177 km.
## Parameter rata-rata jarak yang diuji adalah 4 km.
## Simpangan baku jarak sampel adalah 3.534776 km.
## Ukuran sampel adalah 1206 orang.

```

5.5.1.1 Two-tailed Test (Tidak Berarah) Uji dua arah (two-tailed test) adalah salah satu bentuk pengujian hipotesis statistik yang digunakan ketika peneliti ingin mengetahui apakah nilai rata-rata populasi berbeda secara signifikan dari suatu nilai tertentu (nilai hipotesis, biasanya dilambangkan μ_0), tanpa memandang arah perbedaan tersebut. Artinya, kita tidak hanya mempertanyakan apakah rata-rata lebih besar atau lebih kecil, tetapi fokus pada apakah terdapat perbedaan yang cukup signifikan ke salah satu arah. Dalam uji ini, kita ingin mengetahui apakah rata-rata jarak tempuh mahasiswa menuju kampus berbeda secara signifikan dari 4 km, tanpa menentukan apakah lebih besar atau lebih kecil.

5.5.1.1.1 Merumuskan hipotesis Langkah pertama adalah merumuskan hipotesis. Di sini kita mendefinisikan dua kemungkinan: hipotesis nol (H_0) dan hipotesis alternatif (H_1). H_0 biasanya menyatakan bahwa “tidak ada perbedaan” atau “nilai parameter sama dengan nilai tertentu”. Sebaliknya, H_1 menyatakan adanya perbedaan yang signifikan. Pada uji ini, Hipotesis nol menyatakan bahwa rata-rata sama dengan 4 km, sedangkan hipotesis alternatif menyatakan bahwa rata-rata tidak sama dengan 4 km. Dengan demikian, kita sedang menguji apakah terdapat perbedaan nyata dari nilai acuan tersebut.

```

cat("H0 : rata-rata jarak = 4 km", # fungsi "cat()" digunakan untuk mencetak teks ke console dengan form
    "\n\nH1 : rata-rata jarak != 4 km") # "!=" berarti "tidak sama dengan" (menandakan uji dua sisi).

## H0 : rata-rata jarak = 4 km
##
## H1 : rata-rata jarak != 4 km

```

5.5.1.1.2 Memilih distribusi *sampling*, wilayah dan titik kritis Setelah hipotesis dirumuskan, kita perlu menentukan distribusi sampling dan titik kritis. Distribusi sampling adalah distribusi dari nilai rata-rata sampel jika kita mengambil banyak sampel dari populasi. Titik kritis adalah nilai ambang yang menentukan wilayah penolakan hipotesis nol. Dalam uji dua arah, daerah kritis terletak di kedua ujung distribusi normal dengan luas masing-masing sebesar $\alpha/2$. Misalnya, dengan $\alpha = 0,05$, maka masing-masing sisi memiliki 0,025. Artinya, hanya jika nilai uji berada jauh di salah satu sisi, kita akan menolak H_0 .

```
# Hipotesis yang dipakai tidak berarah, maka wilayah kritis adalah alpha/2
alpha <- 0.05
Z.crit_2tail <- abs(qnorm(alpha/2)) # Nilai yang digunakan adalah nilai absolut, karena dibagi 2, tanda
cat("Titik kritis uji two-tailed:", Z.crit_2tail)
```

```
## Titik kritis uji two-tailed: 1.959964
```

5.5.1.1.3 Menghitung statistik uji Setelah menentukan titik kritis, kita menghitung nilai statistik uji menggunakan data yang kita miliki. Nilai ini menunjukkan seberapa jauh rata-rata sampel menyimpang dari nilai hipotesis populasi, dalam satuan standar error. Jika nilai ini berada di luar batas kritis, maka penyimpangan tersebut dianggap terlalu besar untuk hanya terjadi karena kebetulan.

```
# hasil perhitungan statistik uji diabsolutkan untuk mengambil nilainya saja tanpa tandanya
Z_meanJarak <- abs(ht_mean_ipop(sample_mean, pop_mean, sample_sd, ukuran_sampel))
cat("Hasil statistik uji rata-rata:", Z_meanJarak)
```

```
## Hasil statistik uji rata-rata: 0.9650591
```

5.5.1.1.4 Menarik kesimpulan Langkah terakhir adalah menarik kesimpulan berdasarkan perbandingan antara nilai statistik uji dan titik kritis. Jika nilai statistik uji (absolut) lebih kecil dari titik kritis, maka kita gagal menolak H_0 , artinya perbedaan yang terlihat kemungkinan hanya kebetulan sampel. Jika lebih besar atau sama, kita menolak H_0 , artinya ada bukti yang cukup bahwa rata-rata jarak tidak sama dengan 4.

```
if (Z_meanJarak < Z.crit_2tail) {
  tolak_h0 <- FALSE
  cat("Hipotesis kosong gagal ditolak. Perbedaan rata-rata sampel", round(sample_mean, 2), "dengan", pop_mean)
} else {
  tolak_h0 <- TRUE
  cat("Hipotesis kosong ditolak. Rata-rata jarak pada sampel mahasiswa sebesar", round(sample_mean, 2), "km")
}
```

```
## Hipotesis kosong gagal ditolak. Perbedaan rata-rata sampel 3.9 dengan 4 hanyalah kebetulan
```

5.5.1.2 One-tailed Test (Berarah) Uji satu arah (one-tailed test) digunakan ketika kita memiliki dugaan arah perbedaan. Misalnya, kita ingin tahu apakah rata-rata sampel lebih kecil dari 4 km, bukan sekadar berbeda. Dalam konteks ini, daerah kritis hanya berada di salah satu sisi distribusi. Uji satu arah biasanya lebih sensitif karena semua probabilitas ditempatkan di satu sisi, sehingga lebih mudah untuk mendeteksi perbedaan yang sesuai dengan arah hipotesis alternatif.

5.5.1.2.1 Merumuskan Hipotesis Untuk uji satu arah, hipotesis alternatif ditentukan dengan arah tertentu. Misalnya, jika kita menduga rata-rata jarak lebih kecil dari 4 km, maka:

Dengan cara ini, kita hanya akan menolak H_0 jika bukti menunjukkan bahwa rata-rata memang lebih kecil. Jika ternyata rata-rata lebih besar, maka hal itu tidak mendukung H_1 .


```
cat("H0 : rata-rata jarak = 4 km",
    "\n\nH1 : rata-rata jarak < 4 km")
```

```
## H0 : rata-rata jarak = 4 km
##
## H1 : rata-rata jarak < 4 km
```

5.5.1.2.2 Memilih distribusi sampling, wilayah, serta titik kritis Karena hipotesis alternatif hanya mengarah ke satu sisi, maka daerah kritis hanya ditetapkan di sisi kiri atau kanan distribusi normal. Jika H_1 menyatakan rata-rata lebih kecil, maka titik kritis ditentukan pada sisi kiri. Sebaliknya, jika H_1 menyatakan rata-rata lebih besar, maka titik kritis ada di sisi kanan. Dengan $\alpha = 0,05$, seluruh probabilitas kesalahan ditempatkan di satu sisi distribusi.

```
tail <- "kiri" # H1 menyatakan lebih kecil dari (left-tailed)

if (tail == "kiri") {
  Z.crit_1tail <- qnorm(alpha)
} else {
  Z.crit_1tail <- qnorm(1-alpha)
}

cat("Titik kritis uji one-tailed:", Z.crit_1tail)
```

```
## Titik kritis uji one-tailed: -1.644854
```

5.5.1.2.3 Menghitung statistik uji Setelah titik kritis ditentukan, kita menghitung nilai statistik uji. Pada uji satu arah, tanda dari nilai Z menjadi sangat penting. Hal ini karena nilai negatif atau positif menunjukkan arah penyimpangan rata-rata sampel dari nilai populasi.

```
# Hasil perhitungan statistik uji TIDAK diabsolutkan
Z_meanJarak <- ht_mean_1pop(sample_mean, pop_mean, sample_sd, ukuran_sampel)

cat("Hasil statistik uji rata-rata:", Z_meanJarak)
```

```
## Hasil statistik uji rata-rata: -0.9650591
```

5.5.1.2.4 Menarik kesimpulan Langkah terakhir adalah membandingkan nilai statistik uji dengan titik kritis. Jika nilai uji lebih kecil dari titik kritis atau jatuh ke dalam daerah kritis, maka H_0 ditolak, artinya perbedaan rata-rata kemungkinan hanyalah kebetulan. Sebaliknya, jika nilai uji lebih besar, maka H_0 gagal ditolak yang berarti tidak ada bukti cukup untuk menyatakan adanya perbedaan.

```
if (Z_meanJarak < Z.crit_1tail) {
  tolak_h0 <- TRUE
  cat("Hipotesis kosong ditolak. Rata-rata jarak pada sampel mahasiswa sebesar", round(sample_mean, 2),
} else {
  tolak_h0 <- FALSE
  cat("Hipotesis kosong gagal ditolak. Perbedaan rata-rata sampel dengan populasi hanyalah kebetulan")
}
```

```
## Hipotesis kosong gagal ditolak. Perbedaan rata-rata sampel dengan populasi hanyalah kebetulan
```

5.5.2 Proporsi

Proporsi adalah perbandingan jumlah elemen yang memenuhi suatu kriteria terhadap jumlah total. Misalnya, dalam kasus mahasiswa, kita bisa melihat proporsi mahasiswa yang tinggal di kos/asrama dibandingkan dengan seluruh mahasiswa dalam sampel.

Pengujian proporsi bermanfaat ketika kita ingin menguji apakah proporsi sampel yang kita peroleh berbeda secara signifikan dari nilai proporsi populasi yang diasumsikan. Contoh kasusnya adalah: “Apakah benar 50% mahasiswa di Bandar Lampung tinggal di kos/asrama?” Jika hasil sampel kita menunjukkan 47% atau 55%, kita perlu uji hipotesis untuk menentukan apakah perbedaan tersebut kebetulan atau memang signifikan.

5.5.2.0.1 Membuat fungsi Untuk memudahkan proses, kita membuat sebuah fungsi di R yang dapat menghitung statistik uji Z untuk proporsi satu populasi. Fungsi ini akan menerima input berupa:

```
ht_prop_1pop <- function(p_hat, P, n) {  
  se <- sqrt((p_hat * (1-p_hat))/n) # menghitung standard error untuk rata-rata  
  Z_hitung <- (p_hat - P)/se # menghitung nilai Z dari statistik  
  return(Z_hitung)  
}
```

5.5.2.0.2 Meringkas Data Sebelum melakukan uji hipotesis, kita perlu mengetahui berapa proporsi mahasiswa yang tinggal di masing-masing tipe tempat tinggal. Kita bisa gunakan fungsi `group_by()` dan `summarize()` untuk menghitung jumlah mahasiswa berdasarkan tipe tempat tinggal, lalu menambahkan kolom proporsi. Dengan cara ini, kita mendapatkan ringkasan data yang memperlihatkan distribusi tempat tinggal mahasiswa.

```
summarize_tempatTinggal <- data_mahasiswa |>  
  group_by("Tipe tinggal" = tipe_tinggal_baku) |>  
  summarize("Jumlah" = n()) |>  
  mutate("proporsi" = Jumlah/sum(Jumlah))
```

```
summarize_tempatTinggal
```

```
## # A tibble: 2 x 3  
##   `Tipe tinggal`      Jumlah proporsi  
##   <chr>             <int>     <dbl>  
## 1 Kos/Asrama        504      0.418  
## 2 Rumah Keluarga/Pribadi 702      0.582
```

5.5.2.0.3 Menyimpan Nilai Proporsi Setelah kita memperoleh ringkasan data, kita simpan nilai proporsi mahasiswa yang tinggal di kos atau asrama ke dalam sebuah variabel. Dengan cara ini, nilai tersebut dapat digunakan dengan mudah dalam perhitungan selanjutnya tanpa harus menghitung ulang.

```
proporsi_kosAsrama <- summarize_tempatTinggal$proporsi[1]  
proporsi_kosAsrama
```

```
## [1] 0.4179104
```

5.5.2.0.4 Mendeklarasikan variabel pengujian Pada tahap ini, kita menentukan variabel-variabel utama yang akan digunakan dalam uji hipotesis. Semua variabel inilah yang nantinya akan dimasukkan ke dalam fungsi uji hipotesis yang telah dibuat.

```
# Mendeklarasikan variabel-variabel yang akan diuji
sample_prop <- proporsi_kosAsrama
pop_prop <- 0.5

# Menghasilkan data frame ringkasan dataset dengan menghitung jumlah mahasiswa sebagai ukuran sampel
ukuran_sampel <- data_mahasiswa |>
  summarize(ukuran_sampel = n())
# Menyimpan angka jumlah mahasiswa sebagai variabel n
ukuran_sampel <- ukuran_sampel$ukuran_sampel

# Menampilkan variabel-variabel yang akan diuji
cat("Proporsi penghuni Kos/Asrama sampel adalah", sample_prop,
    "\nParameter proporsi penghuni Kos/Asrama yang diuji adalah", pop_prop,
    "\nUkuran sampel adalah", ukuran_sampel, "orang.")

## Proporsi penghuni Kos/Asrama sampel adalah 0.4179104
## Parameter proporsi penghuni Kos/Asrama yang diuji adalah 0.5
## Ukuran sampel adalah 1206 orang.
```

5.5.2.1 Two-tailed Test Pada pengujian hipotesis proporsi dengan two-tailed test, kita ingin mengetahui apakah proporsi sampel berbeda secara signifikan dari proporsi populasi yang dihipotesiskan. Artinya, kita hanya fokus pada perbedaan, bukan pada arah perbedaan. Misalnya, kita ingin menguji apakah proporsi mahasiswa yang tinggal di kos/asrama benar-benar sama dengan 50% atau tidak. Jika hasilnya jauh lebih besar atau jauh lebih kecil, keduanya dianggap sebagai bukti menolak hipotesis nol.

5.5.2.1.1 Merumuskan hipotesis Sama seperti uji dua arah pada rata-rata, uji dua arah pada proporsi diawali dengan merumuskan hipotesis. Di sini kita mendefinisikan dua kemungkinan: hipotesis nol (H_0) dan hipotesis alternatif (H_1). H_0 biasanya menyatakan bahwa “tidak ada perbedaan” atau “nilai parameter sama dengan nilai tertentu”. Sebaliknya, H_1 menyatakan adanya perbedaan yang signifikan. Dalam uji dua arah, rumus hipotesis ditulis sebagai:

```
cat("H0 : prop. Kos/Asrama = 0,5",
    "\n\nH1 : prop. Kos/Asrama != 0,5")

## H0 : prop. Kos/Asrama = 0,5
##
## H1 : prop. Kos/Asrama != 0,5
```

5.5.2.1.2 Memilih distribusi sampling, wilayah dan titik kritis Sama seperti sebelumnya, setelah hipotesis ditetapkan, kita perlu menentukan distribusi yang digunakan. Untuk kasus proporsi dengan ukuran sampel besar, digunakan distribusi normal. Kemudian, karena kita menggunakan uji dua arah, wilayah kritis akan terbagi dua, masing-masing di ekor kiri dan kanan distribusi. Kali ini, kita menggunakan taraf signifikansi $= 0.10$.

```
# Hipotesis yang dipakai tidak berarah, maka wilayah kritis adalah alpha/2
alpha <- 0.10 # (tingkat signifikansi 10%)
Z.crit_2tail <- abs(qnorm(alpha/2)) # Nilai yang digunakan adalah nilai absolut
```

```
cat("Titik kritis uji two-tailed:", Z.crit_2tail)
```

```
## Titik kritis uji two-tailed: 1.644854
```

5.5.2.1.3 Menghitung Statistik Uji Setelah titik kritis ditentukan, kita menghitung nilai statistik uji Z berdasarkan proporsi sampel dan proporsi populasi yang diuji.

```
# Hasil perhitungan statistik uji diabsolutkan
Z_propKos <- abs(ht_prop_1pop(sample_prop, pop_prop, ukuran_sampel))

Z_propKos
```

```
## [1] 5.779963
```

5.5.2.1.4 Menarik kesimpulan Langkah terakhir adalah membandingkan nilai statistik uji dengan titik kritis. Dari sini kita bisa menarik kesimpulan apakah hipotesis nol ditolak atau gagal ditolak. Jika nilai statistik uji (absolut) lebih kecil dari titik kritis, maka H_0 gagal ditolak, artinya perbedaan yang terlihat kemungkinan hanya kebetulan sampel. Jika lebih besar atau sama, maka H_0 ditolak, artinya ada bukti yang cukup bahwa proporsi tidak sama dengan 0,5.

```
if (Z_propKos < Z.crit_2tail) {
  tolak_h0 <- FALSE
  cat("Hipotesis kosong gagal ditolak. Perbedaan proporsi penghuni Kos/Asrama sebanyak", round(sample_p
} else {
  tolak_h0 <- TRUE
  cat("Hipotesis kosong ditolak. Perbedaan proporsi penghuni Kos/Asrama sebanyak", round(sample_prop, 3
}
```

```
## Hipotesis kosong ditolak. Perbedaan proporsi penghuni Kos/Asrama sebanyak 0.418 dari 0.5 adalah sign
```

5.5.2.2 One-tailed Test Pada uji hipotesis proporsi satu arah, kita hanya berfokus dengan satu sisi distribusi, yakni apakah proporsi lebih besar atau lebih kecil dari nilai yang dihipotesiskan (kiri untuk dugaan "<", kanan untuk dugaan ">"). Uji ini lebih spesifik karena hipotesis alternatif sudah menetapkan arah. Misalnya, kita ingin tahu apakah proporsi mahasiswa kos/asrama lebih kecil dari 50%. Dalam uji ini, hanya sisi kiri distribusi yang menjadi perhatian.

5.5.2.2.1 Merumuskan hipotesis Hipotesis dirumuskan sebagai berikut:

```
cat("H0 : prop. Kos/Asrama = 0,5",
    "\n\nH1 : prop. Kos/Asrama < 0,5")
```

```
## H0 : prop. Kos/Asrama = 0,5
##
## H1 : prop. Kos/Asrama < 0,5
```

5.5.2.2 Memilih distribusi Sampling, wilayah dan titik kritis Distribusi sampling yang digunakan tetap distribusi normal. Namun, karena ini uji satu arah, maka seluruh ditempatkan pada salah satu sisi. Jika hipotesis alternatifnya menyatakan lebih besar, maka daerah kritis ada di sebelah kanan. Jika menyatakan lebih kecil, maka daerah kritis ada di sebelah kiri. Dengan taraf signifikansi α , titik kritis dihitung dengan $qnorm(1 - \alpha)$ untuk sisi kanan atau $qnorm(\alpha)$ untuk sisi kiri.

```
tail <- "kiri"

if (tail == "kiri") {
  Z.crit_1tail <- qnorm(alpha)
} else {
  Z.crit_1tail <- qnorm(1-alpha)
}

cat("Titik kritis uji one-tailed:", Z.crit_1tail)
```

```
## Titik kritis uji one-tailed: -1.281552
```

5.5.2.2.3 Menghitung statistik uji Sama seperti pada uji dua arah, kita menghitung nilai statistik uji Z dengan fungsi yang sudah dibuat sebelumnya tetapi kali ini kita tidak mengambil nilai absolut karena tanda statistik memberi informasi arah.

```
# Hasil perhitungan statistik uji TIDAK diabsolutkan
Z_propKos <- ht_prop_1pop(sample_prop, pop_prop, ukuran_sampel)

Z_propKos
```

```
## [1] -5.779963
```

5.5.2.2.4 Menarik kesimpulan Kesimpulan diambil dengan membandingkan nilai statistik uji ($Z_propKos$) dengan titik kritis ($Z.crit_1tail$). Untuk uji satu arah (kiri), kita menolak H_0 jika nilai statistik uji lebih kecil dari titik kritis. Jika tidak, kita gagal menolak H_0 . Kesimpulan ini menjelaskan apakah ada cukup bukti untuk menyatakan proporsi memang lebih kecil dari nilai acuan.

