R Notebook

Code ▾

Cleaning the grounds raw data

Hide

```
library(dplyr)
```

```
Warning: package ‘dplyr’ was built under R version 4.2.3
Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

Hide

```
library(lubridate)
```

```
Warning: package ‘lubridate’ was built under R version 4.2.3
Attaching package: ‘lubridate’

The following objects are masked from ‘package:base’:

    date, intersect, setdiff, union
```

Hide

```
# Define the paths to the CSV files
file_paths <- c("perth.csv", "adilade.csv", "brisbane.csv",
                "geelong.csv", "melbourne.csv", "hobart.csv",
                "sydney.csv")

# Initialize an empty list to store the dataframes
cleaned_dataframes <- list()

# Loop through each file
for (i in 1:length(file_paths)) {
  file_path <- file_paths[i]

  # Read the CSV file
  df <- read.csv(file_path)

df <- df %>%
  mutate(Date = format(mdy(Start.Date), "%d/%m/%Y")) %>%
  select(-Start.Date)




  # Chnage coloumn name of the team
  names(df) [names(df)== "Team"] <- "Team 1"
  # Change column name 'Opposition' to 'Team2'
  names(df)[names(df) == "Opposition"] <- "Team2"

  # Remove "v " from each row of the column 'Team2'
  df$Team2 <- gsub("v ", "", df$Team2)

  # Add a new column 'Decision'
  df$Decision <- ifelse(df$Toss == "won" & df$Bat == "1st", "Elected to bat", "Elected to bowl")
  # Remove rows where 'Result' is 'aban', 'n/r', or 'tied'
  df <- df %>%
    filter(!(Result %in% c("aban", "n/r", "tied")))
  # Remove the entire 'BR' column
  df$BR <- NULL

  # Remove columns named 'X', 'X.1', and 'X.2'
 # Check if columns 'X', 'X.1', and 'X.2' exist in the dataframe and remove them if they do
 columns_to_remove <- c("X", "X.1", "X.2")
  existing_columns <- intersect(columns_to_remove, names(df))

if (length(existing_columns) > 0) {
  df <- df %>%
    select(-all_of(existing_columns))
}

  # Remove rows where 'Team 1' is empty
  # Remove rows where 'Team 1' is empty and there is data in 'Decision'
df <- df %>%
  filter(!(is.na(`Team 1`) | `Team 1` == "") | is.na(Decision))
  # Store the cleaned dataframe in the list
  cleaned_dataframes[[i]] <- df

  # Save the cleaned data (optional, uncomment the next line to save)
  write.csv(df, file = gsub(".csv", "_cleaned.csv", file_path), row.names = FALSE)
}

# Check for any NA dates that failed to parse
sapply(cleaned_dataframes, function(df) sum(is.na(df$Date)))
```

```
[1] 0 0 0 0 0 0 0
```

Cleaning the T20 data

Hide

```
library(dplyr)
#1.Read the CSV file (make sure to use the correct file path)
df <- read.csv("t20 data.csv")

# 2. Rename columns
names(df)[names(df) == "Team"] <- "Team1"
names(df)[names(df) == "Opposition"] <- "Team2"
df$Team2 <- gsub("v ", "", df$Team2)  # Remove the 'v' from Team2 names

#3. Make decision coloumns
df <- df %>%
  mutate(Decision = ifelse(Toss == "won" & Bat == "1st", "Elected to bat", "Elected to bowl"))

# 4. Remove consecutive rows with same Team2 name
df <- df %>%
  mutate(prev_team2 = lag(Team2)) %>%  # Create a temporary column for the previous row's Team2
  filter(Team1 != prev_team2 | is.na(prev_team2)) %>%  # Filter out the second row of the pair
  select(-prev_team2)   # Remove the temporary column

#5.  Remove the last row
df <- df[-nrow(df), ]

#6. Remove BR coloumn no need unneccasary
df <- select(df, -BR)



# Find the indices of rows where 'Margin' is '-'
hyphen_indices <- which(df$Margin == "-")

# Check if there are any '-' values, then update 'Result' accordingly
if (length(hyphen_indices) > 0) {
  # For the first '-' in 'Margin', set 'Result' to "No result"
  df$Result[hyphen_indices[1]] <- "No result"

  # For the rest of the '-' in 'Margin', if any, set 'Result' to "Match abandoned"
  if (length(hyphen_indices) > 1) {
    df$Result[hyphen_indices[-1]] <- "Match abandoned"
  }
}

# Remove rows where result is "match abandoned" or "no result"
df <- df[!(df$Result %in% c("Match abandoned", "No result")), ]



 # Save the cleaned data
write.csv(df, "cleaned_t20.csv", row.names = FALSE)
View(df)
```

Visualising which teams won the matches by choosing any of the decisons

Hide

```
# Load the necessary library

library(dplyr)
library(ggplot2)
install.packages("readr")
```

```
WARNING: Rtools is required to build R packages but is not currently installed. Please download
and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/asus/AppData/Local/R/win-library/4.2’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/readr_2.1.4.zip'
Content type 'application/zip' length 1151374 bytes (1.1 MB)
downloaded 1.1 MB
```

