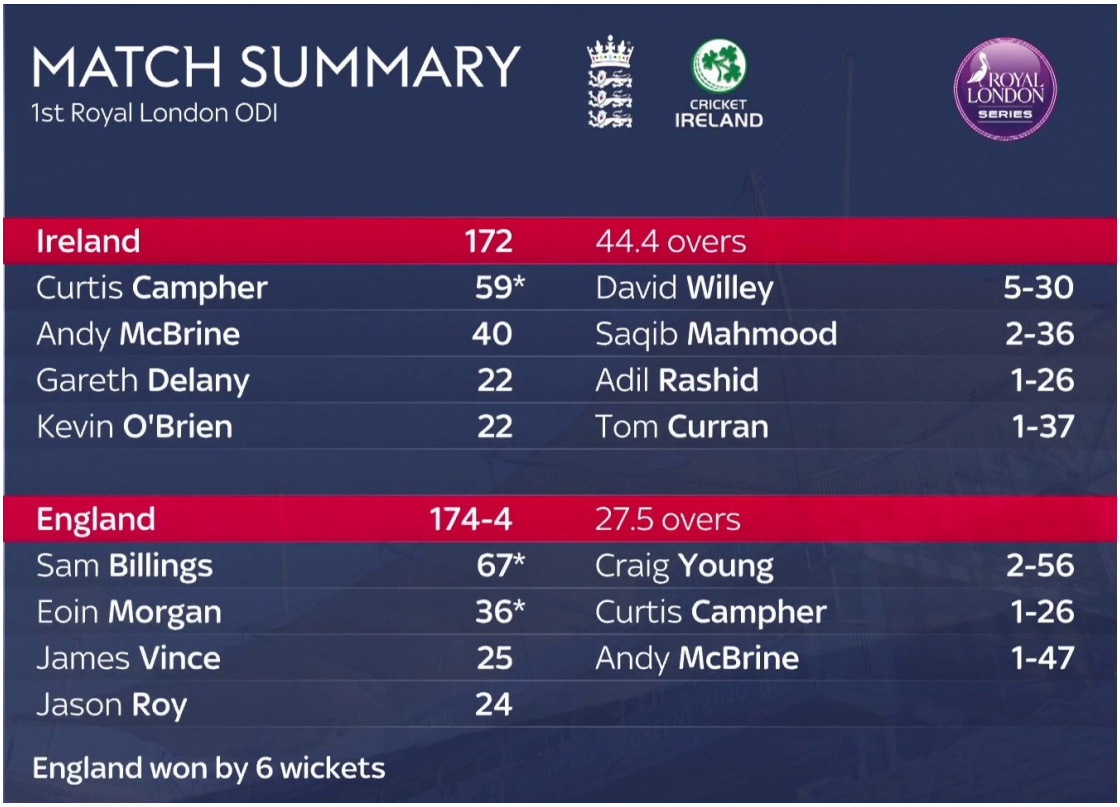
Cricket Match Report

Reproducible Match Summary with R

We can generate a detailed match report for a randomly selected T20 International (T20I) cricket match using ball-by-ball data. The report includes a match summary, batting and bowling statistics, and the final result, presented in a visually appealing table. We use various R packages to process the data and create the report, with each step explained.

A typical match summary, as shown below, captures the key moments of the game:



Setting up the Environment

Load essential R packages to handle data processing, file reading and table formatting.

```
library(pacman)
pacman::p_load(tidyverse, readxl, reader, plyr, gt, gtsummary)
```

Defining Wicket Types

To compute bowling statistics, we define the dismissal types credited to bowlers.

```
wkt_types <- c("bowled", "caught", "caught and bowled", "hit wicket", "lbw", "stumped")
```

Choosing a Random Match

We select a single T20I match randomly from the dataset using the unique identifier.

```
m_id <- t20_2$match_id |>
  unique() |>
  sample(1)
```

The identifier `m_id` pinpoints a specific T20I match for analysis. We then filter the dataset to focus on the chosen match.

```
match_data <- t20_2 |>
  filter(match_id == m_id)
```