```
if (Z_propKos < Z.crit_1tail) {
  tolak_h0 <- TRUE
  cat("Hipotesis kosong ditolak. Perbedaan proporsi penghuni Kos/Asrama sebanyak", round(sample_prop, 3), "%")
} else {
  tolak_h0 <- FALSE
  cat("Hipotesis kosong gagal ditolak. Perbedaan proporsi penghuni Kos/Asrama sebanyak", round(sample_prop, 3), "%")
}
```

```
## Hipotesis kosong ditolak. Perbedaan proporsi penghuni Kos/Asrama sebanyak 0.418 dari 0.5 adalah signifikan
```

5.6 Uji Hipotesis 2 Populasi Independen

Dua populasi dikatakan independen apabila tidak ada pengaruh antara populasi 1 dengan populasi lain. Artinya, data dari populasi pertama dan kedua berdiri sendiri, tidak saling berkaitan. Contohnya adalah membandingkan rata-rata jarak tempat tinggal mahasiswa ITERA dengan mahasiswa UINRIL, di mana setiap kelompok mahasiswa berasal dari populasi yang berbeda.

5.6.1 Rata-rata

5.6.1.0.1 Merumuskan fungsi Pada langkah ini, kita akan membuat fungsi uji hipotesis rata-rata untuk dua populasi independen. Fungsi ini diperlukan karena kita ingin menghitung nilai statistik uji (Z) berdasarkan data ringkasan dari dua sampel. Dengan fungsi ini, kita bisa langsung memasukkan nilai-nilai statistik dari kedua populasi dan mendapatkan nilai Z yang akan dibandingkan dengan titik kritis.

```
ht_mean_2popIn <- function(xbar1, xbar2, sd1, sd2, n1, n2) {  
  se <- sqrt(  
    (  
      (sd1^2)/(n1 - 1)  
    ) + (  
      (sd2^2)/(n2 - 1)  
    )  
  )  
  Z_hitung <- (xbar1 - xbar2)/se # menghitung nilai Z dari statistik  
  return(Z_hitung)  
}
```

5.6.1.0.2 Menghitung mean 2 populasi Sebelum menjalankan uji hipotesis, kita perlu mendapatkan data ringkasan berupa rata-rata, simpangan baku, dan jumlah sampel dari masing-masing populasi. Pada tahap ini, kita akan membuat tabel ringkasan jarak mahasiswa dari rumah ke kampus berdasarkan kelompok kampus. Data ini nantinya akan menjadi input untuk fungsi yang sudah kita buat di langkah sebelumnya. Dengan cara ini, kita bisa lebih mudah membandingkan dua populasi secara langsung.

```
summarize_Jarak2Pop <- data_mahasiswa |>  
  group_by(Kampus = kampus) |>  
  summarize(`Rata-rata` = mean(jarak_km),  
            `Simpangan baku` = sd(jarak_km),  
            Jumlah = n())  
  
summarize_Jarak2Pop
```

```
## # A tibble: 3 x 4  
##   Kampus `Rata-rata` `Simpangan baku` Jumlah  
##   <chr>      <dbl>          <dbl> <int>  
## 1 ITERA      5.01            2.82   427  
## 2 UBL        3.38            3.74   379  
## 3 UINRIL     3.21            3.74   400
```

5.6.1.0.3 Mendeklarasikan variabel uji Setelah mendapatkan ringkasan data, kita harus menyimpan nilai rata-rata, simpangan baku, dan ukuran sampel ke dalam variabel khusus. Variabel-variabel ini akan dipakai langsung sebagai input fungsi uji hipotesis. Dengan mendeklarasikannya lebih dulu, proses penghitungan menjadi lebih rapi dan mudah diikuti.

```
mean_jarakUINRIL <- summarize_Jarak2Pop$`Rata-rata`[3]  
mean_jarakITERA <- summarize_Jarak2Pop$`Rata-rata`[1]  
  
sd_jarakUINRIL <- summarize_Jarak2Pop$`Simpangan baku`[3]  
sd_jarakITERA <- summarize_Jarak2Pop$`Simpangan baku`[1]
```

```
n_jarakUINRIL <- summarize_Jarak2Pop$Jumlah[3]
n_jarakITERA <- summarize_Jarak2Pop$Jumlah[1]
```

5.6.1.1 Two-tailed Test

5.6.1.1.1 Merumuskan hipotesis Sama seperti sebelumnya, langkah pertama dalam uji hipotesis adalah menyusun hipotesis nol (H_0) dan hipotesis alternatif (H_1). Untuk kasus two-tailed test, hipotesis alternatif menyatakan adanya perbedaan rata-rata tanpa menyebutkan arah (lebih besar atau lebih kecil). Artinya, kita hanya ingin tahu apakah terdapat perbedaan signifikan atau tidak, tanpa peduli siapa yang lebih besar. Hipotesis nol menyatakan tidak ada perbedaan, atau selisih rata-rata sama dengan nol.

```
cat("H0: mu_jarakUINRIL - mu_jarakITERA = 0",
    "\n\nH1: mu_jarakUINRIL - mu_jarakITERA != 0")
```

```
## H0: mu_jarakUINRIL - mu_jarakITERA = 0
##
## H1: mu_jarakUINRIL - mu_jarakITERA != 0
```

5.6.1.1.2 Memilih distribusi sampling, wilayah, dan titik kritis Setelah hipotesis dirumuskan, kita harus menentukan distribusi sampling yang dipakai serta wilayah kritisnya. Karena ukuran sampel cukup besar, distribusi normal standar (Z) digunakan sebagai acuan. Untuk two-tailed test, wilayah kritis terbagi dua sama besar di kedua sisi kurva distribusi. Nilai kritis diperoleh dari tabel distribusi normal pada $/2$, lalu dibuat dalam bentuk absolut karena titik kritis simetris di kedua sisi.

```
# Hipotesis yang dipakai tidak berarah, maka wilayah kritis adalah alpha/2
alpha <- 0.001

Z.crit_2tail <- abs(qnorm(alpha/2)) # Nilai yang digunakan adalah nilai absolut

cat("Titik kritis uji two-tailed:", Z.crit_2tail)
```

```
## Titik kritis uji two-tailed: 3.290527
```

5.6.1.1.3 Menghitung statistik uji Setelah mengetahui titik kritis, kita hitung nilai statistik uji (Z) dari data yang dimiliki. Nilai ini diperoleh dengan memasukkan rata-rata, simpangan baku, dan ukuran sampel ke fungsi `ht_mean_2popIn`. Karena pengujian dilakukan dua sisi, nilai Z hasil perhitungan diabsolutkan agar bisa dibandingkan langsung dengan titik kritis.

```
# Hasil perhitungan statistik uji diabsolutkan
Z_meanJarak2Pop <- abs(ht_mean_2popIn(mean_jarakUINRIL, mean_jarakITERA,
                                       sd_jarakUINRIL, sd_jarakITERA,
                                       n_jarakUINRIL, n_jarakITERA))

Z_meanJarak2Pop

## [1] 7.743731
```

5.6.1.1.4 Menarik kesimpulan Langkah terakhir adalah membandingkan nilai statistik uji dengan titik kritis. Jika Z hitung lebih kecil dari Z kritis, maka H_0 gagal ditolak, artinya perbedaan rata-rata jarak antara dua populasi kemungkinan hanyalah kebetulan. Jika Z hitung lebih besar, maka H_0 ditolak, yang berarti terdapat cukup bukti untuk menyatakan adanya perbedaan.

```
if (Z_meanJarak2Pop < Z.crit_2tail) {
  tolak_h0 <- FALSE
  cat("Hipotesis kosong gagal ditolak. Perbedaan rata-rata jarak mahasiswa UINRIL dengan ITERA dari sampel signifikan")
} else {
  tolak_h0 <- TRUE
  cat("Hipotesis kosong ditolak. Perbedaan rata-rata jarak mahasiswa UINRIL dengan ITERA dari sampel signifikan")
}
```

```
## Hipotesis kosong ditolak. Perbedaan rata-rata jarak mahasiswa UINRIL dengan ITERA dari sampel signifikan
```

5.6.1.2 One-tailed Test

5.6.1.2.1 Merumuskan hipotesis Untuk one-tailed test, hipotesis alternatif (H_1) menyebutkan arah perbedaan yang spesifik, misalnya rata-rata populasi 1 lebih kecil dari populasi 2. Berbeda dengan two-tailed test yang hanya memeriksa “ada perbedaan”, one-tailed test lebih fokus untuk mengetahui apakah perbedaan itu terjadi sesuai arah yang dihipotesiskan. Hipotesis nol (H_0) tetap menyatakan tidak ada perbedaan, tetapi pengujian difokuskan hanya pada satu sisi distribusi.

```
cat("H0: mu_jarakUINRIL - mu_jarakITERA = 0",
    "\n\nH1: mu_jarakUINRIL - mu_jarakITERA < 0")
```

```
## H0: mu_jarakUINRIL - mu_jarakITERA = 0
##
## H1: mu_jarakUINRIL - mu_jarakITERA < 0
```

5.6.1.2.2 Memilih distribusi sampling, wilayah, dan titik kritis Dalam pengujian one-tailed, kita perlu menentukan sisi mana yang akan digunakan sebagai wilayah kritis. Jika hipotesis alternatif menyebutkan “lebih kecil”, maka wilayah kritis berada di sisi kiri distribusi. Sebaliknya, jika hipotesis menyebutkan “lebih besar”, wilayah kritis diletakkan di sisi kanan.

```
tail <- "kiri"

if (tail == "kiri") {
  Z.crit_1tail <- qnorm(alpha)
} else {
  Z.crit_1tail <- qnorm(1-alpha)
}

cat("Titik kritis uji one-tailed:", Z.crit_1tail)
```

```
## Titik kritis uji one-tailed: -3.090232
```


5.6.1.2.3 Menghitung Statistik uji Selanjutnya, kita menghitung nilai statistik uji menggunakan fungsi `ht_mean_2popIn`. Berbeda dengan two-tailed test, pada one-tailed test nilai Z tidak diabsolutkan karena arah perbedaan menjadi penting.

```
# Hasil perhitungan statistik uji TIDAK diabsolutkan
Z_meanJarak2Pop <- ht_mean_2popIn(mean_jarakUINRIL, mean_jarakITERA,
                                   sd_jarakUINRIL, sd_jarakITERA,
                                   n_jarakUINRIL, n_jarakITERA)

Z_meanJarak2Pop
```

```
## [1] -7.743731
```

5.6.1.2.4 Menarik kesimpulan Tahap terakhir adalah menarik kesimpulan dengan cara membandingkan nilai statistik uji terhadap titik kritis. Jika nilai Z hitung jatuh pada wilayah kritis (misalnya lebih kecil dari Z kritis kiri), maka hipotesis nol ditolak. Artinya, ada bukti cukup untuk menyatakan rata-rata jarak mahasiswa ITERA memang lebih kecil dari mahasiswa UINRIL sesuai hipotesis arah. Jika tidak, maka hipotesis nol gagal ditolak, dan kita simpulkan bahwa arah perbedaan yang diduga tidak terbukti secara signifikan.

```
if (Z_meanJarak2Pop < Z.crit_1tail) {
  tolak_h0 <- TRUE
  cat("Hipotesis kosong ditolak. Mahasiswa ITERA secara rata-rata tinggal lebih jauh dari kampus dibanding UINRIL.")
} else {
  tolak_h0 <- FALSE
  cat("Hipotesis kosong gagal ditolak. Tidak cukup bukti menyatakan mahasiswa ITERA tinggal lebih jauh dari kampus dibanding UINRIL.")
}
```

```
## Hipotesis kosong ditolak. Mahasiswa ITERA secara rata-rata tinggal lebih jauh dari kampus dibandingkan UINRIL.
```

5.6.2 Proporsi

Pada bagian ini kita akan melakukan uji hipotesis terhadap proporsi dua populasi yang independen. Berbeda dengan rata-rata, proporsi digunakan ketika data berbentuk kategori (misalnya pilihan tinggal di kos/asrama atau tinggal bersama keluarga). Prinsip pengujian tetap sama, yaitu membandingkan apakah perbedaan proporsi antar populasi signifikan atau hanya karena kebetulan sampel.

5.6.2.0.1 Membuat fungsi Sama seperti sebelumnya, langkah pertama adalah membuat fungsi khusus untuk menghitung nilai statistik uji pada uji hipotesis proporsi dua populasi independen. Fungsi ini menerima input berupa proporsi sampel masing-masing populasi (\hat{p}_1 dan \hat{p}_2) serta ukuran sampelnya (n_1 dan n_2). Di dalam fungsi, dihitung terlebih dahulu proporsi gabungan (\hat{p}_{gab}) yang mewakili keseluruhan sampel. Kemudian dihitung standard error (SE), dan akhirnya didapatkan nilai Z hitung.

```
# Membuat fungsi uji hipotesis (hypothesis testing, ht) untuk rata-rata (mean)
# dengan 1 populasi (2pop) Independen (In)
# Keterangan input:
#   - xbar1 : statistik rata-rata populasi #1
#   - xbar2 : statistik rata-rata populasi #2
#   - sd1   : statistik simpangan baku populasi #1
#   - sd2   : statistik simpangan baku populasi #2
#   - n1    : ukuran sampel dari populasi #1
```

```
# - n2 : ukuran sampel dari populasi #2
ht_prop_2popIn <- function(p_hat1, p_hat2, n1, n2) {
  p_gab <- ((n1*p_hat1 + n2*p_hat2))/(n1 + n2) # Total proporsi yang diuji
                                                    # untuk kedua populasi

  se <- sqrt(
    (p_gab*(1-p_gab)) * ( # Indentasi diberikan untuk memudahkan pembacaan
      (1/n1) + (1/n2)      # karena banyaknya kurung
    )
  )
  Z_hitung <- (p_hat1 - p_hat2)/se # menghitung nilai Z dari statistik
  return(Z_hitung)
}
```

5.6.2.0.2 Mendeklarasikan variabel uji Setelah fungsi dibuat, kita perlu menyiapkan data yang akan diuji. Pertama, data mahasiswa dikelompokkan berdasarkan kampus dan tipe_tinggal_baku. Selanjutnya dihitung proporsi tiap kategori (misalnya proporsi mahasiswa yang tinggal di kos/asrama). Data kemudian diubah dari format long ke wide sehingga tiap kategori menjadi variabel. Dari hasil ini kita bisa mengambil proporsi masing-masing populasi, serta ukuran sampel total pada tiap kampus.

```
summarize_tempatTinggal2Pop <- data_mahasiswa |>
  group_by(kampus, tipe_tinggal_baku) |>
  summarize(Jumlah = n()) |>
  mutate(proporsi = Jumlah/sum(Jumlah)) |> # menghitung proporsi tiap kategori.
  select(-Jumlah) |>
  spread(tipe_tinggal_baku, proporsi) # mengubah data long menjadi wide (kolom
```

`summarise()` has grouped output by 'kampus'. You can override using the `.groups` argument.

```
summarize_tempatTinggal2Pop
```

```
## # A tibble: 3 x 3
## # Groups:   kampus [3]
##   kampus `Kos/Asrama` `Rumah Keluarga/Pribadi`
##   <chr>      <dbl>          <dbl>
## 1 ITERA      0.550            0.450
## 2 UBL        0.224            0.776
## 3 UINRIL     0.46             0.54
```

```
proporsi_kosAsrama_ITERA <- summarize_tempatTinggal2Pop$`Kos/Asrama`[1]
proporsi_kosAsrama_UINRIL <- summarize_tempatTinggal2Pop$`Kos/Asrama`[3]
```

```
n_kosAsrama_2Pop <- data_mahasiswa |>
  count(kampus)
```

```
n_kosAsrama_ITERA <- n_kosAsrama_2Pop$n[1]
n_kosAsrama_UINRIL <- n_kosAsrama_2Pop$n[3]
```

Two-tailed Test

```

# Gunakan CI = 93%, artinya alpha = 7% = 0.07
alpha <- 0.07

# Hitung Z kritis 2-tailed
Z.crit_2tail <- abs(qnorm(alpha/2))

# Hitung statistik uji proporsi 2 populasi
# Asumsi: Menggunakan data proporsi kos/asrama ITERA vs UINRIL yang sudah disiapkan sebelumnya
Z_prop2Pop <- abs(ht_prop_2popIn(proporsi_kosAsrama_UINRIL, proporsi_kosAsrama_ITERA,
                                n_kosAsrama_UINRIL, n_kosAsrama_ITERA))

cat("Titik Kritis:", Z.crit_2tail, "\nZ Hitung:", Z_prop2Pop, "\n")

```

```

## Titik Kritis: 1.811911
## Z Hitung: 2.597132

```

```

if (Z_prop2Pop < Z.crit_2tail) {
  cat("Hipotesis nol gagal ditolak.")
} else {
  cat("Hipotesis nol ditolak.")
}

```

```

## Hipotesis nol ditolak.

```

```

# Gunakan CI = 96%, artinya alpha = 4% = 0.04
alpha <- 0.04
tail <- "kiri" # Asumsi uji lebih kecil (UINRIL < ITERA misalnya)

# Hitung Z kritis 1-tailed
if (tail == "kiri") {
  Z.crit_1tail <- qnorm(alpha)
} else {
  Z.crit_1tail <- qnorm(1-alpha)
}

# Hitung statistik uji (tidak absolut)
Z_prop2Pop <- ht_prop_2popIn(proporsi_kosAsrama_UINRIL, proporsi_kosAsrama_ITERA,
                              n_kosAsrama_UINRIL, n_kosAsrama_ITERA)

cat("Titik Kritis:", Z.crit_1tail, "\nZ Hitung:", Z_prop2Pop, "\n")

```

5.6.2.1 One-tailed Test

```

## Titik Kritis: -1.750686
## Z Hitung: -2.597132

```

```

if (Z_prop2Pop < Z.crit_1tail) {
  cat("Hipotesis nol ditolak (Signifikan).")
} else {
  cat("Hipotesis nol gagal ditolak.")
}

```

Hipotesis nol ditolak (Signifikan).

5.7 Uji Hipotesis 2 Populasi Dependen

Menguji hipotesis 2 populasi yang dependen pada dasarnya sama persis dengan uji hipotesis 1 populasi, karena statistik yang diuji adalah **perbedaan nilai sampel**. Biasanya berbentuk “sebelum-sesudah”. Contoh: mengukur nilai mahasiswa sebelum dan sesudah diberi pelatihan. Bedanya dengan independen : kita tidak bandingkan dua kelompok berbeda, tapi pasangan data dari kelompok yang sama dalam dua kondisi.

Perbedaan nilai sampel jumlah nilainya hanya 1, sehingga bisa diperlakukan seperti halnya perhitungan uji hipotesis 1 populasi. Simpangan baku yang diketahui pun adalah simpangan baku **perbedaan nilainya, sehingga jumlah nilainya hanya 1 juga**.

5.8 Soal Latihan

6 Modul-6: Analisis Hubungan Korelatif: Korelasi Variabel Nominal dan Ordinal

Setelah mempelajari modul ini, Anda diharapkan dapat:

1. mampu menghasilkan koefisien korelasi variabel di tingkat nominal dengan perangkat lunak komputer **STP-9.2**
2. mampu menghasilkan koefisien korelasi variabel di tingkat ordinal dengan perangkat lunak komputer **STP-10.2**

6.1 Pendahuluan

Analisis korelasi adalah salah satu teknik analisis yang termasuk ke dalam lingkup statistika bivariat, statistika yang analisisnya melibatkan dua variabel **sekaligus**. Pada praktikum-praktikum sebelumnya kita hanya melibatkan satu variabel saja, misalnya:

- pada praktikum ke-2, kita mengaplikasikan persentase hanya pada variabel **Jenis.Kelamin, Fakultas, Tingkat.Semester, Uang.Saku**, atau **jenis.tempat.tinggal** saja, tetapi tidak menghubungkannya satu sama lain,

```
data.ubl |>
  tbl_summary(include = c(Jenis.Kelamin,
                          Fakultas,
                          Tingkat.Semester,
                          Uang.Saku,
                          jenis.tempat.tinggal)) |>
  as_flex_table()
```

- pada praktikum ke-3, kita membuat histogram untuk variabel **Umur** saja

```

histogram <- ggplot(data.ubl.cleaned) +
  geom_histogram(mapping = aes(x = Umur), binwidth = 1, fill = "darkseagreen", color = "white") +
  labs(
    title = "Distribusi Umur Mahasiswa",
    x = "Umur (Tahun)",
    y = "Frekuensi"
  ) +
  theme_minimal()

histogram

```

- pada praktikum-praktikum tentang analisis statistik inferensial, kita hanya tertarik pada proporsi atau rata-rata suatu variabel saja, yang pastinya tunggal.

Dalam praktikum ini kita akan mempraktikkan analisis korelasi pada dua variabel **secara bersamaan**. Ketika kita tertarik pada analisis untuk dua variabel secara bersamaan, artinya kita sedang menganalisis sebuah **keterkaitan** atau sebuah **hubungan**. Dalam praktikum ini, hubungan itu disebut **korelasi**.

Korelasi yang akan kita pelajari adalah korelasi pada pasangan variabel dengan tingkat pengukuran nominal dan juga ordinal.

6.2 Pustaka (*Libraries*) yang Diperlukan dan Memuat Data

Seperti biasa, kita perlu memuat pustaka (*libraries*) yang diperlukan dalam pengolahan data kita. Untuk memuat tabel silang kita akan gunakan pustaka `gtsummary` kembali seperti pada praktikum ke-2

```

library(tidyverse)
library(readr)

```

Memuat dataset

Kita akan menggunakan dataset keempat kampus di Kota Bandar Lampung dan sekitarnya sebagai bahan. Tulis ulang dan jalankan baris perintah berikut untuk mengolah data keempat kampus

```

# Membaca data
data_mahasiswa <- read_csv2("datasets/Data Praktikum 06.csv")

```

```
## i Using "','" as decimal and '."' as grouping mark. Use `read_delim()` for more control.
```

```
## Rows: 1557 Columns: 22
```

```
## -- Column specification -----
```

```
## Delimiter: ";"
```

```
## chr (8): Kampus_PT, Jenis Kelamin, Fakultas, Prodi, Tingkat Semester, Uang Saku, kendaraan utama, j
```

```
## dbl (14): Nomor urut, Umur, jumlah mobil, jumlah motor, jumlah sepeda, jarak, biaya sepekan, Jumlah l
```

```
##
```

```
## 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.
```

Mengatur Faktor untuk Variabel Kategoris

Setelah membaca data, kita perlu mengatur variabel-variabel nominal dan ordinal dengan mengatur nilai-nilainya berupa *factor*.

Namun sebelum itu, kita akan membuat vektor-vektor untuk kita aplikasikan menjadi *factor* dengan memanfaatkan `group_by` dan `summarize` yang berasal dari pustaka `dplyr`.