```
package ‘readr’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\asus\AppData\Local\Temp\RtmpEXtQcg\downloaded_packages
```

Hide

```
library(readr)
```

```
Warning: package ‘readr’ was built under R version 4.2.3
```

Hide

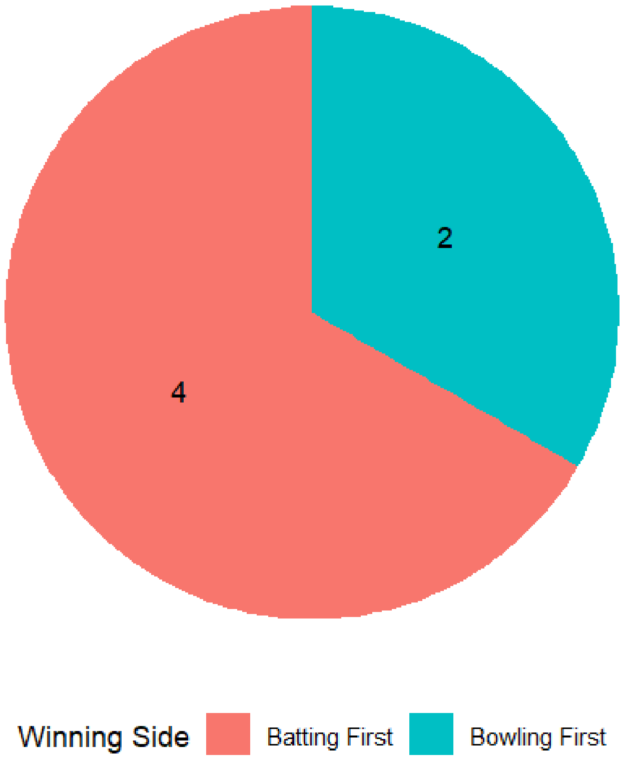
```
# Load the dataset
df <- read_csv("cleaned_t20.csv")
```

```
Rows: 40 Columns: 9── Column specification ──────────────────────────────────────────────────────
───────────────────────
Delimiter: ","
chr (9): Team1, Result, Margin, Toss, Bat, Team2, Ground, Start.Date, Decision
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.
```

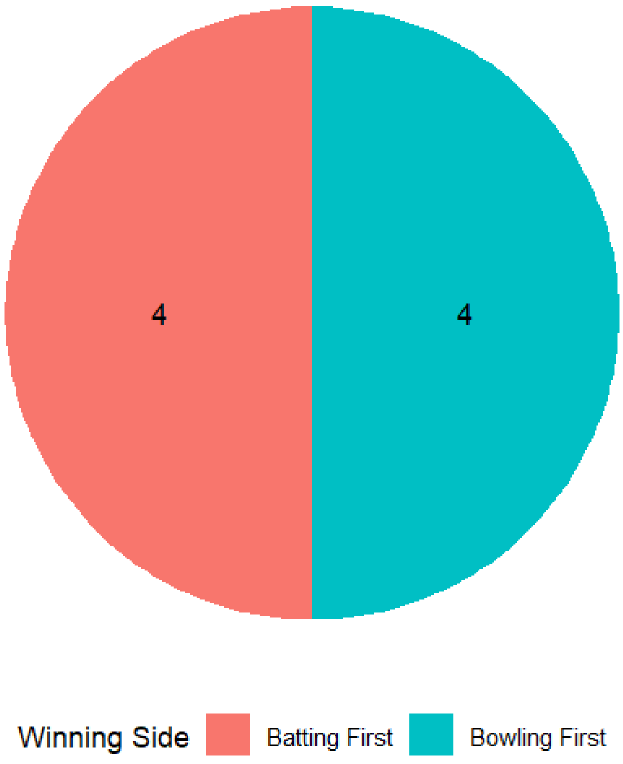
Hide

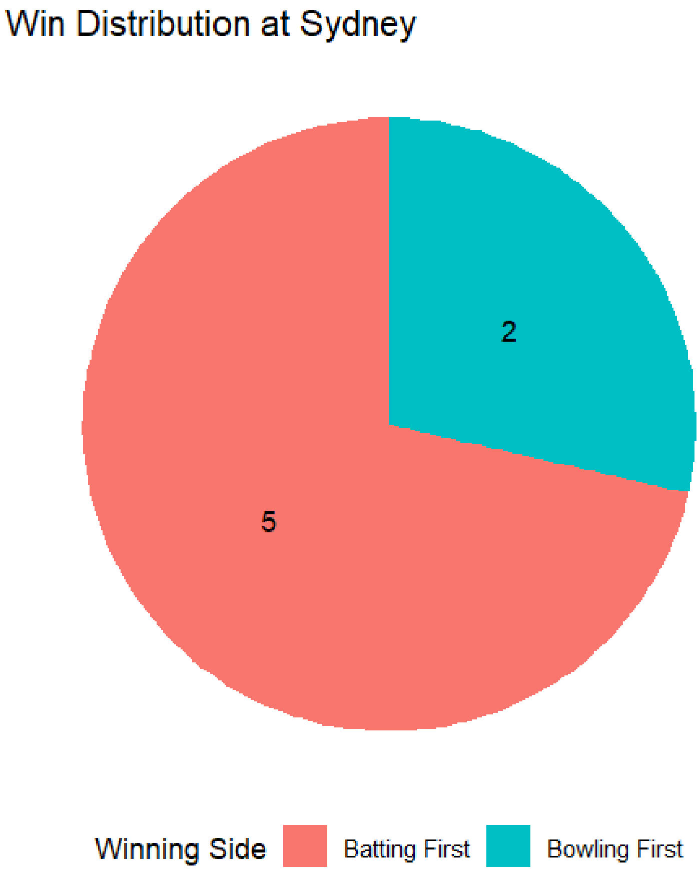
```
# Add a column to indicate if the winning team batted first or second
df <- df %>%
      mutate(WinningSide = ifelse(Result == "won" & Bat == "1st", "Batting First",
                                  ifelse(Result == "won" & Bat == "2nd", "Bowling First", "No Re
sult")))