For instance, if the `match_id` is 1415716, and the data appears as follows.

```
match_data
# # A tibble: 6 × 29
# match_id start_date venue      innings  ball batting_team bowling_team striker non_striker bowl
# <int> <date>     <chr>        <int> <dbl> <chr>        <chr>        <chr>   <chr>       <chr>
# 1  1415716 2024-06-08 Nassau Co…       1   0.1 Netherlands  South Africa M Levi… MP O'Dowd   M
# 2  1415716 2024-06-08 Nassau Co…       1   0.2 Netherlands  South Africa M Levi… MP O'Dowd   M
# 3  1415716 2024-06-08 Nassau Co…       1   0.3 Netherlands  South Africa M Levi… MP O'Dowd   M
# 4  1415716 2024-06-08 Nassau Co…       1   0.4 Netherlands  South Africa M Levi… MP O'Dowd   M
# 5  1415716 2024-06-08 Nassau Co…       1   0.5 Netherlands  South Africa Vikram… MP O'Dowd   M
# 6  1415716 2024-06-08 Nassau Co…       1   0.6 Netherlands  South Africa Vikram… MP O'Dowd   M
# #   19 more variables: runs_off_bat <int>, extras <int>, wides <dbl>, noballs <dbl>, byes <dbl>
# #   legbyes <dbl>, penalty <dbl>, wicket_type <chr>, player_dismissed <chr>, over <dbl>,
# #   over_type <chr>, isDot <dbl>, isOne <dbl>, isTwo <dbl>, isThree <dbl>, isFour <dbl>,
# #   isSix <dbl>, isOut <dbl>, team_runs <int>
```

- The output displays a sample of the selected match, a tibble with ball-by-ball data for match ID 1415716, including columns like `match_id` , `start_date` , `venue` , `innings` , `ball` , team details player names and other performance metrics.

Formatting the Match Date

We extract and format the match date for display purposes.

```
mat_date <- match_data$start_date |> first()
mat_date <- format(as.Date(mat_date), "%d %B, %Y")
mat_date
#> [1] "08 June, 2024"
```

Listing All Batters

We compile a unique list of all players who batted, including strikers and non-strikers.

```
all_players <- match_data |>
  dplyr::select(batting_team, striker, innings) |>
  dplyr::distinct() |>
  dplyr::rename(player = striker) |>
  bind_rows(
    match_data |>
      dplyr::select(batting_team, non_striker, innings) |>
      dplyr::distinct() |>
      dplyr::rename(player = non_striker))|>
  dplyr::distinct()
all_players
# A tibble: 18 × 3
#    batting_team player          innings
#    <chr>        <chr>             <int>
#  1 Netherlands  M Levitt              1
#  2 Netherlands  Vikramjit Singh       1
#  3 Netherlands  MP O'Dowd             1
#  4 Netherlands  SA Engelbrecht        1
#  5 Netherlands  BFW de Leede          1
#  6 Netherlands  SA Edwards            1
#  7 Netherlands  AT Nidamanuru         1
#  8 Netherlands  LV van Beek           1
#  9 Netherlands  TJG Pringle           1
# 10 Netherlands  PA van Meekeren       1
# 11 South Africa RR Hendricks          2
# 12 South Africa T Stubbs              2
# 13 South Africa AK Markram            2
# 14 South Africa H Klaasen             2
# 15 South Africa DA Miller             2
# 16 South Africa M Jansen              2
# 17 South Africa KA Maharaj            2
# 18 South Africa Q de Kock             2
```

Code explanation:

- Select the relevant columns for `striker` and remove duplicates.
- Rename `striker` to `player` .
- Combines striker data with non-striker data (processed similarly) and remove duplicates across both roles.