```

# MEMBUAT FAKTOR UNTUK KAMPUS
# Meringkas data berdasarkan kolom Kampus PT
faktor_kampus <- data_mahasiswa |>
  group_by(kampus = Kampus_PT) |>
  summarize(jumlah = n())
# Membuat vektor untuk faktor Kampus PT dari kolom nama Kampus
faktor_kampus <- faktor_kampus$kampus

# MEMBUAT FAKTOR UNTUK FAKULTAS
# Meringkas data berdasarkan kolom Fakultas
faktor_fakultas <- data_mahasiswa |>
  group_by(fakultas = Fakultas) |>
  summarize(jumlah = n())
# Membuat vektor untuk faktor Fakultas dari kolom nama fakultas
faktor_fakultas <- faktor_fakultas$fakultas

# MEMBUAT FAKTOR UNTUK PROGRAM STUDI
# Meringkas data berdasarkan kolom Prodi
faktor_prodi <- data_mahasiswa |>
  group_by(prodi = Prodi) |>
  summarize(jumlah = n())
# Membuat vektor untuk faktor Prodi dari kolom nama Prodi
faktor_prodi <- faktor_prodi$prodi

# MEMBUAT FAKTOR UNTUK JENIS KELAMIN
# Meringkas data berdasarkan kolom Jenis Kelamin
faktor_jk <- data_mahasiswa |>
  group_by(jk = `Jenis Kelamin`) |>
  summarize(jumlah = n())
# Membuat vektor untuk faktor Jenis Kelamin dari kolom nama Jenis Kelamin
faktor_jk <- faktor_jk$jk

# MEMBUAT FAKTOR UNTUK TINGKAT SEMESTER
# Meringkas data berdasarkan kolom Tingkat Semester
faktor_semester <- data_mahasiswa |>
  group_by(semester = `Tingkat Semester`) |>
  summarize(jumlah = n())
# Membuat vektor faktor Tingkat Semester dari kolom Tingkat Semester
faktor_semester <- faktor_semester$semester

# MEMBUAT FAKTOR UNTUK UANG SAKU
# Meringkas data berdasarkan kolom Uang Saku
faktor_uangsaku <- data_mahasiswa |>
  group_by(uang = `Uang Saku`) |>
  summarize(jumlah = n())
# Memperbaiki urutan uang saku
faktor_uangsaku$uang[c(1,4)] <- faktor_uangsaku$uang[c(4,1)]
faktor_uangsaku$jumlah[c(1,4)] <- faktor_uangsaku$jumlah[c(4,1)]

faktor_uangsaku$uang[c(2,4)] <- faktor_uangsaku$uang[c(4,2)]
faktor_uangsaku$jumlah[c(2,4)] <- faktor_uangsaku$jumlah[c(4,2)]

faktor_uangsaku$uang[c(3,4)] <- faktor_uangsaku$uang[c(4,3)]

```

```
faktor_uangsaku$jumlah[c(3,4)] <- faktor_uangsaku$jumlah[c(4,3)]

# Membuat vektor faktor Tingkat Uang Saku dari kolom Uang Saku
faktor_uangsaku <- faktor_uangsaku$uang

# MEMBUAT FAKTOR UNTUK KENDARAAN UTAMA
# Meringkas data berdasarkan kolom Kendaraan Utama
faktor_kendaraanutama <- data_mahasiswa |>
  group_by(kendaraan = `kendaraan utama`) |>
  summarize(jumlah = n())
# Membuat vektpr faktor untuk kendaraan utama dari kolom 'kendaraan'
faktor_kendaraanutama <- faktor_kendaraanutama$kendaraan

# MEMBUAT FAKTOR JENIS TEMPAT TINGGAL
# Meringkas data berdasarkan kolom Jenis Tempat Tinggal
faktor_jenistempattinggal <- data_mahasiswa |>
  group_by(tempat_tinggal = `jenis tempat tinggal`) |>
  summarize(jumlah = n())
# Membuat vektor faktor untuk jenis tempat tinggal berdasarkan kolom Jenis Tempat Tinggal
faktor_jenistempattinggal <- faktor_jenistempattinggal$tempat_tinggal
```

Barulah selanjutnya kita bisa mengatur *factor* untuk setiap variabel kategoris

```
data_mahasiswa <- data_mahasiswa |>
  mutate (Kampus_PT = factor(Kampus_PT, levels = faktor_kampus),
    `Jenis Kelamin` = factor(`Jenis Kelamin`, levels = faktor_jk),
    Fakultas = factor(Fakultas, levels = faktor_fakultas),
    Prodi = factor(Prodi, levels = faktor_prodi),
    `Tingkat Semester` = factor(`Tingkat Semester`, levels = faktor_semester,
      ordered = TRUE), # Variabel ordinal
    `Uang Saku` = factor(`Uang Saku`, levels = faktor_uangsaku,
      ordered = TRUE), # Variabel ordinal
    `kendaraan utama` = factor(`kendaraan utama`, levels = faktor_kendaraanutama),
    `jenis tempat tinggal` = factor(`jenis tempat tinggal`, levels = faktor_jenistempattinggal)
  )
```

Kita dapat memeriksa hasil pekerjaan kita dengan perintah `glimpse`

```
# Memeriksa hasil pengaturan factor
glimpse(data_mahasiswa)
```

```
## Rows: 1,557
## Columns: 22
## $ Kampus_PT          <fct> ITERA, ITERA, ITERA, ITERA, ITERA, ITERA, ITERA, ITERA, ITERA, IT
## $ `Nomor urut`       <dbl> 17, 211, 342, 42, 56, 254, 267, 289, 330, 259, 48, 169, 225, 242,
## $ `Jenis Kelamin`    <fct> Perempuan, Perempuan, Perempuan, Perempuan, Laki-laki, Perempuan,
## $ Umur               <dbl> 20, 20, 22, 21, 22, 22, 21, 21, 20, 23, 19, 23, 21, 19, 22, 23, 2
## $ Fakultas           <fct> FTIK, FTIK, FTI, FTIK, FTIK, FTIK, FTIK, FTIK, FTI, FTIK, FTIK, F
## $ Prodi              <fct> Perencanaan Wilayah dan Kota, Perencanaan Wilayah dan Kota, Tekn
## $ `Tingkat Semester` <ord> Semester 3-4, Semester 5-6, Semester 7-8, Semester 7-8, Semester
## $ `Uang Saku`        <ord> "1 - 2 Jt", "2,1 - 3 Jt", "2,1 - 3 Jt", "2,1 - 3 Jt", "1 - 2 Jt",
```

```
## $ `jumlah mobil` <dbl> 1, 0, 3, 3, 1, 3, 3, 3, 3, 2, 1, 0, 3, 2, 2, 2, 1, 0, 2, 0, 1, 2,
## $ `jumlah motor` <dbl> 0, 1, 2, 3, 0, 2, 2, 3, 2, 2, 0, 1, 1, 2, 1, 1, 1, 1, 0, 0, 1,
## $ `jumlah sepeda` <dbl> 1, 1, 3, 3, 0, 2, 3, 4, 2, 2, 1, 0, 1, 2, 1, 2, 1, 0, 0, 0, 0, 0,
## $ `kendaraan utama` <fct> Sepeda motor pribadi, Mobil pribadi, Mobil pribadi, Mobil pribadi
## $ `jenis tempat tinggal` <fct> Kos sendiri, Rumah pribadi, Rumah pribadi, Rumah pribadi, Rumah m
## $ jarak <dbl> 3.47, 4.88, 3.09, 5.46, 3.70, 6.18, 4.87, 5.08, 3.82, 7.51, 7.89,
## $ `biaya sepekan` <dbl> 400, 280, 225, 200, 200, 200, 200, 200, 200, 180, 150, 150, 150,
## $ `Jumlah Perjalanan Senin` <dbl> 2, 4, 2, 0, 4, 0, 3, 4, 2, 0, 1, 4, 2, 0, 3, 2, 2, 0, 3, 2, 4, 4,
## $ `Jumlah Perjalanan Selasa` <dbl> 3, 4, 2, 3, 3, 0, 4, 4, 2, 2, 1, 3, 2, 2, 2, 0, 2, 3, 4, 2, 4, 4,
## $ `Jumlah Perjalanan Rabu` <dbl> 3, 5, 0, 4, 3, 3, 4, 4, 0, 2, 1, 2, 2, 2, 4, 2, 2, 2, 3, 2, 4, 4,
## $ `Jumlah Perjalanan Kamis` <dbl> 0, 4, 2, 3, 2, 0, 4, 4, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 4, 2, 4, 4,
## $ `Jumlah Perjalanan Jumat` <dbl> 0, 5, 0, 3, 2, 2, 4, 5, 2, 2, 1, 3, 3, 4, 4, 2, 2, 0, 4, 4, 5, 7,
## $ `Jumlah Perjalanan Sabtu` <dbl> 2, 0, 3, 0, 3, 3, 0, 2, 2, 0, 1, 2, 2, 0, 0, 2, 2, 2, 4, 0, 0, 0,
## $ `Jumlah Perjalanan Ahad` <dbl> 4, 3, 0, 2, 2, 0, 3, 2, 2, 0, 1, 2, 0, 0, 2, 0, 2, 2, 2, 2, 0, 0,
```

6.3 Pembuatan Tabel Silang

Setelah kita memastikan data kita bersih, kita siap untuk menganalisis korelasi antarvariabel kategoris.

Korelasi antarvariabel kategoris memerlukan “bahan baku” berupa **tabel silang** atau *cross table*. Kebanyakan literatur juga menggunakan istilah *contingency table*. Tabel silang adalah tabel dua arah, yakni yang baik baris maupun kolomnya adalah **nilai-nilai dari dua variabel** yang disilangkan.

Chunk berikut memperlihatkan bagaimana membuat tabel silang antara variabel **jenis tempat tinggal** dengan **kendaraan utama**.

```
library(gtsummary)
library(flextable)

# Menghasilkan tabel silang antara variabel kendaraan utama dan jenis tempat tinggal
# untuk keperluan presentasi
data_mahasiswa |>
  select(`jenis tempat tinggal`, `kendaraan utama`) |>
  tbl_summary(by = `jenis tempat tinggal`) |>
  as_flex_table()
```

Characteristic	Asrama N = 25 ¹	Kos bersama-sama N = 120 ¹	Kos sendiri N = 468 ¹	Rumahnya sendiri
kendaraan utama				
Berjalan kaki	23 (92%)	20 (17%)	39 (8.3%)	
Layanan online	0 (0%)	19 (16%)	86 (18%)	
Menumpang dengan teman/keluarga	0 (0%)	18 (15%)	24 (5.1%)	
Mobil pribadi	0 (0%)	6 (5.0%)	15 (3.2%)	
Sepeda	0 (0%)	0 (0%)	8 (1.7%)	
Sepeda motor pribadi	2 (8.0%)	57 (48%)	296 (63%)	

¹n (%)

Tabel silang yang merupakan hasil dari *script* di atas adalah seperti berikut. Baris menunjukkan kategori dalam variabel **kendaraan utama**, sementara kolom menunjukkan kategori dalam variabel **jenis tempat**

tinggal. Nilai-nilai dalam sel-selnya adalah frekuensi/jumlah data yang termasuk ke dalam kondisi berupa kategori kedua variabel.

Perhatikan cara menginterpretasi informasi dari tabel tersebut:

- jumlah pengguna layanan online yang tinggal di kos bersama-sama adalah 19 orang, yang setara dengan 16%.
- tidak ada pengguna sepeda yang tinggal di Asrama, Kos bersama-sama, atau mengontrak rumah secara pribadi. Keberadaannya juga langka pada kalangan mahasiswa yang tinggal di rumah pribadi atau di saudaranya.

Characteristic	Asrama N = 25 ¹	Kos bersama- sama N = 120 ¹	Kos sendiri N = 468 ¹	Rumah mengontrak bersama N = 223 ¹	Rumah mengontrak pribadi N = 43 ¹	Rumah pribadi N = 553 ¹	Rumah saudara N = 125 ¹
kendaraan utama							
Berjalan kaki	23 (92%)	20 (17%)	39 (8.3%)	17 (7.6%)	11 (26%)	0 (0%)	1 (0.8%)
Layanan online	0 (0%)	19 (16%)	86 (18%)	28 (13%)	6 (14%)	37 (6.7%)	12 (9.6%)
Menumpang dengan teman/keluarga	0 (0%)	18 (15%)	24 (5.1%)	40 (18%)	0 (0%)	11 (2.0%)	4 (3.2%)
Mobil pribadi	0 (0%)	6 (5.0%)	15 (3.2%)	5 (2.2%)	3 (7.0%)	110 (20%)	13 (10%)
Sepeda	0 (0%)	0 (0%)	8 (1.7%)	8 (3.6%)	0 (0%)	1 (0.2%)	1 (0.8%)
Sepeda motor pribadi	2 (8.0%)	57 (48%)	296 (63%)	125 (56%)	23 (53%)	394 (71%)	94 (75%)

¹n (%)

Figure 3: Tabel Silang antara variabel jenis kendaraan utama dengan jenis tempat tinggal

Selain dengan `tbl_summary()` yang gunanya cenderung untuk presentasi, kita dapat menggunakan perintah `table()` untuk menghasilkan tabel silang yang siap dioperasikan ke dalam perhitungan koefisien korelasi di tahap selanjutnya. Perintah `table()` mengambil dua masukan, yakni vektor variabel-variabel yang kita akan analisis, dipisahkan dengan tanda koma (,).

```
table(data_mahasiswa$`jenis tempat tinggal`, # vektor variabel-variabel ini kita akses dari
      data_mahasiswa$`kendaraan utama`)      # dataset menggunakan operator '$'
```

```
##
##               Berjalan kaki Layanan online Menumpang dengan teman/keluarga Mobil pribadi
## Asrama                23                0                0                0
## Kos bersama-sama      20                19                18                0
## Kos sendiri           39                86                24                1
## Rumah mengontrak bersama 17                28                40                5
## Rumah mengontrak pribadi 11                6                0                3
## Rumah pribadi         0                37                11                11
## Rumah saudara         1                12                4                1
```

Latihan: buatlah tabel silang antara variabel Tingkat Semester dan Uang Saku baik dengan `gt_summary` maupun dengan `table()`!

6.4 Analisis Korelasi Antarvariabel Nominal

Dalam analisis korelasi variabel nominal, kita mengukur **keberadaan** dan juga **kekuatan hubungan**. Ukuran yang kita pakai untuk menyatakan hal tersebut adalah **koefisien korelasi variabel nominal**, yang di antaranya adalah koefisien ϕ (**phi**), **V Cramer** dan **koefisien C**. Ketiga koefisien tersebut dihitung dari nilai χ^2 (“chi kuadrat”).

Selain itu, terdapat juga λ (**lambda**) yang merupakan koefisien korelasi variabel nominal berbasis *error*, atau disebut juga PRE (*proportional reduction of error*).

Ukuran dari keempat koefisien tersebut adalah besarnya yang berkisar **antara 0 dan 1** dengan **0 berarti tidak ada hubungan sama sekali** dan **1 berarti hubungan yang ada sangat kuat**.

6.4.1 Koefisien Korelasi Variabel Nominal Berbasis χ^2

Yang pertama adalah koefisien korelasi berbasis nilai χ^2 . Karena berbasis nilai χ^2 , maka kita harus menghitung terlebih dahulu nilai χ^2 antara dua variabel kita.

Untuk menghitung χ^2 kita menggunakan dua perintah utama dalam R, yakni `table()` dan `chisq.test()`. Perintah `table()` berfungsi membuat tabel silang yang berguna sebagai “bahan baku” perhitungan χ^2 kita, kemudian perintah `chisq.test()` menghitung nilai χ^2 tersebut.