ground_wins <- df %>%
  group_by(Ground, WinningSide) %>%
  summarise(Wins = sum(WinningSide != "No Result"),
            .groups = 'drop') # Ensure groups are dropped after summarising



unique_grounds <- unique(df$Ground)

for (ground in unique_grounds) {
  ground_data <- filter(ground_wins, Ground == ground)

  p <- ggplot(ground_data, aes(x = "", y = Wins, fill = WinningSide)) +
    geom_bar(stat = "identity", width = 1) +
    coord_polar("y", start = 0) +
    geom_text(aes(label = Wins), position = position_stack(vjust = 0.5)) +
    labs(title = paste("Win Distribution at", ground),
         fill = "Winning Side") +
    theme_void() +
    theme(legend.position = "bottom")

  print(p) # Print the pie chart for the current ground
}
```

Win Distribution at Geelong

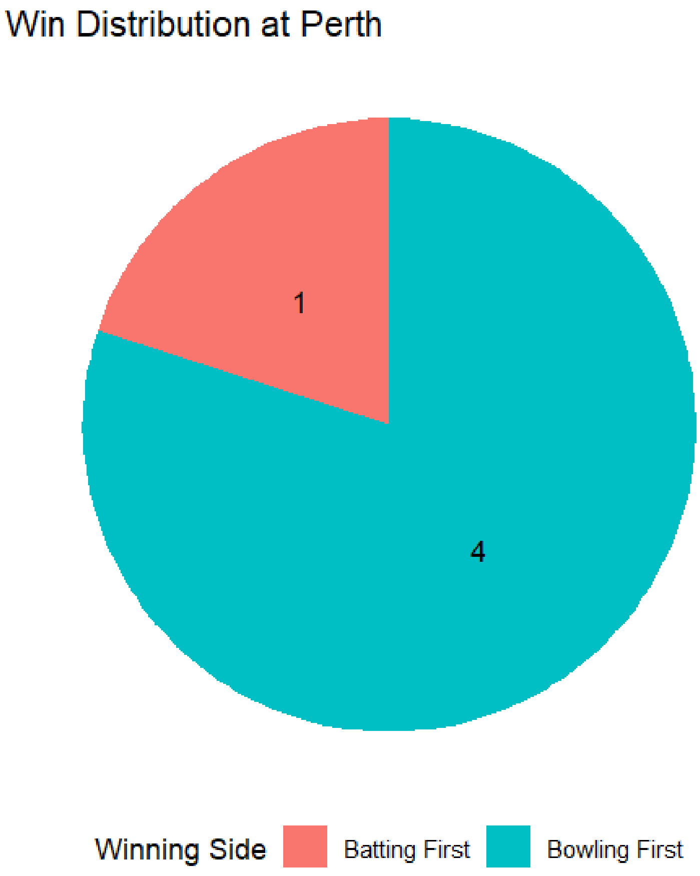


Win Distribution at Hobart

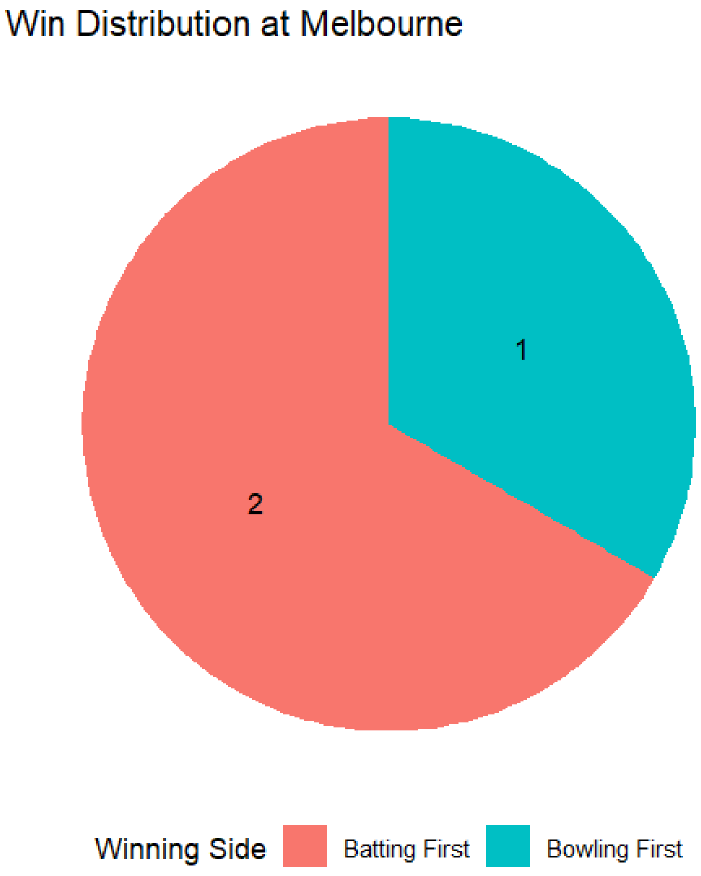


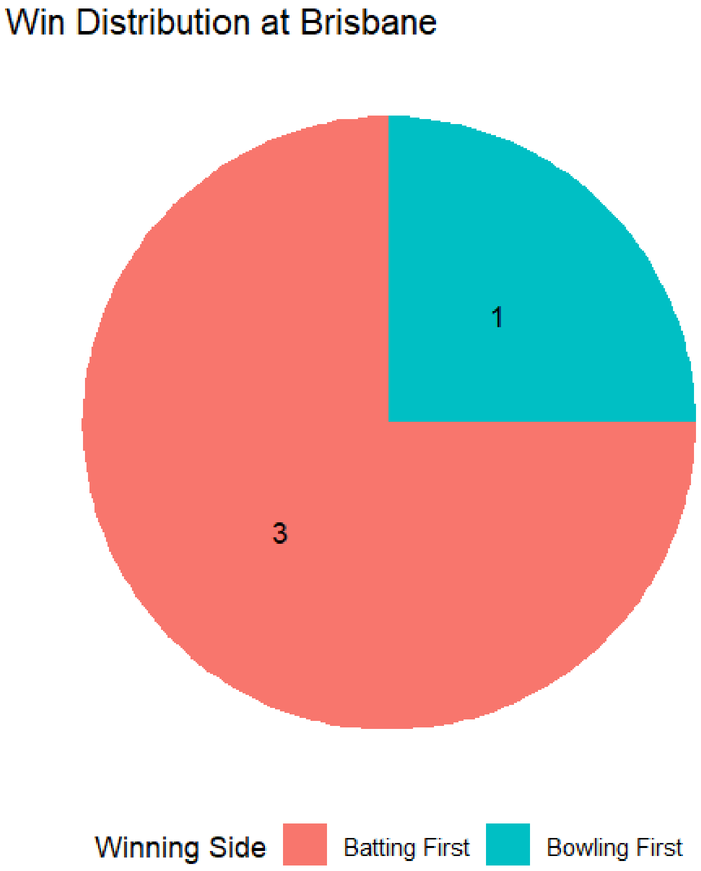