Computation of Batting and Bowling Performance

We calculate batting metrics such as runs, balls faced, fours, and sixes for each player.

```
bat_dtls_1 <- match_data |>
  dplyr::group_by(innings, batting_team, striker) |>
  dplyr::summarise(runs = sum(runs_off_bat), balls = n(),
            fours = sum(isFour), sixes = sum(isSix))
bat_dtls_1
# # A tibble: 17 × 7
# # Groups:   innings, batting_team [2]
#    innings batting_team striker          runs balls fours sixes
#      <int> <chr>        <chr>           <int> <int> <dbl> <dbl>
#  1       1 Netherlands  AT Nidamanuru       0     2     0     0
#  2       1 Netherlands  BFW de Leede        6    16     0     0
#  3       1 Netherlands  LV van Beek        23    23     3     0
#  4       1 Netherlands  M Levitt            0     4     0     0
#  5       1 Netherlands  MP O'Dowd           2     7     0     0
#  6       1 Netherlands  PA van Meekeren     1     1     0     0
#  7       1 Netherlands  SA Edwards         10     9     0     1
#  8       1 Netherlands  SA Engelbrecht     40    46     2     1
#  9       1 Netherlands  TJG Pringle         0     2     0     0
# 10       1 Netherlands  Vikramjit Singh    12    17     1     0
# 11       2 South Africa AK Markram          0     3     0     0
# 12       2 South Africa DA Miller          59    52     3     4
# 13       2 South Africa H Klaasen           4     7     0     0
# 14       2 South Africa KA Maharaj          0     1     0     0
# 15       2 South Africa M Jansen            3     5     0     0
# 16       2 South Africa RR Hendricks        3    10     0     0
# 17       2 South Africa T Stubbs           33    38     1     1
```

Code explanation:

- Group by `innings` , `tbatting_team` , and `striker` .
- Calculate total runs scored (excluding extras), number of balls faced, number of fours and sixes.

Incorporating Dismissal Information

We merge batting stats with dismissal data to track whether players were out.

```r
bat_dtls_2 <- match_data |>
  dplyr::distinct(player_dismissed) |>
  dplyr::mutate(out = 1) |>   # Mark dismissed players
  full_join(bat_dtls_1,
            by = c("player_dismissed" = "striker")) |>
  full_join(all_players,
            by = c("player_dismissed" = "player", "batting_team", "innings")) |>
  dplyr::filter(player_dismissed != "0", !is.na(innings)) |>
  dplyr::mutate(across(where(is.numeric), ~replace_na(., 0))) |>
  dplyr::mutate(innings = ifelse(innings == 0,
                                 max(innings[batting_team == batting_team]),
                                 innings)) |>
  dplyr::arrange(innings, desc(runs), balls, .by_group = TRUE)
bat_dtls_2
```

```
# # A tibble: 18 × 8
#    player_dismissed   out innings batting_team  runs balls fours sixes
#    <chr>            <dbl>   <int> <chr>        <int> <int> <dbl> <dbl>
#  1 SA Engelbrecht       1       1 Netherlands     40    46     2     1
#  2 LV van Beek          1       1 Netherlands     23    23     3     0
#  3 Vikramjit Singh      1       1 Netherlands     12    17     1     0
#  4 SA Edwards           1       1 Netherlands     10     9     0     1
#  5 BFW de Leede         1       1 Netherlands      6    16     0     0
#  6 MP O'Dowd            1       1 Netherlands      2     7     0     0
#  7 PA van Meekeren      0       1 Netherlands      1     1     0     0
#  8 AT Nidamanuru        1       1 Netherlands      0     2     0     0
#  9 TJG Pringle          1       1 Netherlands      0     2     0     0
# 10 M Levitt             1       1 Netherlands      0     4     0     0
# 11 DA Miller            0       2 South Africa    59    52     3     4
# 12 T Stubbs             1       2 South Africa    33    38     1     1
# 13 H Klaasen            1       2 South Africa     4     7     0     0
# 14 M Jansen             1       2 South Africa     3     5     0     0
# 15 RR Hendricks         1       2 South Africa     3    10     0     0
# 16 Q de Kock            0       2 South Africa     0     0     0     0
# 17 KA Maharaj           0       2 South Africa     0     1     0     0
# 18 AK Markram           1       2 South Africa     0     3     0     0
```

Code explanation:

- Extract unique dismissed players and label them as 1.
- Merge with batting stats and add all batters.
- Remove invalid entries and replace NA values with 0.
- Correct zero innings values.