```
# Menyimpan tabel
t4tinggal_kend <- table(data_mahasiswa$kendaraan utama`,
                        data_mahasiswa$jenis tempat tinggal`)
# Menyimpan hasil chi-square test
chisq_t4tinggal_kend <-
  t4tinggal_kend |>
  chisq.test()
```

```
## Warning in chisq.test(t4tinggal_kend): Chi-squared approximation may be incorrect
```

Seperti yang bisa kita lihat, hasil perhitungan berupa variabel R `chisq_t4tinggal_kend` berada di bagian data, dan jika kita perluas, maka di dalamnya terdapat banyak variabel-variabel lain. Variabel-variabel lain tersebut menyimpan berbagai nilai yang menjadi bagian dari hasil perhitungan. Nilai-nilai yang penting untuk

- `.$statistic` menyimpan nilai χ^2 dari tabel. Ini digunakan untuk menghitung kekuatan korelasi
- `.$parameter` menyimpan nilai *degree of freedom* (df)
- `.$p.value` menyimpan nilai probabilitas untuk menerima H_0 yang berarti tidak ada hubungan antara dua variabel yang dianalisis
- `.$expected` menyimpan tabel frekuensi yang diharapkan dari data kita

Tampilkan seluruh nilai tersebut

```
chisq_t4tinggal_kend$statistic
```

```
## X-squared
## 600.8622
```

```
chisq_t4tinggal_kend$parameter
```

```
## df  
## 30
```

```
chisq_t4tinggal_kend$p.value
```

```
## [1] 1.963807e-107
```

```
chisq_t4tinggal_kend$expected
```

```
##  
##                               Asrama Kos bersama-sama Kos sendiri Rumah mengontrak bersama R  
## Berjalan kaki                1.7822736           8.554913   33.364162           15.897881  
## Layanan online               3.0186256           14.489403   56.508671           26.926140  
## Menumpang dengan teman/keluarga 1.5574823           7.475915   29.156069           13.892742  
## Mobil pribadi                2.4405909           11.714836   45.687861           21.770071  
## Sepeda                       0.2890173           1.387283    5.410405            2.578035  
## Sepeda motor pribadi          15.9120103           76.377649  297.872832          141.935132  
##  
##                               Rumah saudara  
## Berjalan kaki                8.911368  
## Layanan online               15.093128  
## Menumpang dengan teman/keluarga 7.787412  
## Mobil pribadi                12.202954  
## Sepeda                       1.445087  
## Sepeda motor pribadi          79.560051
```

Kita akan menghitung nilai koefisien V Cramer untuk menyatakan korelasi antara jenis kendaraan yang digunakan sebagai moda ke kampus dan jenis tempat tinggal mahasiswa menggunakan rumus berikut dan berdasarkan nilai χ^2 yang kita dapatkan dari tes sebelumnya.

$$V = \sqrt{\frac{\chi^2}{n \times \min(c-1, r-1)}}$$

```
# Menyimpan nilai Chi-squared menggunakan metode "as.numeric"  
chisq_t4tinggal_kend <- as.numeric(chisq_t4tinggal_kend$statistic)  
  
# Menghitung jumlah data  
n <- data_mahasiswa |>  
  summarize(jumlah = n())  
n <- n$jumlah  
  
# Menghitung jumlah kategori dari 'jenis tempat tinggal' sebagai jumlah kolom  
kolom <- data_mahasiswa |>  
  group_by(jenis = `jenis tempat tinggal`) |>  
  summarize(jumlah = n())  
kolom <- length(kolom$jenis)  
  
# Menghitung jumlah kategori dari 'kendaraan utama' sebagai jumlah baris
```

```

baris <- data_mahasiswa |>
  group_by(kend = `kendaraan utama`) |>
  summarize(jumlah = (n))
baris <- length(baris$kend)

# Menghitung koefisien V
v_t4tinggal_kend <- sqrt(chisq_t4tinggal_kend/(n * min(kolom-1, baris-1)))
v_t4tinggal_kend

```

```
## [1] 0.2778166
```

Kesimpulan: Hasil dari perhitungan koefisien V adalah **0,253**. Angka ini menunjukkan hubungan yang sedikit lemah, karena berada di bawah nilai tengah-tengah yaitu 0,5. Artinya, hubungan antara jenis tempat tinggal dengan jenis kendaraan yang sering digunakan ke kampus tidak begitu kuat.

Cara lain: kita dapat menggunakan pustaka `rcompanion` untuk menghitung koefisien V Cramer, yaitu dengan perintah `cramerV()`. Perintah ini mengambil langsung vektor kolom variabel yang kita akan analisis, sehingga menggunakan notasi `$` untuk mengakses kolom ``jenis tempat tinggal`` dan ``kendaraan utama``.

Tentunya kita harus menginstal pustaka tersebut jika belum memilikinya dengan perintah `install.packages('rcompanion')`

```
install.packages("rcompanion")
```

```
library(rcompanion)
```

```
cramerV(data_mahasiswa$`jenis tempat tinggal`, data_mahasiswa$`kendaraan utama`)
```

```
## Cramer V
## 0.2778
```

6.4.2 Koefisien Korelasi Variabel Nominal Berbasis *Error*

Untuk analisis korelasi variabel nominal berbasis *error* yang menggunakan koefisien λ kita dapat menggunakan perintah `lambda()` yang ada di dalam pustaka `RCPA3`. Terlebih dahulu tentunya lakukan instalasi pustaka tersebut dengan perintah `install.packages('RCPA3')`.

Perintah `lambda()` mengambil masukan berupa tabel silang seperti yang kita buat pada perhitungan χ^2 , yakni perintah `table()`.

```
install.packages("RCPA3")
```

```
library(RCPA3)
```

```

# Menghitung koefisien lambda
lambda_t4tinggal_kend <- lambda(t4tinggal_kend, digits = 3, detailed = TRUE)
# digits menandakan berapa desimal angka yang kita hasilkan
# detailed menandakan apakah kita memperlihatkan nilai-nilai error yang dihitung

```

6.5 Analisis Korelasi Antarvariabel Ordinal

Dalam analisis korelasi variabel ordinal, kita tidak hanya dapat mengukur keberadaan dan juga kekuatan hubungan, tetapi juga **arah hubungan**. Yang dimaksud dengan ‘arah hubungan’ adalah **searah** atau **berlawanan**. Hubungan searah berarti semakin tinggi nilai variabel #1, maka semakin tinggi juga nilai variabel #2, begitu pula sebaliknya. Akan tetapi, jika arah hubungan berlawanan, semakin tinggi nilai variabel #1 maka semakin rendah nilai variabel #2, begitu pula sebaliknya.

Arah hubungan ditunjukkan oleh **tanda koefisien** dan kekuatannya ditunjukkan oleh **besar angka koefisien**. Oleh karena itu, nilai korelasi pada variabel ordinal berkisar antara -1 hingga $+1$ dengan tanda $-$ untuk menandakan hubungan berlawanan dan $+$ untuk hubungan searah.

Kita dapat menggunakan perintah `GKgamma()` dari pustaka `vcdExtra`. Lakukan instalasi pustaka terlebih dahulu dengan perintah `install.packages('vcdExtra')`.

Perintah `GKgamma()` mengambil masukan sama seperti `chisq.test()`, yakni tabel silang yang dihasilkan dari perintah `table()`.

Dalam praktikum ini, kita akan menganalisis kekuatan dan arah hubungan antara Uang Saku dan Tingkat Semester. Apakah mahasiswa dengan tingkat semester makin tinggi uang sakunya makin tinggi pula, atau sebaliknya?

```
install.packages("vcdExtra")
```

```
library(vcdExtra)
```

```
# Membuat variabel table() sebagai masukan
uangSaku_tingkatSemester <- table(data_mahasiswa$`Uang Saku`,
                                   data_mahasiswa$`Tingkat Semester`)

# Menghitung nilai Gamma dari variabel table() yang sudah dibuat
G_uangSaku_tingkatSemester <- GKgamma(uangSaku_tingkatSemester)

# Menampilkan nilai Gamma
G_uangSaku_tingkatSemester
```

```
## gamma      : -0.009
## std. error  : 0.033
## CI         : -0.073 0.056
```

Seperti halnya nilai variabel hasil perhitungan `chisq.test()`, hasil perhitungan `GKgamma()` juga terdiri atas banyak variabel. Nilai gamma sendiri dapat kita akses dengan `.$gamma`.

Nilai lain yang penting bagi kita adalah nilai `.$C` dan `.$D`. Nilai-nilai tersebut adalah nilai operasi dalam sel tabel silang yang merupakan penyusun dalam perhitungan koefisien tersebut mengikuti rumus berikut:

$$G = \frac{C - D}{C + D}$$

```
# Menampilkan nilai C dan D
print(G_uangSaku_tingkatSemester$C)
```

```
## [1] 283397
```

```
print(G_uangSaku_tingkatSemester$D)
```

```
## [1] 288282
```

7 Modul-7: Analisis Hubungan Korelatif: Korelasi Variabel Metrik

Setelah mempelajari modul ini, Anda diharapkan mampu menghasilkan koefisien korelasi variabel di tingkat metrik dengan perangkat lunak komputer. **STP-11.2**

7.1 Pendahuluan

Analisis korelasi adalah salah satu teknik analisis yang termasuk ke dalam lingkup statistika bivariat, statistika yang analisisnya melibatkan dua variabel **sekaligus**. Pada praktikum sebelumnya kita sudah mengenal koefisien-koefisien berikut untuk menganalisis korelasi antara dua variabel nominal dan ordinal:

- V Cramer dan λ (lambda) untuk dua variabel nominal
- Gamma (G) untuk dua variabel ordinal

Dalam praktikum ini kita akan mempraktikkan analisis korelasi pada **dua variabel metrik**. Pada praktikum sebelumnya kita mengetahui bahwa dari koefisien-koefisien korelasi kita dapat mengetahui:

- **kekuatan** untuk dua variabel nominal
- **kekuatan** dan **arah** untuk dua variabel ordinal

Untuk dua variabel metrik, selain **kekuatan** dan juga **arah hubungan**, kita juga dapat menganalisis **pola** hubungan tersebut, yakni persebaran titik-titik data secara **grafis**.

Dalam praktikum ini kita akan mempraktikkan perhitungan dua jenis koefisien: **Spearman's rho** (ρ) dan **Pearson's r**.

7.2 Pustaka (*Libraries*) yang Diperlukan dan Memuat Data

Seperti biasa, kita perlu memuat pustaka (*libraries*) yang diperlukan dalam pengolahan data kita. Dalam analisis korelasi variabel metrik kita tidak lagi menggunakan tabel silang, tetapi kita langsung menganalisis kolom-kolom yang ada di dataset kita.

```
library(tidyverse)
library(readr)
```

Memuat dataset

Kita akan menggunakan dataset keempat kampus di Kota Bandar Lampung dan sekitarnya sebagai bahan. Tulis ulang dan jalankan baris perintah berikut untuk mengolah data keempat kampus

```
# Membaca data
data_mahasiswa <- read_csv2("datasets/Data Praktikum 07.csv")
```

```
## i Using "','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
## Rows: 1557 Columns: 22
```

```
## -- Column specification -----
```

```
## Delimiter: ";"
```

```
## chr (8): Kampus_PT, Jenis Kelamin, Fakultas, Prodi, Tingkat Semester, Uang Saku, kendaraan utama, j
```

```
## dbl (14): Nomor urut, Umur, jumlah mobil, jumlah motor, jumlah sepeda, jarak, biaya sepekan, Jumlah l
```

```
##
```

```
## 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.
```

Mengatur Faktor untuk Variabel Kategoris: Opsional

Karena kita hanya akan berurusan dengan variabel-variabel metrik, maka kita bisa saja melewati tahap ini.

7.3 Pola Hubungan Data

7.4 6.3. Pola Hubungan Data

Dalam analisis korelasi variabel-variabel metrik kita dapat menganalisis satu lagi sifat hubungan dalam dua variabel yang kita perhitungkan: **pola hubungan data**. Pola hubungan adalah bentuk sebaran titik-titik yang dapat kita lihat dengan diagram pencar (*scatter plot*).

Membuat Diagram Pencar

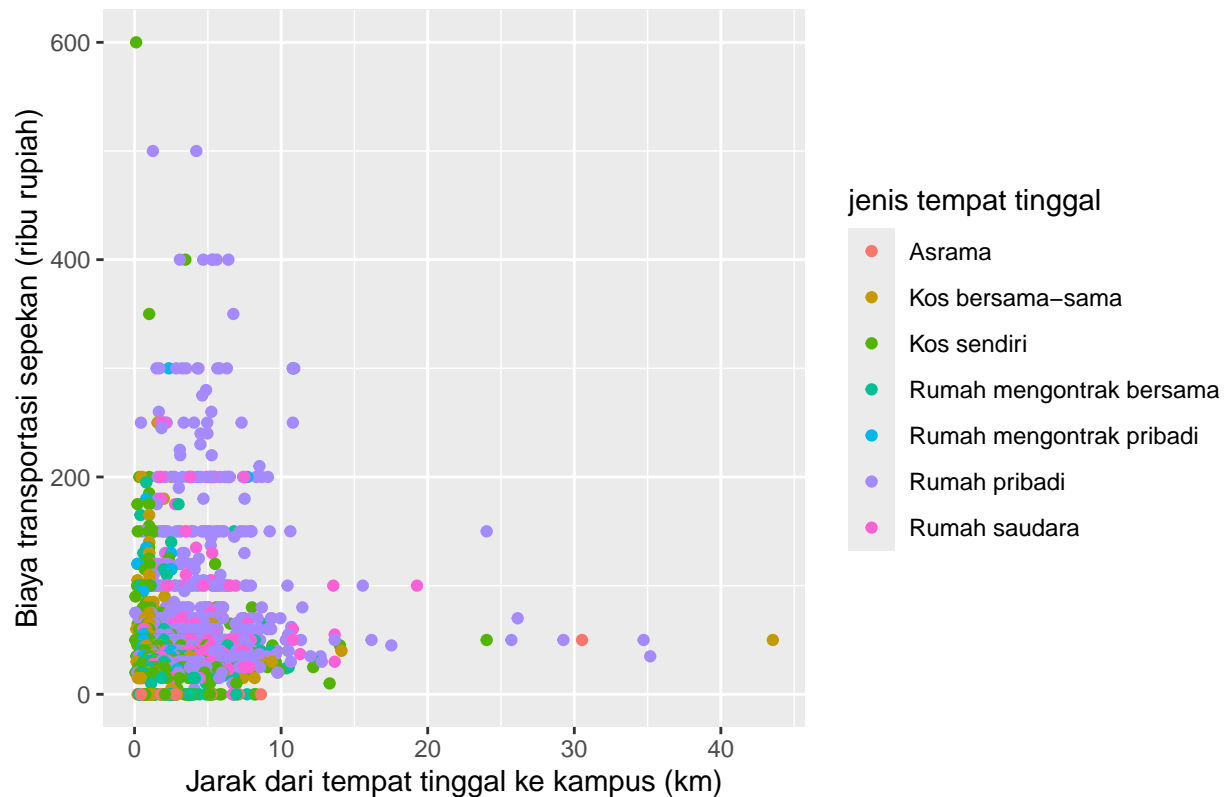
Membuat diagram pencar dapat dilakukan dengan menerapkan perintah `geom_point()` dari pustaka `ggplot2` yang dimuat bersama pustaka `tidyverse`.

Kita akan melihat pola sebaran data kita dilihat dari variabel jarak dari kampus (`jarak`) dengan biaya yang dikeluarkan untuk transportasi selama sepekan (`biaya sepekan`). Variabel-variabel yang akan kita lihat hubungannya itu kita masukkan ke `x` dan `y` yang adalah dalam perintah `aes()`.

```
# Membuat diagram pencar antara variabel jarak dan biaya transportasi sepekan
scatter_plot <- ggplot(data = data_mahasiswa,
  mapping = aes(x = jarak, # variabel di sumbu X
    y = `biaya sepekan`) + # variabel di sumbu Y
  geom_point(aes(color = `jenis tempat tinggal`)) + # perintah untuk menampilkan diagram pencar
  labs(title = "Jarak tempat tinggal vs. Biaya Transportasi Sepekan",
    y = "Biaya transportasi sepekan (ribu rupiah)",
    x = "Jarak dari tempat tinggal ke kampus (km)")

# Menampilkan diagram
scatter_plot
```

Jarak tempat tinggal vs. Biaya Transportasi Sepekan



Dari diagram yang dihasilkan kita dapat menarik interpretasi hubungan antara kedua variabel secara visual:

- Tidak ada kecenderungan arah hubungan antara jarak tempat tinggal ke kampus dengan biaya transportasi per pekan
- Terdapat responden yang tinggal dekat dengan kampus (<10 km) tetapi biayanya tetap tinggi (>Rp200 ribu), juga yang tinggal jauh dari kampus (20-40 km) tetapi biayanya rendah (<Rp200 ribu)

Dari hasil diagram ini kita sudah bisa menduga bahwa hubungan antara jarak tempat tinggal ke kampus dengan biaya yang dikeluarkan tidak terlalu erat dan arahnya tidak beraturan.

Akan tetapi, untuk lebih jelas, kita perlu meninjaunya lewat angka koefisien korelasi.

7.5 Analisis Korelasi Spearman's ρ

Setidaknya ada dua kondisi yang menganjurkan kita menganalisis korelasi suatu pasangan variabel metrik dengan koefisien ρ Spearman:

- Koefisien ρ Spearman biasanya digunakan untuk menganalisis korelasi dua variabel peringkat (*rank*). Dengan kata lain, koefisien ini lebih cocok dikenakan pada variabel-variabel dengan tingkat pengukuran **interval**, seperti peringkat, *rating* atau data lain yang tidak memiliki titik nol absolut yang bermakna.
- Kita **tidak menemukan adanya hubungan linear** antara dua variabel metrik yang kita analisis. Jika menurut pola data kita ditemukan hubungan linear, kita dianjurkan menggunakan koefisien ρ Spearman ini.

Kita akan menggunakan koefisien korelasi ρ Spearman ini untuk menganalisis hubungan antara jarak tempat tinggal dengan biaya yang dikeluarkan per pekan.

Koefisien korelasi untuk variabel metrik di R dapat dianalisis dengan perintah `cor()` yang mengambil masukan berupa vektor data angka variabel-variabel yang kita analisis. Adapun jenis korelasi dapat kita pilih dengan menambahkan argumen `method` = yang dapat bernilai `"spearman"`, `"kendall"`, atau `"pearson"`, sesuai dengan metode yang kita gunakan.

```
# Mengatur variabel x dan y
x <- data_mahasiswa$jarak
y <- data_mahasiswa$`biaya sepekan`

cor(x, y, method = "spearman")
```

```
## [1] 0.08576446
```

Sebagaimana koefisien-koefisien korelasi lainnya, nilai ρ Spearman berkisar antara -1 hingga $+1$ yang menyatakan hubungan berlawanan yang kuat hingga hubungan searah yang kuat. Secara umum, tanda positif pada koefisien tersebut menunjukkan hubungan yang searah antara biaya transportasi sepekan dengan jarak tempuh ke kampus. Akan tetapi, dilihat dari besar nilainya, sulit mengatakan bahwa terdapat hubungan yang kuat antara jarak tempuh ke kampus dengan biaya perjalanan sepekan.

7.6 Analisis Korelasi Pearson's r

Untuk analisis menggunakan koefisien korelasi Pearson's r , kita akan memodifikasi sedikit data kita. Kita akan melihat hubungan antara jumlah perjalanan di hari kerja (*weekdays*) dengan jarak tempat tinggal ke kampus **untuk pengguna transportasi online saja**. Dengan demikian, kita perlu membuat dataset terpisah dari dataset utama kita.

Terlebih dahulu, kita perlu membuat variabel khusus Jumlah Perjalanan Weekdays yang merupakan penjumlahan dari kolom-kolom Jumlah Perjalanan Senin hingga Jumlah Perjalanan Jumat. Perhatikan cara pembuatannya yang menggunakan perintah `rowSums()` dan `across()` yang merupakan perintah khusus untuk operasi-operasi antarkolom. Tanda : bermakna "pilih kolom dari jumlah perjalanan hari senin sampai kolom jumlah perjalanan hari jumat". Hal ini memungkinkan karena dalam dataset kita kolom-kolom tersebut posisinya berdekatan.

```
# Membuat kolom jumlah perjalanan weekdays
data_mahasiswa <- data_mahasiswa |>
  mutate(`Jumlah Perjalanan Weekdays` = rowSums(
    across(`Jumlah Perjalanan Senin`:`Jumlah Perjalanan Jumat`)
  ))
```

Setelah itu, barulah kita membuat dataset khusus pengguna layanan online saja. Kita menggunakan perintah `filter()` dengan operator `==` yang bermakna *saringlah data dengan nilai `kendaraan utama` sama dengan "Layanan online"*.

```
# Memilih responden mahasiswa pengguna angkutan daring saja
# dan membuatnya menjadi dataset baru
data_mahasiswa_online <- data_mahasiswa |>
  filter(`kendaraan utama` == "Layanan online")
```

Kita dapat mengecek hasilnya dengan melakukan perintah `group_by()` dan `summarize()`. Hasilnya akan menampilkan kendaraan utama kita hanya bernilai "Layanan online"

```
# Menampilkan hasil filter
data_mahasiswa_online |>
  group_by(`kendaraan utama`) |>
  summarize("jumlah" = n())
```

```
## # A tibble: 1 x 2
##   `kendaraan utama` jumlah
##   <chr>             <int>
## 1 Layanan online      188
```

Kemudian, kita bisa menghitung koefisien korelasi Pearson's r -nya.