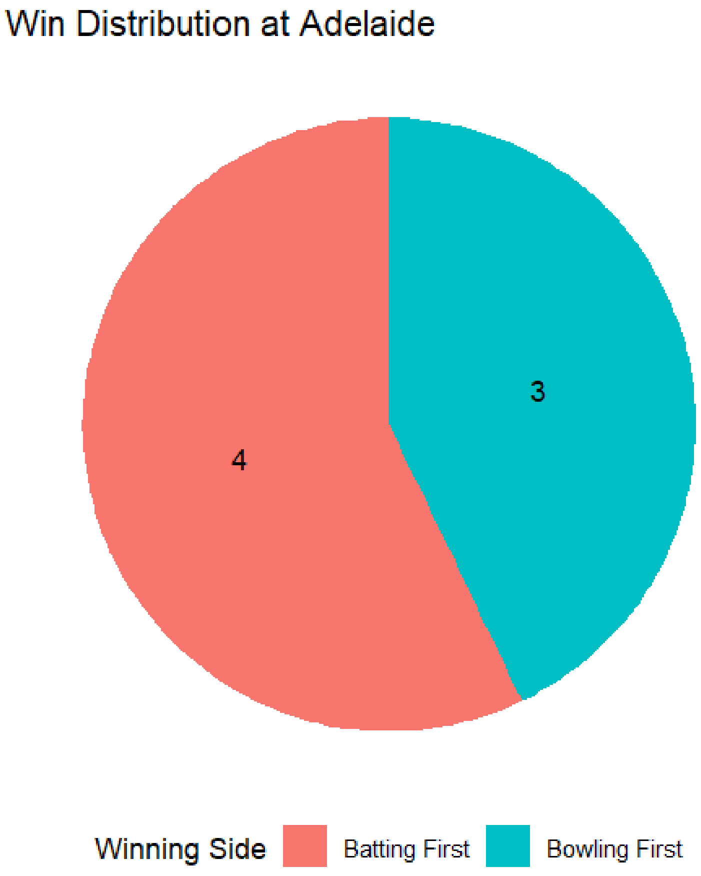
Win Distribution at Sydney



Win Distribution at Perth


Win Distribution at Melbourne
1
2
Winning Side
Batting First
Bowling First


Win Distribution at Brisbane
1
3
Winning Side
Batting First
Bowling First



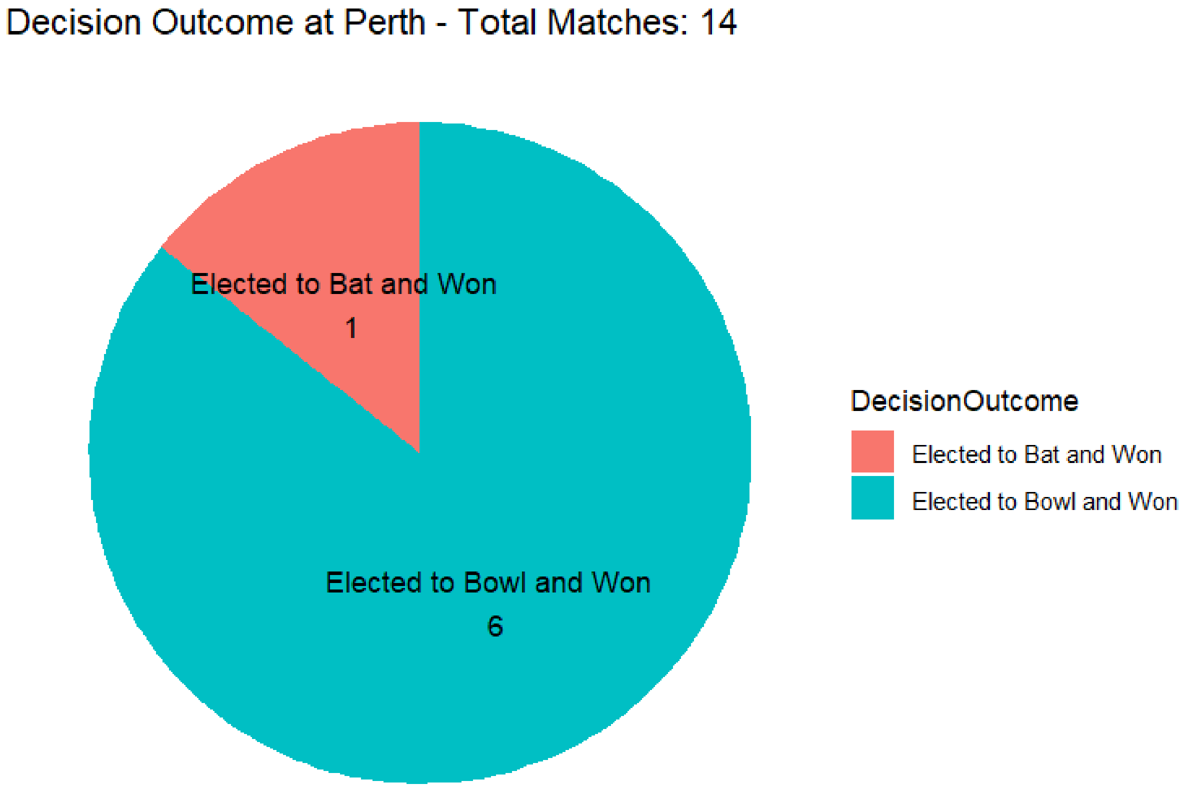
Hide

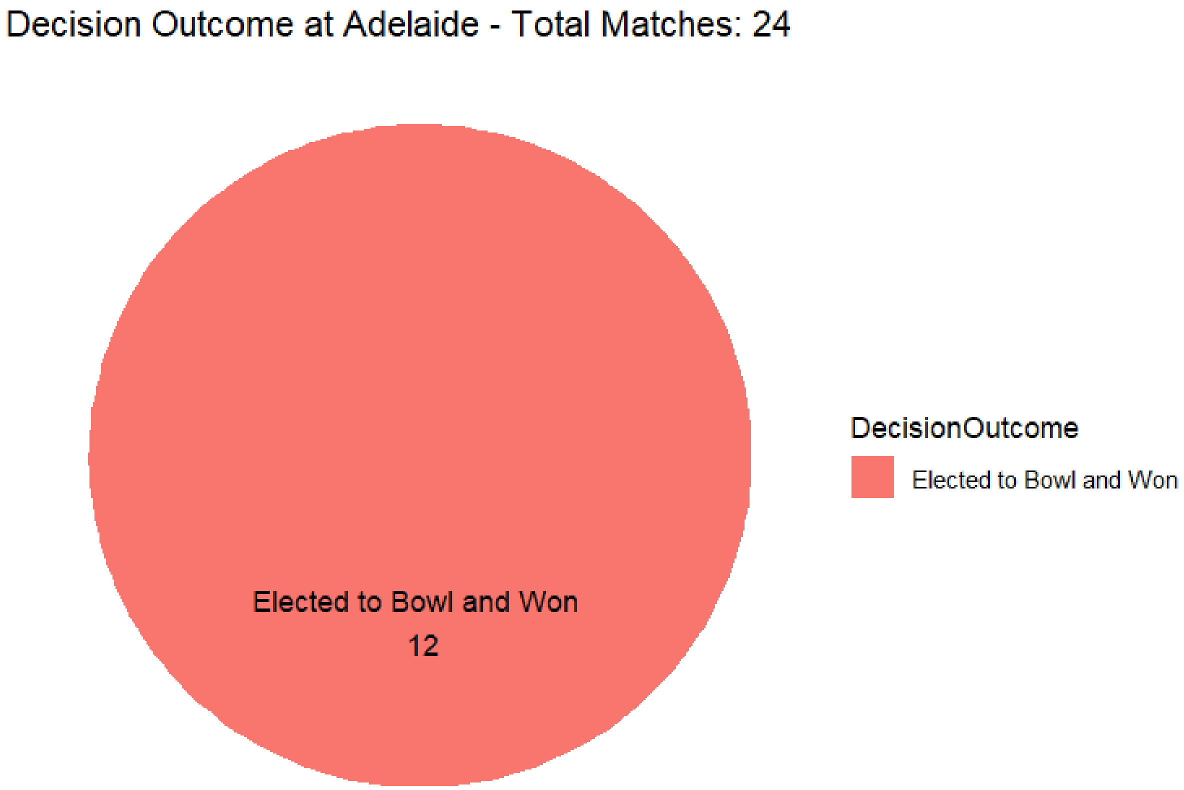