Analyzing Bowling Performance

We compute runs conceded and wickets taken by bowlers, excluding extras.

```r
bowl_dtls_1 <- match_data |>
  dplyr::mutate(isBowlWkt = if_else(wicket_type %in% wkt_types, 1, 0)) |> # Identify valid bowlin
  dplyr::filter(legbyes == 0 & byes == 0) |>
  dplyr::group_by(bowling_team, bowler) |>
  dplyr::summarise(runs = sum(team_runs),
                   wicket = sum(isBowlWkt))

bowl_dtls_1
# # A tibble: 11 × 4
# # Groups:   bowling_team [2]
#    bowling_team bowler           runs wicket
#    <chr>        <chr>           <int>  <dbl>
#  1 Netherlands  BFW de Leede       34      1
#  2 Netherlands  LV van Beek        21      2
#  3 Netherlands  PA van Meekeren    13      0
#  4 Netherlands  TJG Pringle        14      0
#  5 Netherlands  VJ Kingma          12      2
#  6 Netherlands  Vikramjit Singh    11      0
#  7 South Africa A Nortje           19      2
#  8 South Africa K Rabada           27      0
#  9 South Africa KA Maharaj         24      0
# 10 South Africa M Jansen           20      2
# 11 South Africa OEG Baartman       11      4
```

Code explanation:

- Flags valid bowler wickets and excludes extras.
- Groups by team and bowler and calculate total runs and wickets.

Compute Economy Rates

We determine the number of valid balls bowled and calculate economy rates.

```r
bowl_dtls_2 <- match_data |>
  dplyr::filter(wides == 0 & noballs == 0) |>
  dplyr::group_by(innings, bowling_team, bowler) |>
  dplyr::summarise(balls = n()) |>
  left_join(bowl_dtls_1, by = c("bowling_team", "bowler")) |>
  dplyr::mutate(over = paste0(floor(balls/6), ".", balls%%6),  # Convert balls to overs
                econ = round(runs/balls*6, 2))                # Calculate economy rate
bowl_dtls_2
# # A tibble: 11 × 8
# # Groups:   innings, bowling_team [2]
#    innings bowling_team bowler          balls  runs wicket over   econ
#      <int> <chr>        <chr>           <int> <int>  <dbl> <chr> <dbl>
#  1       1 South Africa A Nortje           24    19      2 4.0    4.75
#  2       1 South Africa K Rabada           24    27      0 4.0    6.75
```

```
#  3       1 South Africa KA Maharaj         24    24      0 4.0    6
#  4       1 South Africa M Jansen           24    20      2 4.0    5
#  5       1 South Africa OEG Baartman       24    11      4 4.0    2.75
#  6       2 Netherlands  BFW de Leede       23    34      1 3.5    8.87
#  7       2 Netherlands  LV van Beek        24    21      2 4.0    5.25
#  8       2 Netherlands  PA van Meekeren    24    13      0 4.0    3.25
#  9       2 Netherlands  TJG Pringle        12    14      0 2.0    7
# 10       2 Netherlands  VJ Kingma          24    12      2 4.0    3
# 11       2 Netherlands  Vikramjit Singh     6    11      0 1.0   11
```

Code explanation:

- Counts only valid deliveries and groups by innings, team and bowlers.
- Counts valid balls and merge with runs and wickets.
- Calculate overs and economy rates.

Highlighting Top Performers

We select the top three batters and bowlers per innings based on runs and wickets.

```
bat_stat <- bat_dtls_2 |>
  dplyr::group_by(innings, batting_team) |>
  dplyr::arrange(desc(runs), balls, .by_group = TRUE) |>
  slice_head(n = 3)
bat_stat
# # A tibble: 6 × 8
# # Groups:   innings, batting_team [2]
#   player_dismissed   out innings batting_team  runs balls fours sixes
#   <chr>            <dbl>   <int> <chr>        <int> <int> <dbl> <dbl>
# 1 SA Engelbrecht       1       1 Netherlands     40    46     2     1
# 2 LV van Beek          1       1 Netherlands     23    23     3     0
# 3 Vikramjit Singh      1       1 Netherlands     12    17     1     0
# 4 DA Miller            0       2 South Africa    59    52     3     4
# 5 T Stubbs             1       2 South Africa    33    38     1     1
# 6 H Klaasen            1       2 South Africa     4     7     0     0

bowl_stat <- bowl_dtls_2 |>
  dplyr::group_by(innings, bowling_team) |>
  dplyr::arrange(desc(innings), desc(wicket), runs, .by_group = F) |>
  dplyr::slice_head(n = 3)
bowl_stat
# # A tibble: 6 × 8
# # Groups:   innings, bowling_team [2]
#   innings bowling_team bowler       balls  runs wicket over   econ
#     <int> <chr>        <chr>        <int> <int>  <dbl> <chr> <dbl>
# 1       1 South Africa OEG Baartman    24    11      4 4.0    2.75
# 2       1 South Africa A Nortje        24    19      2 4.0    4.75
```

```
# 3       1 South Africa M Jansen        24    20      2 4.0    5
# 4       2 Netherlands  VJ Kingma       24    12      2 4.0    3
# 5       2 Netherlands  LV van Beek     24    21      2 4.0    5.25
# 6       2 Netherlands  BFW de Leede    23    34      1 3.5    8.87
```