```
# Mengatur variabel x dan y
x <- data_mahasiswa_online$jarak
y <- data_mahasiswa_online$`Jumlah Perjalanan Weekdays`

cor(x, y, method = "pearson")
```

```
## [1] -0.2980376
```

Interpretasi hasil koefisien tersebut sama dengan interpretasi koefisien korelasi variabel ordinal, yakni tanda menunjukkan arah hubungan sementara besar angka menunjukkan kekuatan hubungan.

Latihan: Interpretasikan nilai korelasi r tersebut

Kita akan menelaah hubungan antara dua variabel tersebut berdasarkan tampilan visualnya dari diagram pencar (*scatterplot*).

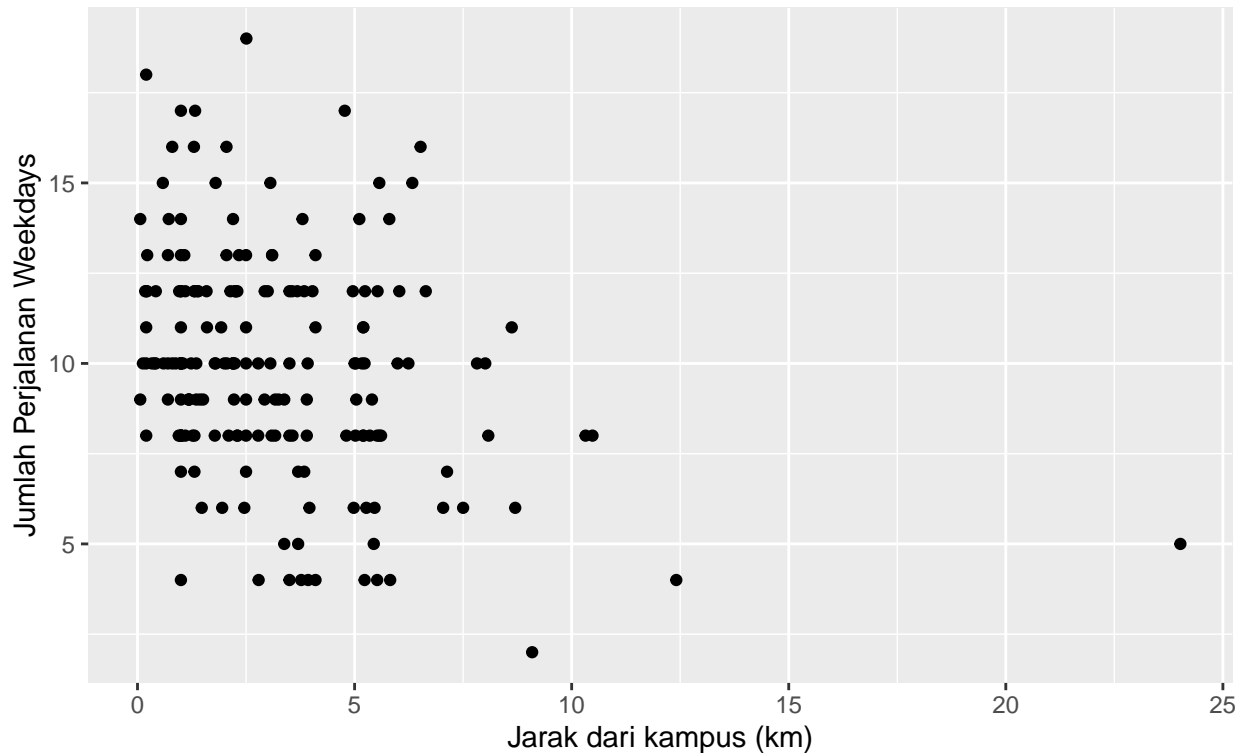
```
# Membuat diagram pencar antara variabel jumlah perjalanan weekdays dan jarak tempat tinggal pengguna o
scatter_plot <- ggplot(data = data_mahasiswa_online,
  mapping = aes(x = jarak, # variabel di sumbu X
    y = `Jumlah Perjalanan Weekdays`)) + # variabel di sumbu Y

geom_point() + # perintah untuk menampilkan diagram pencar
labs(title = "Jumlah Perjalanan Weekdays vs Jarak Tempat Tinggal dari Kampus",
  subtitle = "Pengguna layanan daring",
  y = "Jumlah Perjalanan Weekdays",
  x = "Jarak dari kampus (km)")

# Menampilkan scatter plot
scatter_plot
```

Jumlah Perjalanan Weekdays vs Jarak Tempat Tinggal dari Kampus

Pengguna layanan daring



Perhatikan bentuk sebaran titik-titik dalam diagram pencarnya. Secara sekilas, sebaran titik-titik tersebut membentuk formasi kemiringan ke bawah, akan tetapi kerapatannya kecil.

Latihan: Menggunakan dataset `data_mahasiswa` yang sudah ditambahkan jumlah perjalanan weekdays, analisislah hubungan antara jumlah perjalanan *weekdays* dengan biaya perjalanan per pekan. Tampilkan juga *scatterplot*-nya.

8 Modul-8: Analisis Hubungan Kausalitas: Regresi Linear Sederhana dan Berganda

Setelah mempelajari modul ini, Anda diharapkan dapat:

1. mampu menghasilkan regresi linear sederhana dengan perangkat lunak komputer **STP-13.3**
2. mampu menghasilkan persamaan regresi linear berganda dengan perangkat lunak komputer **STP-14.3**

8.1 Pendahuluan

Analisis regresi linear adalah analisis statistik untuk menyatakan hubungan sebab-akibat (kausalitas) antar minimal dua variabel. Analisis regresi linear yang melibatkan *dua variabel saja* (*satu variabel dependen* dan *satu variabel independen*), kita sebut sebagai **regresi linear sederhana**, sedangkan analisis regresi linar yang melibatkan lebih dari dua variabel (*satu variabel dependen* dan *lebih dari satu variabel independen*) kita sebut sebagai **regresi linear berganda** (*multiple linear regression*)

Analisis regresi linear mewajibkan tingkat pengukuran minimal untuk variabel dependennya adalah **metrik**. Sementara itu, variabel independennya dapat berupa metrik atau bukan (ordinal/nominal).

Analisis regresi linear menghasilkan persamaan yang disebut **persamaan regresi linear**. Bentuk umum persamaan regresi linear tersebut adalah sebagai berikut.

$$y = \beta_0 + \beta_1 x$$

dengan

- y adalah variabel dependen (disebut juga *variabel respons* atau *predicted variable*)
- x adalah variabel independen (disebut juga *variabel eksplanatory* atau *predictor variable*)
- β_0 adalah konstanta yang menjadi *intercept*, yaitu nilai y ketika $x = 0$
- β_1 adalah koefisien yang menyatakan seberapa besar perubahan y ketika satu unit nilai x berubah.

Persamaan regresi linear di atas adalah untuk **analisis regresi linear sederhana**, yakni analisis regresi linear yang melibatkan dua variabel saja. Untuk **regresi linear berganda**, kita hanya perlu menambahkan pasangan β dan x lainnya, sehingga bentuk umum untuk persamaan regresi linear berganda adalah seperti berikut.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$$

dengan

- x_k adalah variabel independen ke- k
- β_k adalah koefisien untuk variabel independen ke- k tersebut

8.2 Penjelasan Kasus

Dalam praktikum kali ini, kita akan memodelkan hubungan kausal antara dua variabel metrik: **jarak tempat tinggal ke kampus** dan **biaya perjalanan sepekan** untuk mahasiswa-mahasiswi yang menggunakan **kendaraan layanan online saja**. Harga perjalanan menggunakan layanan *online* tentunya dipengaruhi oleh jarak tempuh kendaraan tersebut. Hubungan kausal ini menjadi kasus untuk **regresi linear sederhana**.

Untuk **regresi linear berganda**, kita akan melibatkan satu variabel tambahan sebagai variabel independen, yakni variabel berjenis kategoris, sehingga kita memiliki total variabel sejumlah 3 buah (1 variabel dependen, 2 variabel independen).

8.3 Memuat Pustaka (*Libraries*) yang Diperlukan

Seperti biasa, kita perlu memuat pustaka (*libraries*) yang diperlukan dalam pengolahan data kita. Seperti halnya juga analisis korelasi variabel metrik, kita tidak lagi menggunakan tabel silang, tetapi kita langsung menganalisis kolom-kolom yang ada di dataset kita.

```
library(tidyverse) # untuk mengolah data terstruktur
library(readr)     # untuk membaca file csv
library(gtsummary) # untuk memproduksi tabel hasil pembuatan regresi dengan cepat dan rapi
```

8.4 Memuat Dataset

Kita akan menggunakan dataset keempat kampus di Kota Bandar Lampung dan sekitarnya sebagai bahan. Kemudian kita akan menyaring objek-objek yang memiliki nilai variabel kendaraan utama sama dengan 'Layanan online'.

```
# Membaca data dari file csv
data_mahasiswa <- read_csv2("datasets/Data Praktikum 08.csv")
```

```
## i Using '"','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
## Rows: 1557 Columns: 22
```

```
## -- Column specification -----
```

```
## Delimiter: ";"
```

```
## chr (8): Kampus_PT, Jenis Kelamin, Fakultas, Prodi, Tingkat Semester, Uang Saku, kendaraan utama, j
```

```
## dbl (14): Nomor urut, Umur, jumlah mobil, jumlah motor, jumlah sepeda, jarak, biaya sepekan, Jumlah l
```

```
##
```

```
## 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.
```

```
# Memisahkan data mahasiswa yang menggunakan kendaraan online sebagai moda utama
data_mahasiswa_online <- data_mahasiswa |>
  filter(`kendaraan utama` == 'Layanan online')
```

8.5 Pola Hubungan Data

Sebagai pendahuluan, kita akan mengidentifikasi kekuatan, arah, dan pola hubungan antara variabel dependen dan independen kita (biaya perjalanan sepekan dan jarak tempat tinggal-kampus). Untuk itu kita dapat menghitung koefisien korelasi untuk hubungan variabel metrik juga, yakni koefisien korelasi Pearson's r .

Pola hubungan akan kita analisis dengan membuat diagram pencar (*scatter plot*) antara variabel dependen dengan variabel independennya.

Menghitung Koefisien Pearson's r

Menghitung koefisien korelasi Pearson's r dapat kita lakukan dengan perintah `cor()` dengan atribut `method = "pearson"` setelah terlebih dahulu menyimpan nilai masing-masing variabel ke dalam suatu vektor.

```
# Menyimpan vektor jarak dan biaya (perjalanan) sepekan ke dalam variabel
# 'x' dan 'y'
x <- data_mahasiswa_online$jarak
y <- data_mahasiswa_online$`biaya sepekan`

cor(x, y, use = "complete.obs", method = "pearson")
```

```
## [1] -0.1813941
```

Jawablah: Tuliskan interpretasi kalian terhadap koefisien korelasi antara variabel jarak tempuh dengan biaya perjalanan sepekan.

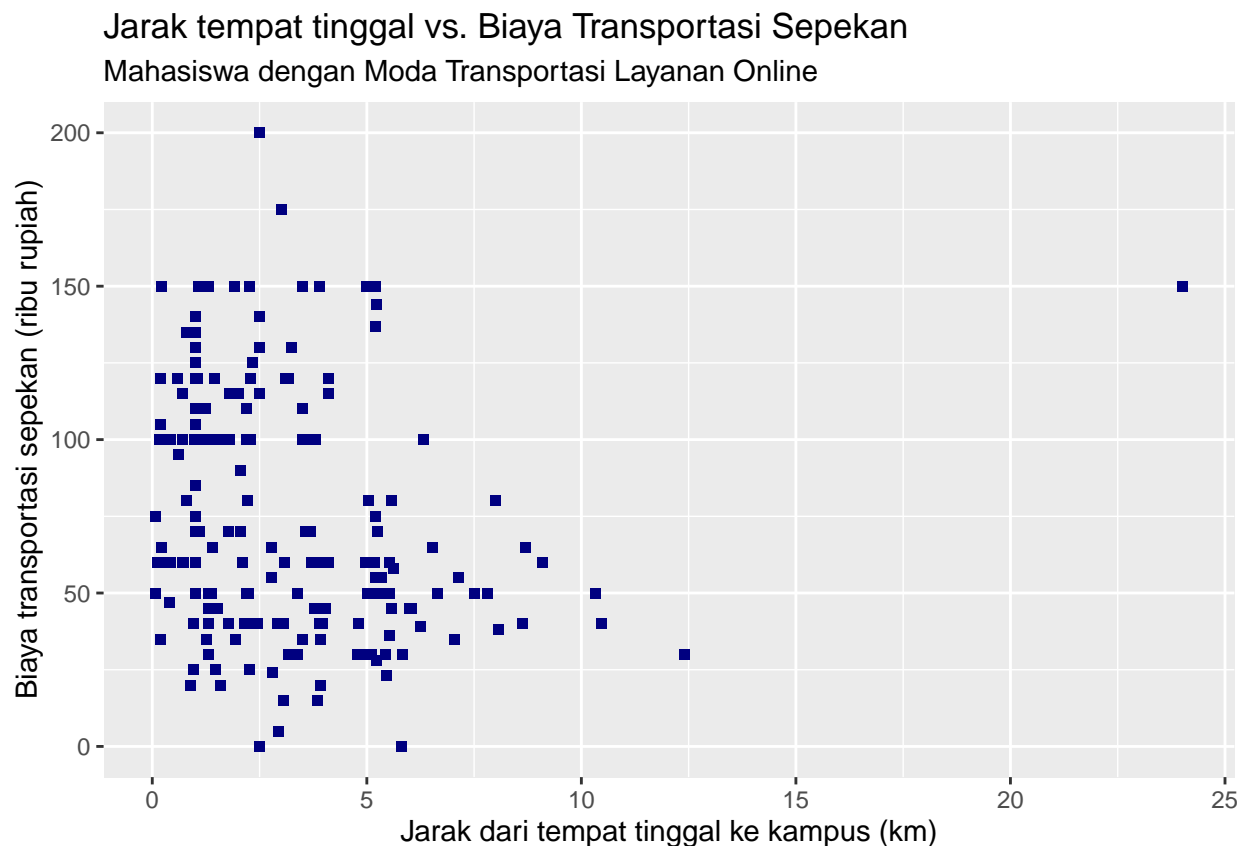
Petunjuk: ulas kekuatan serta arah hubungannya serta maknai secara kontekstual ulasan tersebut

Membuat Diagram Pencar

Membuat diagram pencar dapat dilakukan dengan menerapkan perintah `geom_point()` dari pustaka `ggplot2` yang dimuat bersama pustaka `tidyverse`.

```
# Membuat diagram pencar antara variabel jarak dan biaya transportasi sepekan
# untuk mahasiswa yang pakai layanan online
scp <- ggplot(data = data_mahasiswa_online,
              mapping = aes(x = jarak, # variabel di sumbu X
                           y = `biaya sepekan`
                           ) + # variabel di sumbu Y
              geom_point(color = 'navy', # perintah untuk menampilkan diagram pencar dengan warna biru
                          size = 1.5,    # mengatur ukuran titik
                          shape = 15) +  # mengatur bentuk titik menjadi persegi
              labs(title = "Jarak tempat tinggal vs. Biaya Transportasi Sepekan",
                   subtitle = "Mahasiswa dengan Moda Transportasi Layanan Online",
                   y = "Biaya transportasi sepekan (ribu rupiah)",
                   x = "Jarak dari tempat tinggal ke kampus (km)"))

# Menampilkan diagram
scp
```



Setelah Anda melakukan pembuatan diagram pencar tersebut, Anda akan menyadari bahwa kita memiliki *outlier* (pencilan), yakni objek yang memiliki nilai jarak dari tempat tinggal ke kampus mencapai hampir 25 km.

Keberadaan pencilan ini akan mengganggu hasil analisis kita. Mari kita buktikan dengan menghilangkan data pencilan tersebut.

```
# Menghilangkan observasi pencilan dengan menyaring observasi dengan
# jarak < 15 km
data_mahasiswa_online <- data_mahasiswa_online |>
  filter(jarak < 15)
```

Di sini kita akan menghitung ulang nilai koefisien korelasi kita.

```
# Menghitung korelasi antara biaya sepekan dengan jarak dari dataset yang
# sudah dihilangkan pencilannya
x <- data_mahasiswa_online$jarak
y <- data_mahasiswa_online$`biaya sepekan`

cor(x, y, method = "pearson")
```

```
## [1] -0.3086803
```

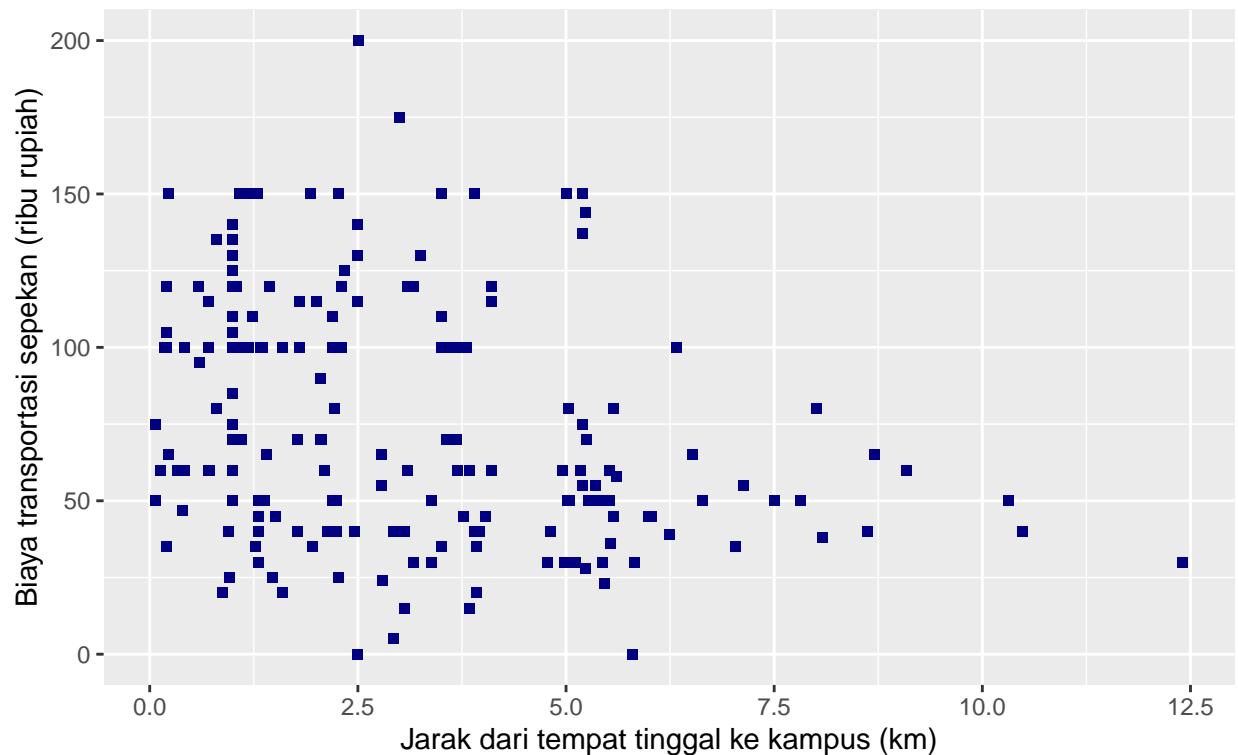
Jawablah: Bagaimana perbedaan nilai koefisiennya?

Kita pun dapat memeriksa hasil perbaikan dataset kita dari diagram pencar yang baru berikut.

```
# Membuat diagram pencar antara variabel jarak dan biaya transportasi sepekan
# untuk mahasiswa yang pakai layanan online, setelah pencilan dihilangkan
scp <- ggplot(data = data_mahasiswa_online,
  mapping = aes(x = jarak, # variabel di sumbu X
                y = `biaya sepekan`)) + # variabel di sumbu Y
  geom_point(color = 'navy', # perintah untuk menampilkan diagram pencar
    size = 1.5, # mengatur ukuran titik
    shape = 15) + # mengatur bentuk titik menjadi persegi
  labs(title = "Jarak tempat tinggal vs. Biaya Transportasi Sepekan",
    subtitle = "Mahasiswa dengan Moda Transportasi Layanan Online",
    y = "Biaya transportasi sepekan (ribu rupiah)",
    x = "Jarak dari tempat tinggal ke kampus (km)")

# Menampilkan diagram yang baru
scp
```

Jarak tempat tinggal vs. Biaya Transportasi Sepekan Mahasiswa dengan Moda Transportasi Layanan Online



8.6 Model Regresi Linear Sederhana

Dalam bagian ini kita akan mempelajari cara menyusun, menampilkan, menafsirkan/menginterpretasi, dan memprediksi nilai variabel dependen berdasarkan model yang kita susun.

8.6.1 Penyusunan Model

Untuk membuat persamaan model regresi linear, kita akan menggunakan perintah `lm` yang sudah disediakan secara bawaan oleh R. Hasil dari perintah ini dapat kita simpan sebagai variabel tertentu.