```
NA
NA
```

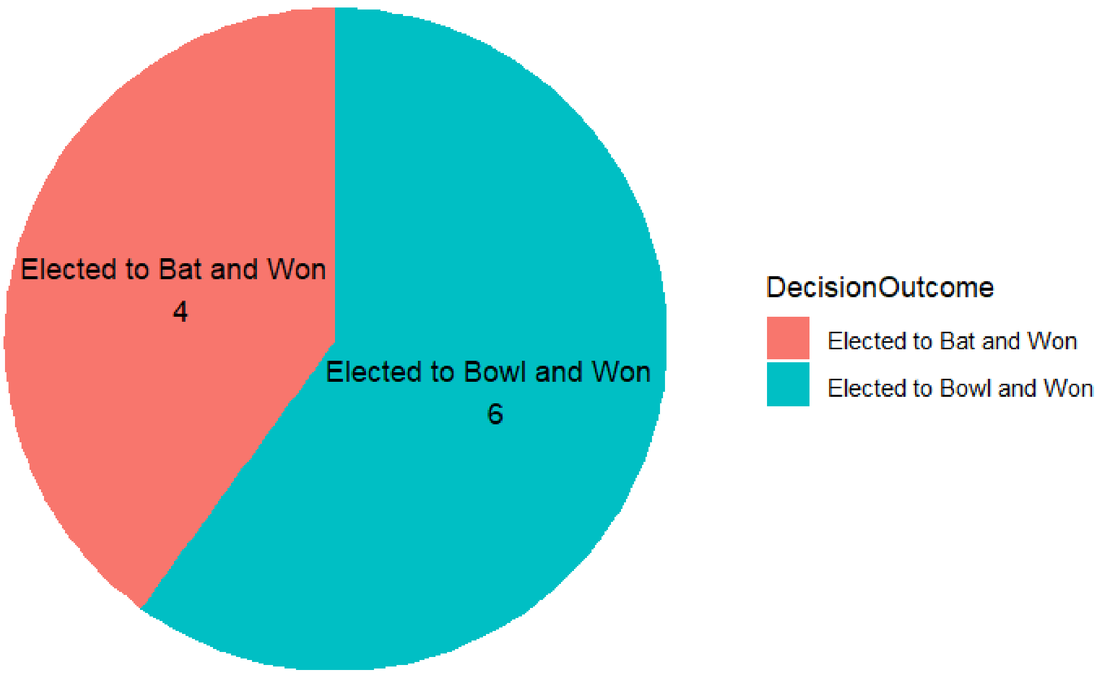
Visualising all the Venue ground and anlysing what decison has the most win percentage

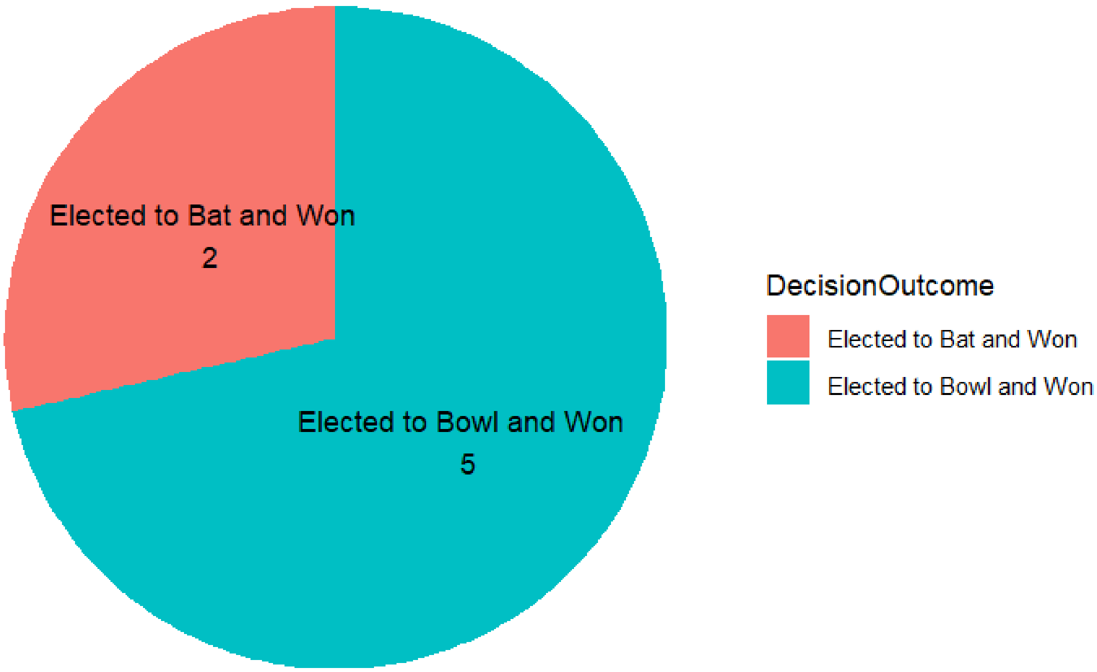
Hide

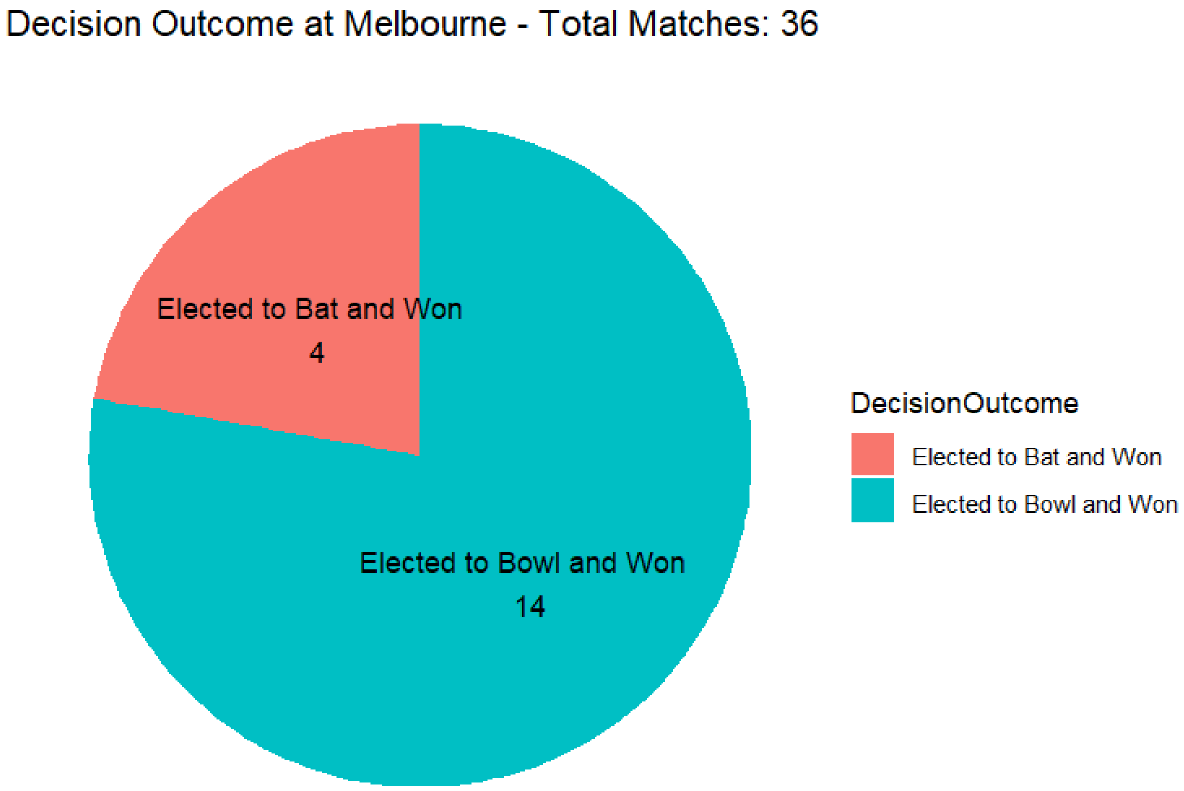
```
# Load necessary libraries
library(ggplot2)
library(dplyr)