Extracting Match Details

We retrieve metadata such as the toss winner, player of the match, tournament and venue.

Toss Winner

```
toss <- match_sum |>
  dplyr::filter(match_id == m_id) |>
  dplyr::pull(toss_winner) |>
  unique()
toss
# [1] "South Africa"
```

Get the player of the match

```
pom <- match_sum |>
  dplyr::filter(match_id == m_id) |>
  dplyr::pull(player_of_match) |>
  unique()
pom
# [1] "DA Miller"
```

Get the Tournament or Series name (if any)

```
event <- match_sum |>
  dplyr::filter(match_id == m_id) |>
  dplyr::pull(event) |>
  unique()
event
# [1] "ICC Men's T20 World Cup"
```

Get the Match Venue

```
venue <- match_sum |>
  dplyr::filter(match_id == m_id) |>
  dplyr::pull(venue) |>
  unique()
venue
# [1] "New York, Nassau County International Cricket Stadium"
```

Summarise Team Total Runs, Wickets, and Overs for each Innings

```
mat_results <- match_data |>
  dplyr::group_by(innings, batting_team) |>
  dplyr::summarise(runs = sum(team_runs),
                   wickets = sum(isOut),
                   balls = sum(wides == 0 & noballs == 0)) |>
  ungroup() |>
  # inner_join(country_flag) |>  # Add flag (if applicable)
  dplyr::mutate(summary = case_when(
    batting_team == toss ~ paste0(batting_team, "      ",
                                  runs, "/",
                                  wickets," (",
                                  floor(balls/6),
                                  ".", balls%%6," ovr.)", "   "),
    .default = paste0(batting_team, "      ",
                      runs, "/",
                      wickets," (",
                      floor(balls/6), ".", balls%%6," ovr.)")))
mat_results
# # A tibble: 2 × 6
#   innings batting_team  runs wickets balls summary
#     <int> <chr>        <int>   <dbl> <int> <chr>
# 1       1 Netherlands    103       9   120 "Netherlands      103/9 (20.0 ovr.)"
# 2       2 South Africa   106       6   113 "South Africa      106/6 (18.5 ovr.)  \U0001fa99"
```

Code explanation:

- Groups by innings and team then compute runs, wickets and valid balls.
- Removes grouping and format a summary string , marking the toss winner with a coin emoji.

Format Top 3 Performers

We format the top batters and bowlers for the report.

Top Batters

```
batter <- bat_stat |>
  ungroup() |>
  dplyr::transmute(bat = player_dismissed,
                   runs = case_when(out == 1 ~ paste0(runs, " (", balls, ")"),
                                    TRUE ~ paste0(runs, "*", " (", balls,")")))
batter
# # A tibble: 6 × 2
#   bat             runs
#   <chr>           <chr>
```

```
# 1 SA Engelbrecht  40 (46)
# 2 LV van Beek     23 (23)
# 3 Vikramjit Singh 12 (17)
# 4 DA Miller       59* (52)
# 5 T Stubbs        33 (38)
# 6 H Klaasen       4 (7)
```

Code explanation:

- Removes grouping and format batter names and runs with an asterisk for not-out players.

Top Bowlers

```
bowler <- bowl_stat |>
  ungroup() |>
  dplyr::transmute(bowl = bowler,
                   wickets = paste0(wicket, "/", runs, " (", over, ")"))
bowler
# # A tibble: 6 × 2
#   bowl         wickets
#   <chr>        <chr>
# 1 OEG Baartman 4/11 (4.0)
# 2 A Nortje     2/19 (4.0)
# 3 M Jansen     2/20 (4.0)
# 4 VJ Kingma    2/12 (4.0)
# 5 LV van Beek  2/21 (4.0)
# 6 BFW de Leede 1/34 (3.5)
```

Code explanation:

- Format bowler names and performance figures.