```
# Membuat model regresi linear
model <- lm(y ~ x, data = data_mahasiswa_online)
```

Argumen yang kita gunakan adalah `var_dependen ~ var_independen` serta nama dataset yang kita gunakan. Tanda `~` (disebut *tilde*) dapat kita masukkan dengan menekan **Shift** + ``` yang ada di sebelah kiri tombol angka 1 di *keyboard* kita.

Sebelumnya kita sudah mendefinisikan `y` dan `x` sebagai variabel untuk vektor biaya transportasi sepekan dan vektor jarak, sehingga kita bisa langsung menggunakannya di atas. Dataset yang kita gunakan adalah dataset mahasiswa yang menggunakan layanan *online* sebagai moda transportasi utama yang kita simpan dalam `data_mahasiswa_online`.

Characteristic	Beta	95% CI	p-value
x	-5.3	-7.7, -2.9	<0.001

Abbreviation: CI = Confidence Interval

8.6.2 Penampilan Model

Untuk menampilkan hasil penyusunan model regresi linear, kita akan menggunakan perintah `summary` yang juga sudah disediakan secara bawaan oleh R.

```
# Menampilkan hasil model regresi linear
summary(model)

##
## Call:
## lm(formula = y ~ x, data = data_mahasiswa_online)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -79.255 -30.717  -3.562   26.758  120.787
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   92.557      4.598   20.129 < 2e-16 ***
## x             -5.321      1.205   -4.414 1.72e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.11 on 185 degrees of freedom
## Multiple R-squared:  0.09528,    Adjusted R-squared:  0.09039
## F-statistic: 19.48 on 1 and 185 DF,  p-value: 1.721e-05
```

Yang ditampilkan dengan perintah tersebut antara lain adalah sebagai berikut. Penjelasan lebih lanjutnya akan diberikan di subbagian selanjutnya.

- **Call** : bentuk persamaan model yang sudah kita input sebelumnya
- **Residuals** : informasi residual model
- **Coefficients** : tabel yang menunjukkan angka-angka dalam persamaan model yang dihasilkan
- Nilai-nilai uji kualitas model seperti **Residual standard error** (*standard error* estimasi), **Multiple R-squared** dan **Adjusted R-squared** (koefisien determinasi), serta **F-statistic** (uji signifikansi model).

Selain menggunakan `summary`, kita juga dapat menggunakan perintah `tbl_regression`, perintah yang disediakan oleh paket `library gtsummary`. Hasil dari perintah ini adalah tabel yang sudah diformat sesuai dengan *templat* artikel ilmiah.

```
# Menampilkan hasil model regresi linear dengan tbl_regression
tbl_regression(model)
```

Characteristic	Beta	99% CI	p-value
(Intercept)	93	81, 105	<0.001
Jarak tempuh ke kampus, km	-5.3	-8.5, -2.2	<0.001

Abbreviation: CI = Confidence Interval

Jika kalian menemukan pertanyaan untuk menginstal sebuah paket di *console*, yakni `broom.helpers`, pilih saja “Yes” untuk mengunduh dan menginstalnya.

Secara bawaan, tabel tidak menampilkan nilai konstanta (*intercept*), menampilkan nilai *confidence level* 95%, dan menampilkan nilai dari variabel independen sesuai variabel yang kita tetapkan sebelumnya, dalam hal ini adalah *x*. Untuk menggantinya, kita dapat melakukan penyesuaian perintah sebagai berikut.

```
tbl_regression(
  model,
  intercept = TRUE, # mengatur intercept-nya ditampilkan
  conf.level = 0.99, # mengatur nilai confidence level menjadi 99%
  label = list(x ~ "Jarak tempuh ke kampus, km") # Mengganti tulisan "x" saja menjadi yang
) # lebih bermakna
```

8.6.3 Penafsiran/interpretasi Model

Yang kita akan tafsirkan dari model kita di antaranya adalah:

- persamaan model
- makna nilai konstanta dan koefisien
- uji kualitas model (ANOVA, F-statistic)
- uji kualitas variabel ($\Pr(>|t|)$)

8.6.3.1 Tafsiran persamaan model Persamaan model dapat kita tafsirkan berdasarkan keluaran dari `summary` berupa *Coefficients* atau hasil dari `tbl_regression` milik `gtsummary`.

Berdasarkan keluaran kedua perintah tersebut, kita dapat mengetahui bagaimana persamaan regresi linear dengan mengganti bentuk umum persamaan regresi linear dengan angka-angka yang dihasilkan di `model`.

Bentuk umum:

$$y = \beta_0 + \beta_1 x$$

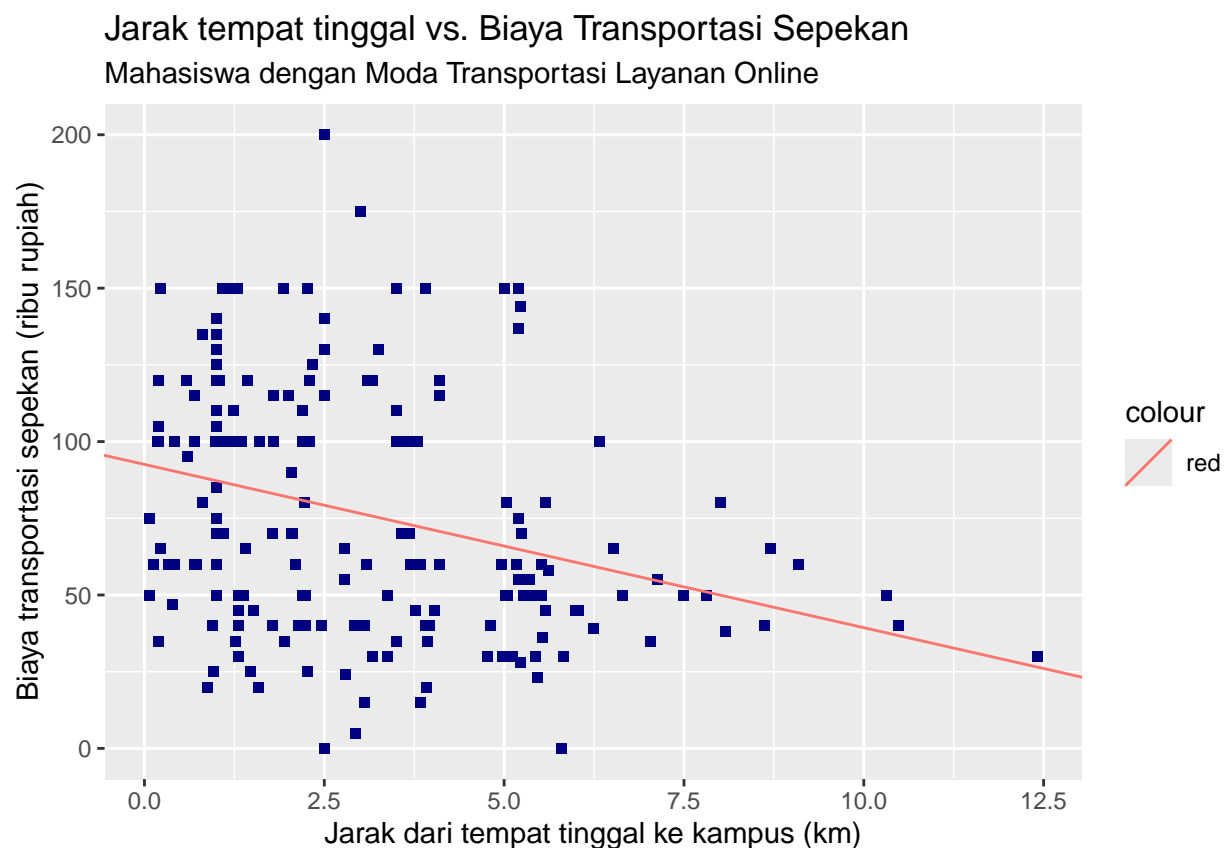
Dengan mengambil hasil dari `summary(model)`, kita dapat menuliskan persamaan regresi linear kita menjadi:

$$y = 93 - 5,3x$$

Dari persamaan model ini kita dapat menggambar garis yang melewati titik-titik data kita. Ini merupakan persamaan garis terbaik dari seluruh kemungkinan persamaan garis yang ada, yang kita tentukan dengan meminimalkan kuadrat terkecil.

```
# Membuat diagram pencar antara variabel jarak dan biaya transportasi sepekan
# untuk mahasiswa yang pakai layanan online, setelah pencilan dihilangkan,
# ditambah garis dari persamaan regresi
scp <- ggplot(data = data_mahasiswa_online,
              mapping = aes(x = jarak, # variabel di sumbu X
                           y = `biaya sepekan`) + # variabel di sumbu Y
              geom_point(color = 'navy',
                         size = 1.5,
                         shape = 15) +
              geom_abline(aes(intercept = coef(model)[1],
                              slope = coef(model)[2],
                              color = "red")) +
              labs(title = "Jarak tempat tinggal vs. Biaya Transportasi Sepekan",
                   subtitle = "Mahasiswa dengan Moda Transportasi Layanan Online",
                   y = "Biaya transportasi sepekan (ribu rupiah)",
                   x = "Jarak dari tempat tinggal ke kampus (km)"))

# Menampilkan diagram pencar
scp
```



- `coef(model)[1]` mengacu pada keluaran fungsi `coef` dari variabel `model` kita yang pertama (`[1]`), yakni angka *intercept*/konstantanya. Ini menjadi nilai dari atribut `intercept` yang terletak dalam fungsi `geom_abline`
- `coef(model)[2]` mengacu pada keluaran fungsi `coef` dari variabel `model` kita yang kedua (`[2]`), yakni angka koefisien variabel independen kita. Ini menjadi nilai dari atribut `slope` yang terletak dalam

fungsi `geom_abline` karena nilai koefisien variabel independen menunjukkan kemiringan garis persamaan regresi linear kita

Persamaan model dapat kita tafsirkan berdasarkan keluaran dari `summary` berupa **Coefficients** atau hasil dari `tbl_regression` milik `gtsummary`.

Berdasarkan keluaran kedua perintah tersebut, kita dapat mengetahui bagaimana persamaan regresi linearnya dengan mengganti bentuk umum persamaan regresi linear dengan angka-angka yang dihasilkan di `model`.

Bentuk umum:

$$y = \beta_0 + \beta_1 x$$

Dengan mengambil hasil dari `summary(model)`, kita dapat menuliskan persamaan regresi linear kita menjadi:

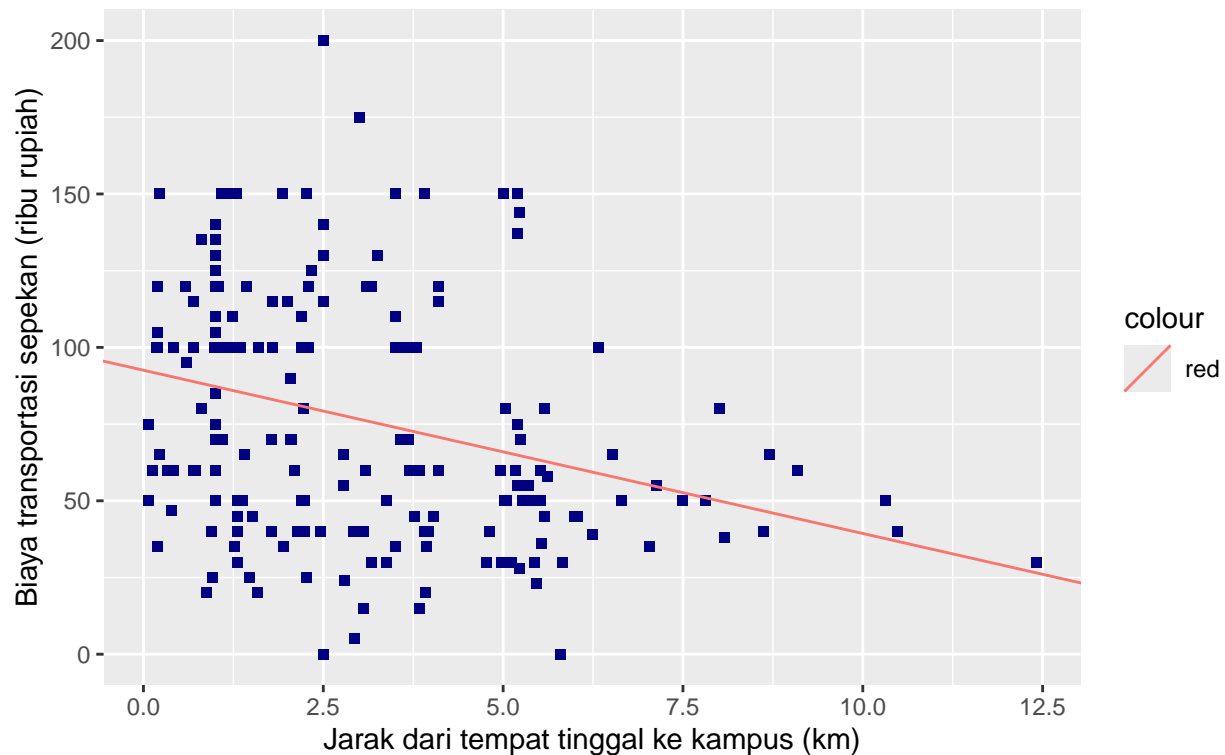
$$y = 93 - 5,3x$$

Dari persamaan model ini kita dapat menggambar garis yang melewati titik-titik data kita. Ini merupakan persamaan garis terbaik dari seluruh kemungkinan persamaan garis yang ada, yang kita tentukan dengan meminimalkan kuadrat terkecil.

```
# Membuat diagram pencar antara variabel jarak dan biaya transportasi sepekan
# untuk mahasiswa yang pakai layanan online, setelah pencilan dihilangkan,
# ditambah garis dari persamaan regresi
scp <- ggplot(data = data_mahasiswa_online,
              mapping = aes(x = jarak, # variabel di sumbu X
                           y = `biaya sepekan`)) + # variabel di sumbu Y
  geom_point(color = 'navy',
             size = 1.5,
             shape = 15) +
  geom_abline(aes(intercept = coef(model)[1],
                  slope = coef(model)[2],
                  color = "red")) +
  labs(title = "Jarak tempat tinggal vs. Biaya Transportasi Sepekan",
       subtitle = "Mahasiswa dengan Moda Transportasi Layanan Online",
       y = "Biaya transportasi sepekan (ribu rupiah)",
       x = "Jarak dari tempat tinggal ke kampus (km)")

# Menampilkan diagram pencar
scp
```

Jarak tempat tinggal vs. Biaya Transportasi Sepekan Mahasiswa dengan Moda Transportasi Layanan Online



- `coef(model)[1]` mengacu pada keluaran fungsi `coef` dari variabel `model` kita yang pertama (`[1]`), yakni angka *intercept*/konstantanya. Ini menjadi nilai dari atribut `intercept` yang terletak dalam fungsi `geom_abline`
- `coef(model)[2]` mengacu pada keluaran fungsi `coef` dari variabel `model` kita yang kedua (`[2]`), yakni angka koefisien variabel independen kita. Ini menjadi nilai dari atribut `slope` yang terletak dalam fungsi `geom_abline` karena nilai koefisien variabel independen menunjukkan kemiringan garis persamaan regresi linear kita

8.6.3.2 Tafsiran nilai konstanta dan koefisien Persamaan regresi linear ini bermakna nilai konstanta sebesar 93 berarti bahwa dengan tinggal di kampus ($x = 0$), mahasiswa masih akan mengeluarkan biaya sebesar 93.000 rupiah (ingat bahwa variabel **biaya sepekan** menggunakan satuan ribuan rupiah) untuk mengakses kampus selama sepekan. Kita bisa pahami hal tersebut sebagai perkiraan harga sewa penginapan di kampus.

Sementara itu, nilai koefisien sebesar $-5,3$ berarti bahwa penambahan jarak sebesar 1 km akan **mengurangi** biaya transportasi sepekan sebesar 5.300 rupiah. Artinya, makin jauh 1 km seorang mahasiswa tinggal dari kampus, makin kecil pula biaya transportasi sepekananya sebesar 5.300 rupiah.

Pertanyaan bonus: Mengapa mahasiswa yang tinggal lebih jauh dari kampus biaya perjalanan sepekananya lebih kecil? Bukankah biaya transportasi makin jauh malah makin mahal?

8.6.3.3 Tafsiran uji kualitas model Uji kualitas model kita terdiri atas *standard error* dari residual, koefisien determinasi atau R^2 , dan uji signifikansi persamaan menggunakan statistik F (**F-statistic**) yang merupakan hasil ANOVA (*analysis of variance*).

- *Standard error* dari residual mencerminkan rata-rata galat (residu), yaitu selisih nilai variabel dependen yang dihitung (biaya perjalanan sepekan hasil prediksi dari model) dengan nilai variabel dependen yang kita peroleh dari data.
 - Hasil **summary** menunjukkan angka 38,11 yang artinya rata-rata kesalahan prediksi dari model kita adalah sebesar 38.110 rupiah.
- Koefisien determinasi atau R^2 menyatakan kesesuaian (*fit*) dari model kita, yakni seberapa banyak proporsi variansi data variabel dependen kita (biaya perjalanan sepekan) dijelaskan oleh variabel independennya (jarak dari kampus ke tempat tinggal).
Sisa dari angka R^2 ini adalah banyak variansi yang tidak dijelaskan oleh variabel independen, artinya dijelaskan oleh variabel lainnya yang tidak ada dalam model dan juga keacakan (misal dalam pengambilan data atau kejadian acak).
 - Hasil **summary** menunjukkan angka 0,095 yang berarti hanya 9,5% variansi biaya perjalanan sepekan dijelaskan oleh variansi jarak perjalanan ke kampus.
 - Makin tinggi nilai R^2 dari model kita, makin besar proporsi variansi variabel dependen kita dijelaskan oleh variabel independennya.
 - Hal ini menunjukkan bahwa pengaruh dari variabel independen kita terhadap variabel dependennya kuat
- Koefisien uji signifikansi (F) (**F-statistics**) adalah uji hipotesis ANOVA (*analysis of variance*) yang membuktikan hipotesis nol bahwa tidak ada hubungan antara variabel dependen dengan variabel independen yang kita modelkan. Penerimaan hipotesis nol ini ditandai dari nilai signifikansi (**p-value**) yang lebih kecil dari 5% (0.05).
 - Hasil **summary** menunjukkan bahwa nilai **p-value** kita adalah 1,721e-05. Angka tersebut menyatakan notasi ilmiah $1,721 \times 10^{-5}$ yang berarti bahwa nilai **p-value** kita sangat-sangat lebih kecil dari 0.05.
 - Ini artinya kita tidak dapat menerima hipotesis nol kita yang menyatakan tidak ada hubungan antara variabel biaya perjalanan sepekan (variabel dependen) dengan variabel jarak tempuh ke kampus (variabel independen).

Simpulan tafsiran kualitas model: kita sudah melihat bahwa hubungan kausal antara variabel jarak ke kampus terhadap biaya perjalanan sepekan adalah terbalik dengan besar kemiringan 5,3. Hubungan kausal tersebut benar-benar ada berdasarkan hasil dari koefisien uji signifikansi (ANOVA) yang *p-value*-nya <0,05. Akan tetapi, hubungan tersebut tidak cukup kuat karena nilai koefisien determinasi yang kecil, yakni hanya 9% saja.

8.6.3.4 Tafsiran uji kualitas variabel Uji kualitas variabel adalah pengujian terhadap signifikansi atau kepentingan variabel independen kita. Hasil pengujian ini diperlihatkan oleh angka *p-value* dari uji t (**t value**)-nya, yakni yang ada di kolom **Pr(>|t|)**.

Apabila nilai **Pr(>|t|)** kita >0,05 maka variabel independen kita dianggap **tidak signifikan**, karena artinya nilai koefisien yang ada di kolom **Estimate** **sebenarnya adalah nol** sehingga **bisa dikeluarkan dari persamaan regresi linear**.

Sebaliknya, jika nilai **Pr(>|t|)** kita <0,05 maka variabel independen kita dianggap **signifikan**, artinya nilai koefisien yang ada di kolom **Estimate** **sebenarnya adalah nilai koefisien tersebut**.

Nilai **Pr(>|t|)** variabel x adalah 1,721e-05 yang berarti koefisien senilai -5.321 adalah **signifikan**.

8.6.4 Melakukan Prediksi

Kita melakukan prediksi dengan menggunakan perintah `predict`. Misalkan kita ingin melakukan prediksi biaya perjalanan sepekan untuk mahasiswa yang tinggal di jarak 90 dan 120 km dari kampus

```
bahan_pred <- tibble(x = c(90, 120))  
  
predict(model, bahan_pred)
```

```
##           1           2  
## -386.3117 -545.9345
```

- Perintah `tibble` adalah perintah untuk membuat dataset berisi kolom `x` yang terdiri atas 2 nilai: 90 km dan 120 km. Kita menyimpan dataset contoh tersebut ke dalam variabel bernama `bahan_pred`
- Dalam membuat dataset untuk prediksi kita harus memperhatikan nama variabel/nama kolom yang kita berikan dalam perintah `tibble`. Nama kolom yang akan kita prediksi harus **sama persis** dengan nama variabel yang kita nyatakan dalam perintah `lm`.

8.7 Model Regresi Linear Berganda

Dalam bagian ini kita hanya akan mempelajari penggunaan variabel *dummy* untuk meningkatkan kekuatan prediksi kita. Di akhir, Anda akan menafsirkan hal-hal yang sudah Anda pelajari pada bagian regresi linear.

Kita akan menggunakan variabel jenis tempat tinggal untuk variabel *dummy*. Pertama, kita akan mengatur jenis nilai pada kolom jenis tempat tinggal menggunakan perintah `factor`.