Determination of Match Outcome

We create a function to summarize the match result.

```
mat_sum <- function() {
  if(mat_results$innings[1] == 1 & mat_results$runs[1] > mat_results$runs[2]){
    return(paste0(mat_results$batting_team[1], " won by ",
                  mat_results$runs[1] - mat_results$runs[2], " runs"))
  } else if(mat_results$innings[2] == 2 & mat_results$runs[2] > mat_results$runs[1]){
    return(paste0(mat_results$batting_team[2], " won by ",
                  10 - mat_results$wickets[2], " wickets"))
  } else {
    return(paste0("Match tied"))
  }
}
```

```
mat_sum()
# [1] "South Africa won by 4 wickets"
```

Match batting summary repeated for formatting purposes (not used further here)

```
mat_bat <- c(rep(mat_results$summary[1], 3), rep(mat_results$summary[2], 3))
```

Assembling the Report Table

We combine top batters and bowlers into a single table.

```
mat_report <- tibble(
  cbind(batter, bowler)) |>
  dplyr::mutate(blank1 = "    ", .before = 3) |>
  dplyr::mutate(blank2 = "    ", .before = 3) |>
  dplyr::mutate(blank3 = "    ", .before = 3) |>
  mutate_all(~replace(., is.na(.), ""))  # Replace NA with blank
mat_report
# # A tibble: 6 × 7
#   bat             runs     blank3 blank2 blank1 bowl         wickets
#   <chr>           <chr>    <chr>  <chr>  <chr>  <chr>        <chr>
# 1 SA Engelbrecht  40 (46)  "    " "    " "    " OEG Baartman 4/11 (4.0)
# 2 LV van Beek     23 (23)  "    " "    " "    " A Nortje     2/19 (4.0)
# 3 Vikramjit Singh 12 (17)  "    " "    " "    " M Jansen     2/20 (4.0)
# 4 DA Miller       59* (52) "    " "    " "    " VJ Kingma    2/12 (4.0)
# 5 T Stubbs        33 (38)  "    " "    " "    " LV van Beek  2/21 (4.0)
# 6 H Klaasen       4 (7)    "    " "    " "    " BFW de Leede 1/34 (3.5)
```