```
# Membuat nilai kolom menjadi variabel kategoris dengan 'factor'  
data_mahasiswa_online$`jenis tempat tinggal` <- factor(data_mahasiswa_online$`jenis tempat tinggal`)
```

Kedua, kita akan membuat model seperti halnya yang kita lakukan dalam analisis regresi linear sederhana. Kita akan menggunakan nama yang lebih representatif.

```
# Membuat variabel-variabel  
jarak <- x # variabel 'x' saja diubah menjadi 'jarak'  
biaya <- y # variabel 'y' saja diubah menjadi 'biaya'  
ling <- data_mahasiswa_online$`jenis tempat tinggal` # tempat tinggal diberi  
# nama 'ling'  
  
# Menyatakan model  
mdl <- lm(biaya ~ jarak + ling, data = data_mahasiswa_online)  
  
# Melihat hasil pemodelan  
summary(mdl)
```

```
##  
## Call:  
## lm(formula = biaya ~ jarak + ling, data = data_mahasiswa_online)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -73.743 -23.421  -2.143   19.424 114.322
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      98.103      8.104  12.105 < 2e-16 ***
## jarak           -4.954      1.118  -4.431 1.63e-05 ***
## lingKos sendiri  -20.779      8.787  -2.365  0.0191 *
## lingRumah mengontrak bersama -17.198     10.409  -1.652  0.1002
## lingRumah mengontrak pribadi  27.744     16.031   1.731  0.0852 .
## lingRumah pribadi   20.348      9.866   2.062  0.0406 *
## lingRumah saudara   10.365     12.689   0.817  0.4151
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.23 on 180 degrees of freedom
## Multiple R-squared:  0.2899, Adjusted R-squared:  0.2663
## F-statistic: 12.25 on 6 and 180 DF, p-value: 1.538e-11
```

Selanjutnya, kita akan menafsirkan hasil dari summary model ini.

8.7.1 Penjelasan Variabel *Dummy*

Dalam persamaan model kita tersebut, variabel `ling` atau `jenis tempat tinggal` diubah menjadi variabel *dummy* :

1. `lingKos sendiri`,
2. `lingRumah mengontrak bersama`,
3. `lingRumah mengontrak pribadi`,
4. `lingRumah pribadi`, dan
5. `lingRumah saudara`.

Nama variabel-variabel tersebut tak lain adalah gabungan kata `ling` dengan tiap-tiap kategori dalam variabel `jenis tempat tinggal`.

Perhatikan bahwa kita jadi memiliki 5 variabel *dummy* dari satu variabel kategoris `jenis tempat tinggal` yang terdiri atas 6 kategori nilai: “Kos bersama-sama”, “Kos sendiri”, “Rumah mengontrak bersama”, “Rumah mengontrak pribadi”, “Rumah pribadi”, dan “Rumah saudara.”

Variabel *dummy* adalah variabel yang bernilai 0 (nol) atau 1 (satu) saja. Nol artinya variabel *dummy* tersebut bernilai “salah” atau “tidak”, sedangkan satu artinya variabel *dummy* tersebut bernilai “benar” atau “ya.”

Variabel-variabel *dummy* yang 5 buah ini hanya akan bernilai 1 atau 0 sesuai dengan kategori dari variabel `jenis tempat tinggal` suatu objek. Perhatikan bahwa kita tidak punya “`lingKos sendiri`” dalam variabel *dummy* kita. Ini artinya apabila kategori `jenis tempat tinggal` suatu objek adalah “Kos sendiri”, kelima variabel *dummy* akan bernilai 0.

Tabel berikut merangkum nilai-nilai variabel *dummy* untuk setiap kategori nilai variabel `jenis tempat tinggal`.

Table 7: Nilai variabel dummy untuk setiap kategori

Kategori	lingKos sendiri	lingRumah mengontrak bersama	lingRumah mengontrak pribadi	lingRumah pribadi	lingRumah saudara
Kos sendiri	1	0	0	0	0
Rumah mengontrak bersama	0	1	0	0	0
Rumah mengontrak pribadi	0	0	1	0	0
Rumah pribadi	0	0	0	1	0
Rumah saudara	0	0	0	0	1
Kos bersama	0	0	0	0	0

8.7.2 Interpretasi Variabel *Dummy*

Dalam menginterpretasi variabel *dummy* kita tidak bisa menggunakan $\Pr(>|t|)$ semata, karena variabel *dummy* pada hakikatnya sepaket. Jadi, walaupun nilai $\Pr(>|t|)$ sebuah variabel *dummy* adalah $>0,05$ ia akan tetap berada dalam persamaan regresi linear.

Interpretasi koefisien variabel *dummy* adalah dengan memahami bahwa nilai variabel dependen berubah sesuai nilai koefisien apabila variabel *dummy* tersebut bernilai **benar** atau **1**. Artinya, jika mahasiswa tinggal di kosan sendirian (`lingKos sendiri = 1`), maka biaya transportasi sepekan berkurang sebesar 20.779 yang juga berarti sebesar 20,7 ribu rupiah. Jika mahasiswa tinggal di rumah pribadi (`lingRumah pribadi = 1`) maka biaya transportasi sepekan bertambah sebesar 20.348 atau 20,3 ribu rupiah.

8.7.3 Prediksi Model

Dari persamaan model regresi linear kita, kita juga dapat memprediksi nilai biaya perjalanan sepekan untuk mahasiswa-mahasiswa dengan kondisi lain. Misalnya kita ingin memprediksi biaya sepekan transportasi mahasiswa yang tinggal di Rumah saudara berjarak 7,9 km dari kampus dan di Rumah kontrakan bersama yang berjarak 10 km dari kampus.

```
# Menyusun dataset untuk diprediksi
mhs <- tibble(jarak = c(7.9, 10),
              ling = c("Rumah saudara", "Rumah mengontrak bersama")
            )

# melihat hasil prediksi
predict mdl, mhs)
```

```
##          1          2
## 69.33008 31.36362
```

Jawablah:

- Bagaimana persamaan model regresi linear berganda kita?
- Tafsirlah kualitas model kita, mulai uji kualitas model (F), R^2 , serta uji kualitas setiap variabel
- Apa saja yang meningkatkan dan menurunkan biaya perjalanan sepekan mahasiswa?

9 Modul-9: Analisis Hubungan Multivariat Interdependensi - Analisis Faktor dan PCA

Setelah mempelajari modul ini, Anda diharapkan dapat menghasilkan komponen prinsip menggunakan perangkat lunak komputer **STP-14.3**

9.1 Pendahuluan

Analisis Komponen Prinsip (*Principal Component Analysis*, PCA) dan Analisis Faktor (*Common Factor Analysis*) adalah metode analisis multivariat yang digunakan untuk meringkas atau mereduksi jumlah variabel yang banyak menjadi beberapa dimensi baru (disebut komponen atau faktor) yang lebih sedikit, namun tetap merepresentasikan informasi dari variabel asli.

Kedua metode ini termasuk dalam analisis interdependensi, di mana seluruh variabel dianggap setara dan saling berhubungan satu sama lain, tanpa ada pembagian variabel independen dan dependen.

9.2 Studi Kasus

Kita akan menggunakan data Bindar [2022], yaitu penelitian mengenai preferensi masyarakat Kota Bandung dalam mengakses lokasi *Car-Free Day* (CFD). Terdapat 12 variabel yang akan dianalisis:

1. **ongkos**: Total biaya perjalanan
2. **bparkir**: Biaya parkir
3. **durasi**: Durasi perjalanan
4. **bareng**: Jumlah rombongan dalam perjalanan
5. **toplajur**: Jumlah lajur jalan terbanyak yang dilalui
6. **usia**: Usia pelaku perjalanan
7. **jmlmotor**: Jumlah sepeda motor di rumah tangga
8. **jmlmobil**: Jumlah mobil di rumah tangga
9. **jmlsepeda**: Jumlah sepeda di rumah tangga
10. **jmldewasa**: Jumlah orang dewasa dalam rumah tangga
11. **jmlanak**: Jumlah anak-anak dalam rumah tangga
12. **jarak**: Jarak tempuh dari rumah ke lokasi CFD

9.3 Memuat Pustaka (*Libraries*)

Kita membutuhkan paket **tidyverse** untuk manipulasi data dan **psych** untuk melakukan uji KMO, Bartlett, serta fungsi analisis faktor dan PCA.

```
library(tidyverse)
library(psych)
```

9.4 Asumsi Awal

Sebelum melakukan analisis komponen prinsip atau analisis faktor, idealnya kita perlu memeriksa asumsi dasar, yaitu:

1. **Linearitas**: Adanya hubungan linear antarvariabel.

2. Normalitas: Variabel berdistribusi normal multivariat.

Kita dapat melakukan inspeksi visual menggunakan matriks diagram pencar (*scatter plot matrix*). Di sini kita menggunakan fungsi `pairs.panels` dari paket `psych` yang memberikan informasi lengkap berupa scatter plot, histogram, dan nilai korelasi.

```
# Membaca data (Kita muat di sini untuk keperluan inspeksi awal)
# Jika data belum dimuat, baris ini akan memuatnya.
data_cfd <- read_csv2("datasets/Data Praktikum 09.csv")
```

```
## i Using "','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
## Rows: 319 Columns: 32
## -- Column specification -----
## Delimiter: ";"
## chr (2): seringnya, frek
## dbl (30): noresp, CFD, jk, usia, jmlmotor, jmlmobil, jmlsepeda, jmldevisa, jmlanak, kendaraan, kenda
##
## 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.
```

```
# Memilih 12 variabel metrik yang akan dianalisis sesuai studi kasus
# Variabel ini sama dengan yang akan digunakan pada tahap persiapan data selanjutnya
data_selected <- data_cfd |>
  select(ongkos, bparkir, durasi, bareng, toplajur, usia,
         jmlmotor, jmlmobil, jmlsepeda, jmldevisa, jmlanak, jarak)

# Membuat Scatter Plot Matrix menggunakan psych::pairs.panels
pairs.panels(data_selected,
  method = "pearson", # Menggunakan korelasi Pearson
  hist.col = "#00AFBB", # Warna histogram
  density = FALSE, # Tidak menampilkan garis density agar lebih bersih
  lm.col = "red", # Mengubah warna data points menjadi abu-abu gelap
  ellipses = TRUE, # Menampilkan elips korelasi untuk melihat pola linearitas
  lm = TRUE, # Menampilkan garis regresi linear lurus
  main = "Scatter Plot Matrix Asumsi Awal")
```

```
## Warning in plot.window(...): "lm.col" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter
```

```
## Warning in title(...): "lm.col" is not a graphical parameter
```

```
## Warning in plot.window(...): "lm.col" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter
```

```
## Warning in title(...): "lm.col" is not a graphical parameter
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter
```

```

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

```

```

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

```

```

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

```

```

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

```

```

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical par

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical par

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical par

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical par

## Warning in plot.window(...): "lm.col" is not a graphical parameter

```



```

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

```

```

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

```

```

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

```

```

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

```

```

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

```

```

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

```

```

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

```

```
## Warning in plot.window(...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter
## Warning in title(...): "lm.col" is not a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parame
## Warning in plot.window(...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter
## Warning in title(...): "lm.col" is not a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter
## Warning in plot.window(...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter
## Warning in title(...): "lm.col" is not a graphical parameter
## Warning in plot.window(...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter
## Warning in title(...): "lm.col" is not a graphical parameter
## Warning in plot.window(...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter
## Warning in title(...): "lm.col" is not a graphical parameter
## Warning in plot.window(...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter
## Warning in title(...): "lm.col" is not a graphical parameter
## Warning in plot.window(...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter
```



```

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, unf = unf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, unf = unf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, unf = unf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

```

```
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical par

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical par

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter
```

```

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical par

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical par

## Warning in plot.window(...): "lm.col" is not a graphical parameter

```

```
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int.abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter
```

```

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

```

```
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter
```

```

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

```

```

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

```



```

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

```



```

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "lm.col" is not a graphical parameter

## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

```

```
## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter
```

```
## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter

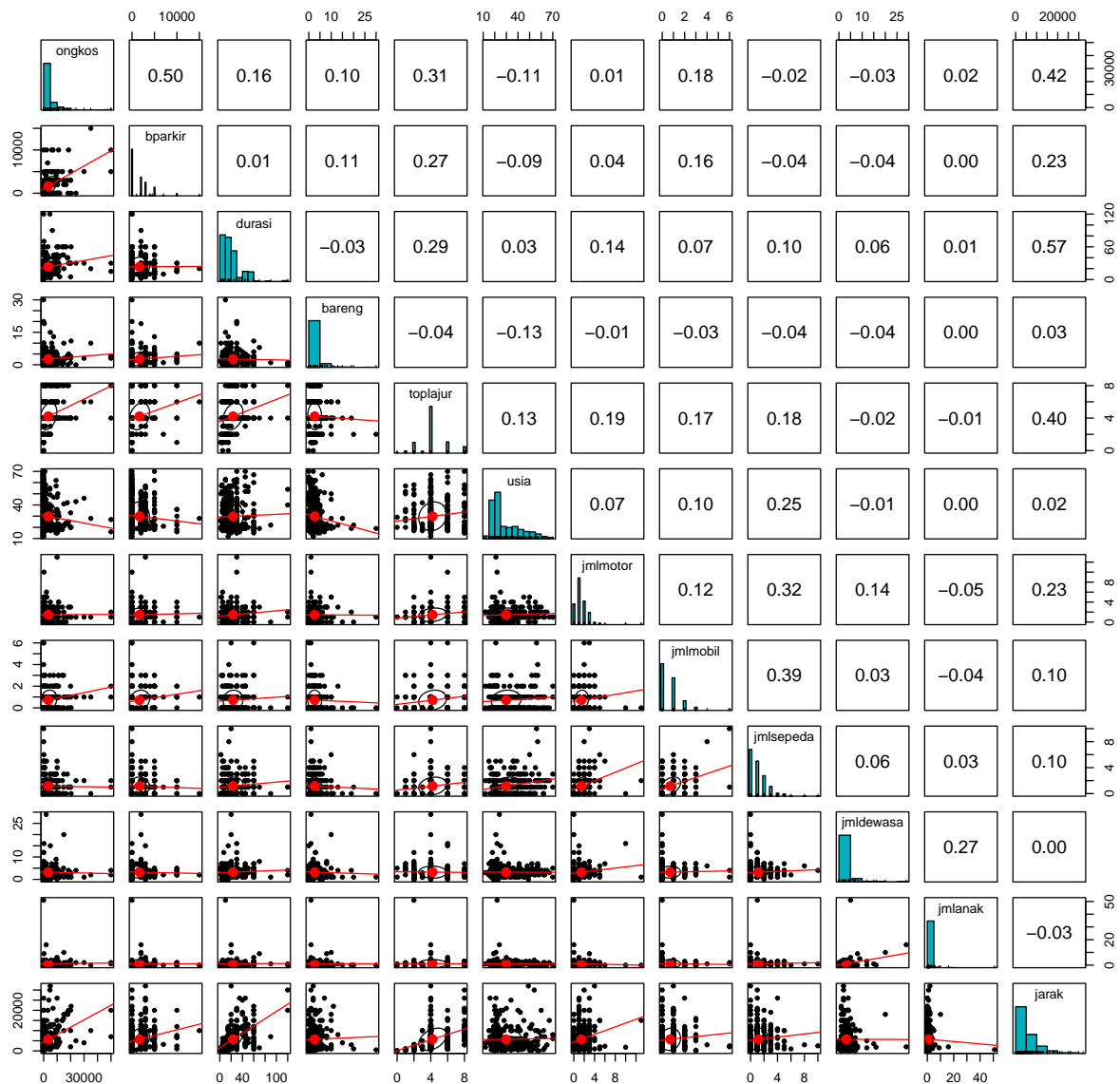
## Warning in axis(side = side, at = at, labels = labels, ...): "lm.col" is not a graphical parameter

## Warning in plot.window(...): "lm.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "lm.col" is not a graphical parameter

## Warning in title(...): "lm.col" is not a graphical parameter
```

Scatter Plot Matrix Asumsi Awal



Penjelasan Kode dan Output:

- **Kode:** Kita menggunakan fungsi `pairs.panels` dari paket `psych`. Argumen `method = "pearson"` digunakan untuk menghitung koefisien korelasi Pearson. `hist.col` mengatur warna histogram di diagonal utama. `ellipses = TRUE` menambahkan elips yang menggambarkan kovarians dan arah hubungan.
- **Output:**
 - **Diagonal Utama:** Menampilkan histogram distribusi frekuensi untuk setiap variabel. Kita dapat melihat apakah distribusi variabel mendekati normal (bentuk lonceng) atau menceng (*skewed*).
 - **Bawah Diagonal:** Menampilkan *scatter plot* (diagram pencar) untuk setiap pasangan variabel. Ini berguna untuk mendeteksi pola hubungan (apakah linear) dan adanya pencilan (*outliers*).
 - **Atas Diagonal:** Menampilkan nilai koefisien korelasi Pearson (r). Ukuran font angka korelasi menyesuaikan dengan besarnya nilai korelasi (semakin besar nilai, semakin besar font). **Interpretasi Singkat:**

- **Normalitas:** Pada diagonal utama, histogram yang berbentuk lonceng menunjukkan distribusi mendekati normal. Distribusi yang menceng (*skewed*) atau tidak simetris menandakan ketidaknormalan.
- **Linearitas:** Perhatikan bagian bawah diagonal (Scatter Plot).
 - * **Garis Tren:** Indikasi linearitas dapat dilihat dari garis tren (biasanya berwarna merah) yang terbentuk di antara titik-titik data. Jika garis tersebut cenderung lurus, maka hubungan antar variabel bersifat linear.
 - * **Elips:** Bentuk elips menunjukkan kekuatan korelasi. **Elips yang pipih (sempit)** menandakan korelasi kuat dan hubungan yang jelas. Sebaliknya, **elips yang cenderung bulat** menandakan korelasi yang lemah.
 - * Perbesar grafik dengan membuka hasil di jendela baru atau klik ‘Zoom’ jika ia muncul di panel ‘Plots’.

Latihan 2:

Berdasarkan *Scatter Plot Matrix* di atas:

1. Sebutkan satu variabel yang menurut Anda memiliki distribusi mendekati normal (lihat histogram diagonal)!
2. Sebutkan pasangan variabel yang memiliki hubungan linear cukup kuat (lihat bentuk elips/sebaran data)!

9.5 Mempersiapkan Data

Kita memastikan kembali variabel yang akan digunakan dalam analisis ini.