Create a Formatted Report Table

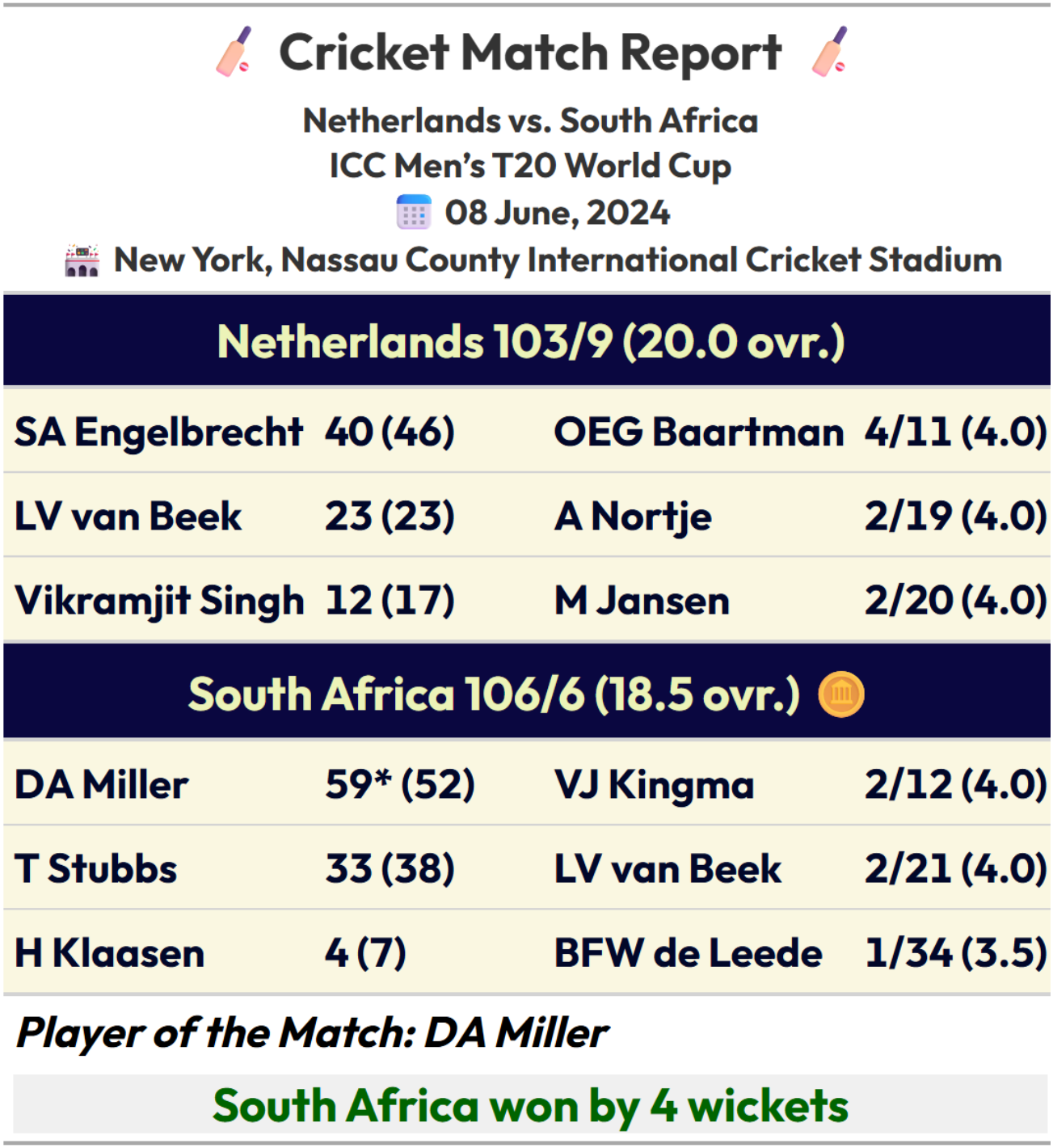
We use the `gt` package to produce a polished table.

```
mr <- mat_report |>
  gt() |>
  tab_header(title = md("  **Cricket Match Report**  "),
             subtitle = md(paste0("**", mat_results$batting_team[1],
                                  " vs. ",
                                  mat_results$batting_team[2],
                                  "<br>", event,
                                  "<br>  ", mat_date,
                                  "<br>  ", venue, "**"))) |>
  tab_row_group(label = mat_results$summary[1], rows = 1:3) |>
  tab_row_group(label = mat_results$summary[2], rows = 4:6) |>
  row_group_order(groups = mat_results$summary) |>
  tab_options(column_labels.hidden = TRUE) |>
```

```
  tab_style(
    style = list(
      cell_fill(color = "#0a0740"),
      cell_text(weight = "bold", color = "#ecf3b8", align = "center", size = "24px")
    ),
    locations = cells_row_groups()
  ) |>
  tab_style(
    style = list(
      cell_fill(color = "#faf7e1"),
      cell_text(weight = "bold", color = "#010729", align = "left")
    ),
    locations = cells_body(rows = c(1:6))
  ) |>
  opt_table_font(font = google_font("Outfit"), size = px(21)) |>
  tab_options(table.align = "center") |>
  tab_source_note(source_note = md(paste0(
      "<div style='text-align: left;
                  font-size: 22px;
                  font-weight: bold;
                  font-style: italic;
                  color: black;
                  width: 100%;'>Player of the Match: ", pom, "</div>"
  ))) |>
  tab_source_note(
    source_note = md(paste0(
      "<div style='text-align: center;
                  font-size: 24px;
                  font-weight: bold;
                  color: darkgreen;
                  background-color: #f0f0f0;
                  display: inline-block;
                  width: 100%;'>",
      mat_sum(),
      "</div>"
    ))
  )
mr
```

Code explanation:

- Converts the table into a `gt` object.
- Adds the title and subtitle with match details.
- Groups rows by innings and applies styling to row groups and cell.
- Sets font and size.
- Adds footnotes as *Player of the Match* and final result



Save the GT table as PNG image

```
gtsave(
  mr,
  filename = "match_report_02.png",
  path = "./plot/",
  vwidth = 950,
```

```
  vheight = 400
)
```