```
# Memilih variabel yang akan dianalisis
data_analisis <- data_cfd |>
  select(ongkos, bparkir, durasi, bareng, toplajur, usia,
         jmlmotor, jmlmobil, jmlsepeda, jmldevisa, jmlanak, jarak)

# Melihat sekilas data
glimpse(data_analisis)
```

```
## Rows: 319
## Columns: 12
## $ ongkos      <dbl> 464.40, 464.40, 464.40, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 10000.00, 30000.00
## $ bparkir     <dbl> 2000, 2000, 3000, 0, 0, 0, 0, 0, 0, 0, 2000, 5000, 0, 0, 0, 0, 0, 0, 3000, 0, 0
## $ durasi      <dbl> 20, 10, 15, 30, 5, 30, 5, 30, 5, 9, 30, 30, 10, 30, 2, 30, 60, 30, 30, 30, 20, 5, 4
## $ bareng      <dbl> 2, 0, 2, 5, 2, 0, 0, 0, 0, 0, 2, 3, 1, 0, 10, 3, 2, 3, 3, 2, 1, 1, 0, 1, 1, 1, 3, 0
## $ toplajur    <dbl> 4, 4, 4, 8, 2, 4, 4, 4, 0, 2, 8, 6, 4, 4, 2, 4, 6, 4, 4, 4, 4, 8, 4, 4, 4, 2, 4
## $ usia        <dbl> 45, 46, 50, 25, 22, 63, 22, 60, 20, 19, 16, 46, 22, 28, 50, 28, 17, 28, 25, 43, 34
## $ jmlmotor    <dbl> 2, 2, 3, 0, 2, 1, 1, 2, 1, 0, 2, 1, 1, 4, 5, 1, 3, 1, 1, 1, 1, 1, 1, 2, 1, 0, 1
## $ jmlmobil    <dbl> 0, 1, 0, 0, 3, 1, 1, 1, 0, 0, 2, 1, 0, 0, 1, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0
## $ jmlsepeda   <dbl> 2, 4, 0, 2, 3, 2, 1, 1, 0, 0, 0, 2, 0, 2, 6, 1, 4, 1, 1, 0, 1, 0, 3, 3, 1, 0, 1, 1
## $ jmldevisa   <dbl> 4, 4, 4, 3, 5, 2, 2, 4, 1, 1, 3, 1, 5, 2, 2, 3, 5, 12, 5, 2, 1, 2, 5, 2, 6, 2, 12,
## $ jmlanak     <dbl> 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 2, 2, 0, 0, 0, 0, 3, 3, 0, 1, 0, 0, 0, 1, 2, 0, 0, 0
## $ jarak       <dbl> 3000, 3000, 3000, 7000, 280, 7000, 600, 7200, 0, 500, 5000, 4000, 2000, 5000, 1000
```

Latihan 1:

Berdasarkan output *glimpse* di atas, jawablah pertanyaan berikut:

1. Berapa jumlah observasi (baris) atau objek dalam data tersebut?
2. Berapa jumlah variabel (kolom) yang aktif digunakan dalam analisis?

9.6 Uji Kelayakan Data (Asumsi)

Sebelum melakukan ekstraksi dimensi, kita perlu memastikan data layak untuk dianalisis faktor/PCA. Dua indikator utama adalah **Uji Bartlett of Sphericity** dan **Measure of Sampling Adequacy (MSA)** atau **Kaiser-Meyer-Olkin (KMO)**.

9.6.1 Uji Bartlett of Sphericity

Uji ini melihat apakah terdapat korelasi antarvariabel dalam data. Syarat: Nilai $p < 0,05$.

```
# Uji Bartlett
cortest.bartlett(data_analisis)

## R was not square, finding R from data

## $chisq
## [1] 573.4283
##
## $p.value
## [1] 5.636504e-82
##
## $df
## [1] 66
```

Latihan 3:

Lihat nilai $p.value$ pada output di atas. Apakah nilai tersebut < 0.05 ? Apa kesimpulan Anda mengenai korelasi antar variabel dalam data ini? (Apakah matriks korelasi berbeda secara signifikan dengan matriks identitas?)

9.6.2 Uji KMO dan MSA

Nilai KMO keseluruhan harus $> 0,5$. Selain itu, nilai MSA per variabel (diagonal pada matriks anti-image correlation) juga harus $> 0,5$.

```
# Uji KMO
KMO(data_analisis)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = data_analisis)
## Overall MSA = 0.63
## MSA for each item =
##      ongkos      bparkir      durasi      bareng      toplajur      usia      jmlmotor      jmlmobil      jmlsepeda      jmldevisa
##      0.66      0.65      0.60      0.63      0.81      0.62      0.61      0.61      0.57      0.48
```

Berdasarkan hasil di atas: - Nilai KMO Keseluruhan (*Overall MSA*) = 0,63 (Layak, $> 0,5$). - Namun, jika dilihat per variabel (*MSA for each item*), terdapat variabel dengan nilai $< 0,5$ yaitu *jmldevisa* (0,48) dan *jmlanak* (0,44).

Sesuai prosedur, kita harus mengeluarkan variabel yang tidak memenuhi syarat MSA.


```
# Mengeluarkan variabel jml Dewasa dan jml Anak
data_analisis_final <- data_analisis |>
  select(-jml Dewasa, -jml Anak)
```

```
# Cek ulang KMO
KMO(data_analisis_final)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = data_analisis_final)
## Overall MSA = 0.65
## MSA for each item =
##      ongkos      bparkir      durasi      bareng      toplajur      usia      jmlmotor      jmlmobil      jmlsepeda      jarak
##      0.66      0.65      0.60      0.63      0.82      0.62      0.64      0.62      0.58      0.64
```

Sekarang seluruh variabel memiliki $MSA > 0,5$ dan KMO keseluruhan naik menjadi 0,68. Data siap diekstraksi.

9.7 Mengekstrak Dimensi Baru

Langkah selanjutnya adalah menentukan berapa jumlah dimensi (faktor/komponen) yang akan dibentuk. Kita dapat menggunakan **Analisis Paralel** atau melihat **Nilai Eigen** (*Eigenvalues*) dan **Scree Plot**.

9.7.1 Nilai Eigen dan Total Variansi

Kita akan melihat berapa banyak variansi yang bisa dijelaskan oleh setiap komponen.

```
# Melakukan PCA tanpa rotasi untuk melihat Eigenvalues
analisis_awal <- principal(data_analisis_final, nfactors = 10, rotate = "none")
```

```
# Menampilkan nilai eigen
analisis_awal$values
```

```
## [1] 2.4878770 1.6496907 1.2355641 1.0172985 0.8514614 0.8261586 0.6137405 0.5061041 0.4703792 0.341
```

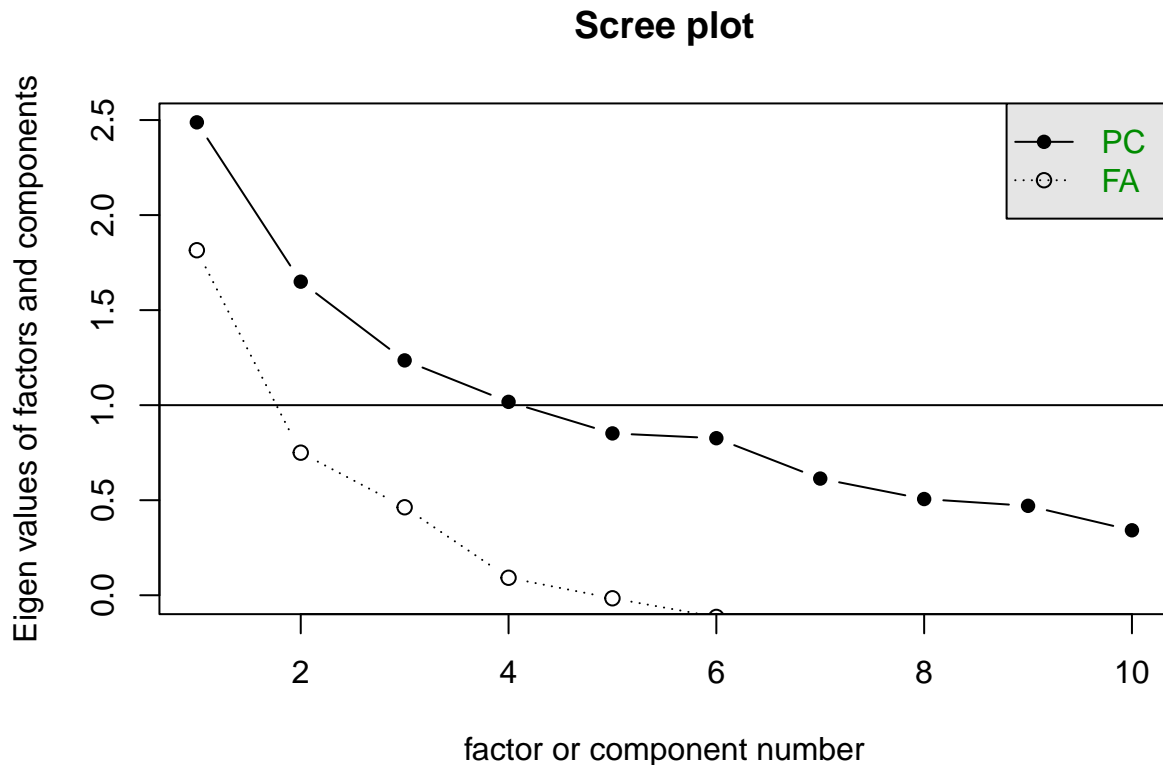
Kriteria umum penentuan jumlah dimensi: 1. **Kaiser's Criterion**: Ambil komponen dengan nilai eigen > 1. 2. **Cumulative Variance**: Ambil jumlah komponen yang menjelaskan total variansi > 60%.

Dari nilai eigen di atas, terdapat 4 komponen dengan nilai > 1 (2.488, 1.650, 1.236, 1.017).

9.7.2 Scree Plot

Grafik ini menunjukkan penurunan nilai eigen. Titik di mana grafik mulai melandai (siku) menunjukkan batas jumlah faktor.

```
# Membuat Scree Plot
scree(data_analisis_final, pc=TRUE)
```



Latihan 4:

1. Berdasarkan **Kaiser's Criterion** (Nilai Eigen > 1) pada sub-bab 9.7.1, berapa komponen yang sebaiknya diekstrak?
2. Berdasarkan **Scree Plot** di atas (lihat titik siku/ *elbow*), berapa komponen yang sebaiknya diekstrak?

Berdasarkan analisis-analisis sebelumnya, diputuskan untuk menggunakan **4 dimensi**.

9.8 Melakukan Analisis Faktor dan Rotasi

Kita akan melakukan ekstraksi 4 faktor menggunakan metode **Analisis Faktor** (untuk pengelompokan variabel) dan **PCA** (untuk pembentukan variat), kemudian melakukan **Rotasi Varimax** agar pengelompokan variabel lebih tegas (nilai *loading* kontras).

9.8.1 Analisis Faktor (Common Factor Analysis)

Analisis ini bertujuan mengelompokkan variabel-variabel yang mirip (korelasi tinggi) ke dalam faktor laten.

```
# Analisis Faktor dengan 4 faktor dan rotasi Varimax
# fm="pa" (Principal Axis) adalah metode umum untuk Common Factor Analysis di R (mirip SPSS)
af_result <- fa(data_analisis_final, nfactors = 4, rotate = "varimax", fm = "pa")
```

```
## maximum iteration exceeded
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, : The estimated weights
## incorrect. Try a different factor score estimation method.
```

```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An ultra-Heywood case w
## carefully
```

```
# Menampilkan hasil loading faktor
print(af_result$loadings, cutoff = 0.3)
```

```
##
## Loadings:
##      PA1      PA2      PA3      PA4
## ongkos              0.667
## bparkir              0.699
## durasi      0.660
## bareng
## toplajur    0.364              0.401
## usia              0.420
## jmlmotor            0.300
## jmlmobil            0.391
## jmlsepeda          1.004
## jarak      0.855
##
##      PA1      PA2      PA3      PA4
## SS loadings  1.404 1.329 1.238 0.424
## Proportion Var 0.140 0.133 0.124 0.042
## Cumulative Var 0.140 0.273 0.397 0.440
```

Catatan: *cutoff* = 0.3 digunakan untuk menyembunyikan nilai loading yang kecil agar tabel lebih mudah dibaca.

Cara Membaca Output: Perhatikan matriks komponen yang dirotasi (*Rotated Factor Matrix*). Setiap kolom (PA1, PA2, dst) merepresentasikan faktor. Nilai angka adalah *factor loading* (korelasi antara variabel dengan faktor).

Contoh Identifikasi Faktor 1 (PA1): Lihat kolom PA1. Variabel dengan nilai *loading* terbesar (dan di atas 0.5) adalah: - durasi (0.98) - jarak (0.96) Maka, Faktor 1 dibentuk oleh *durasi* dan *jarak*.

Latihan 5:

Berdasarkan output di atas, tentukan variabel pembentuk faktor lainnya:

1. **Faktor 2 (PA2):** Variabel apa saja yang memiliki *loading* tinggi di kolom ini?
2. **Faktor 3 (PA3):** Variabel apa saja yang memiliki *loading* tinggi di kolom ini?
3. **Faktor 4 (PA4):** Variabel apa saja yang memiliki *loading* tinggi di kolom ini?

9.8.2 Analisis Komponen Prinsip (PCA)

Jika tujuan kita adalah mereduksi data menjadi skor komponen untuk analisis lanjutan (misal regresi), kita menggunakan PCA.

```
# PCA dengan 4 komponen dan rotasi Varimax
pca_result <- principal(data_analisis_final, nfactors = 4, rotate = "varimax")

# Menampilkan hasil loading komponen (untuk interpretasi)
print(pca_result$loadings, cutoff = 0.3)
```

```
##
## Loadings:
##      RC1      RC3      RC2      RC4
## ongkos          0.791
## bparkir        0.823
## durasi      0.847
## bareng                      0.806
## toplajur    0.490  0.425
## usia              0.324 -0.601
## jmlmotor    0.344          0.597
## jmlmobil      0.385  0.638
## jmlsepeda          0.833
## jarak      0.825
##
##      RC1      RC3      RC2      RC4
## SS loadings  1.840  1.753  1.626  1.171
## Proportion Var 0.184  0.175  0.163  0.117
## Cumulative Var 0.184  0.359  0.522  0.639
```

```
# Menampilkan bobot skor komponen (weights) untuk pembentukan skor
print(pca_result$weights, digits = 3)
```

```
##      RC1      RC3      RC2      RC4
## ongkos    0.0254  0.4485 -0.0730  0.0301
## bparkir   -0.1330  0.5089 -0.0205  0.0298
## durasi     0.5237 -0.1667 -0.0855 -0.0144
## bareng    -0.0276 -0.0299  0.1825  0.7256
## toplajur   0.1979  0.1949  0.0396 -0.1924
## usia      -0.0194 -0.0251  0.1391 -0.4847
## jmlmotor   0.1783 -0.2001  0.3905  0.2852
## jmlmobil  -0.2100  0.2495  0.3974 -0.0890
## jmlsepeda -0.0496 -0.0641  0.5290 -0.0047
## jarak      0.4506  0.0303 -0.0532  0.0381
```

Identifikasi Persamaan Komponen

Komponen Prinsip (RC) adalah kombinasi linear dari variabel asal (yang sudah distandarisasi). Persamaan-nya dapat ditulis sebagai:

$$RC_j = w_{1j}Z_1 + w_{2j}Z_2 + \dots + w_{pj}Z_p$$

Dimana w adalah nilai **Component Score Coefficients (Weights)**, bukan nilai *loading*. Loading hanya menunjukkan korelasi, sedangkan weights menunjukkan bobot kontribusi setiap variabel dalam pembentukan skor komponen.

Contoh: Misalkan kita ingin membentuk persamaan untuk **RC1**. Lihat output **Weights** pada kolom **RC1**:
- durasi: 0.524 - jarak: 0.451 - Variabel lain memiliki bobot yang lebih kecil.

Maka persamaannya:

$$RC1 \approx 0.524(Z_{durasi}) + 0.451(Z_{jarak}) + \dots$$

Menghasilkan Component Score

Untuk mendapatkan nilai skor komponen setiap observasi secara otomatis, kita dapat mengakses objek **\$scores** dari hasil PCA.

```
# Menampilkan 6 baris pertama dari skor komponen
head(pca_result$scores)
```

```
##           RC1           RC3           RC2           RC4
## [1,] -0.1944617 -0.4804209  0.36717340 -0.54187967
## [2,] -0.7752196 -0.1911165  1.50812139 -1.14287093
## [3,] -0.2069996 -0.2693966 -0.04572532 -0.49628033
## [4,]  0.7677605 -0.2647856 -0.21284551  0.01899626
## [5,] -1.6702167 -0.2667413  1.87084197  0.22728535
## [6,]  0.1797747 -0.6385342  0.52415727 -2.00189106
```

Latihan 6:

Berdasarkan output weights PCA di atas, tuliskan persamaan matematis terbentuknya komponen **RC2**! (Sebutkan variabel mana saja yang memiliki kontribusi besar beserta koefisien weights-nya).

9.9 Interpretasi dan Penamaan Dimensi

Langkah terakhir adalah memberi nama pada dimensi yang terbentuk berdasarkan variabel-variabel pembentuknya.

Latihan 7: Tuliskan seluruh kelompok dari analisis faktor dan juga seluruh persamaan komponen yang dihasilkan dari PCA!

10 Referensi

Referensi

- M. A. M. Bindar. Faktor pemilihan moda pengunjung car-free day kota bandung. *Jurnal Perencanaan dan Pengembangan Kebijakan*, 2(3):239–249, 2022.
- CRAN. stplanr: A package for transport planning, 2026. URL <https://cran.r-project.org/web/packages/stplanr/vignettes/stplanr-paper.html>. Accessed: 2026-01-05.
- D. Irorere. Goodbye rstudio, welcome positron, 2026. URL <https://denniseirorere.com/posts/goodbye-rstudio-welcome-positron/>. Accessed: 2026-01-05.
- R. D. Peng. 2 history and overview of r | r programming for data science, 2026. URL <https://bookdown.org/rdpeng/rprogdatascience/history-and-overview-of-r.html>. Accessed: 2026-01-05.
- Posit. About, 2026a. URL <https://posit.co/about/>. Accessed: 2026-01-05.
- Posit. Shiny for r gallery, 2026b. URL <https://shiny.posit.co/r/gallery/>. Accessed: 2026-01-05.
- R Spatial. Spatial data science with r and “terra”, 2026. URL <https://rspatial.org/>. Accessed: 2026-01-05.
- ResearchGate. Structuration in implementing satu data indonesia policy in local government, 2026a. URL https://www.researchgate.net/publication/393695746_Structuration_in_implementing_Satu_Data_Indonesia_policy_in_local_government. Accessed: 2026-01-05.
- ResearchGate. Simulation: A tool for decision making in public policies field, 2026b. URL https://www.researchgate.net/publication/305639581_SIMULATION_A_TOOL_FOR_DECISION_MAKING_IN_PUBLIC_POLICIES_FIELD. Accessed: 2026-01-05.

- Romanian Statistical Review. Using r to get value out of public data, 2026. URL https://www.revistadestatistica.ro/wp-content/uploads/2014/07/RRS_2_2014_a10.pdf. Accessed: 2026-01-05.
- rOpenSci. Sustainable transport planning • stplanr, 2026. URL <https://docs.ropensci.org/stplanr/>. Accessed: 2026-01-05.
- UVA Library. Spatial r: Using the sf package, 2026. URL <https://library.virginia.edu/data/articles/spatial-r-using-sf-package>. Accessed: 2026-01-05.
- Wikipedia contributors. R (programming language), 2026. URL [https://en.wikipedia.org/wiki/R_\(programming_language\)](https://en.wikipedia.org/wiki/R_(programming_language)). Accessed: 2026-01-05.