# List of your dataset file paths
file_paths <- c("perth_cleaned.csv", "adilade_cleaned.csv", "brisbane_cleaned.csv",
                "geelong_cleaned.csv", "melbourne_cleaned.csv", "hobart_cleaned.csv",
                "sydney_cleaned.csv")

# Function to process each dataset and create a pie chart
process_and_plot <- function(file_path) {
  # Read the data
  data <- read.csv(file_path)

  # Assuming the ground name is consistent across each dataset
  ground_name <- unique(data$Ground)[1]

  # Create a new column to indicate if the team that made the decision won
  data$DecisionOutcome <- ifelse(data$Decision == "Elected to bat" & data$Result == "won", "Elec
ted to Bat and Won",
                                 ifelse(data$Decision == "Elected to bowl" & data$Result == "wo
n", "Elected to Bowl and Won", "Lost"))

  # Filter out the "Lost" decisions
  win_data <- filter(data, DecisionOutcome != "Lost")

  # Count the occurrences of each decision and match outcome
  decision_result_counts <- table(win_data$DecisionOutcome)

  # Convert the table to a dataframe for ggplot
  decision_result_df <- as.data.frame(decision_result_counts)
  colnames(decision_result_df) <- c("DecisionOutcome", "Count")

  # Number of matches in the dataset
  num_matches <- nrow(data)

  # Create a pie chart
  pie_chart <- ggplot(decision_result_df, aes(x = "", y = Count, fill = DecisionOutcome)) +
    geom_bar(stat = "identity", width = 1) +
    coord_polar("y", start = 0) +
    theme_void() +
    geom_text(aes(label = paste(DecisionOutcome, "\n", Count)), position = position_stack(vjust
= 0.5)) +
    ggtitle(paste("Decision Outcome at", ground_name, "- Total Matches:", num_matches))

  # Print the pie chart
  print(pie_chart)
}

# Apply the function to each file
for (file_path in file_paths) {
  process_and_plot(file_path)
}
```

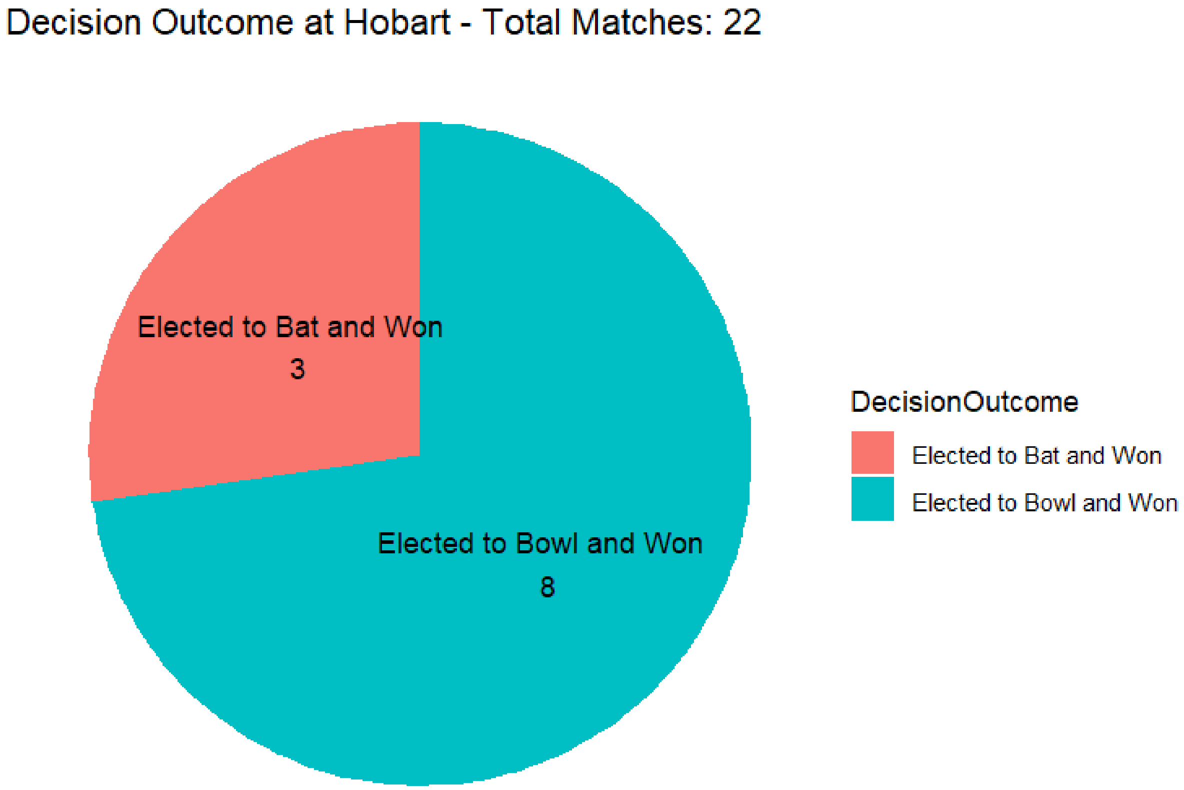


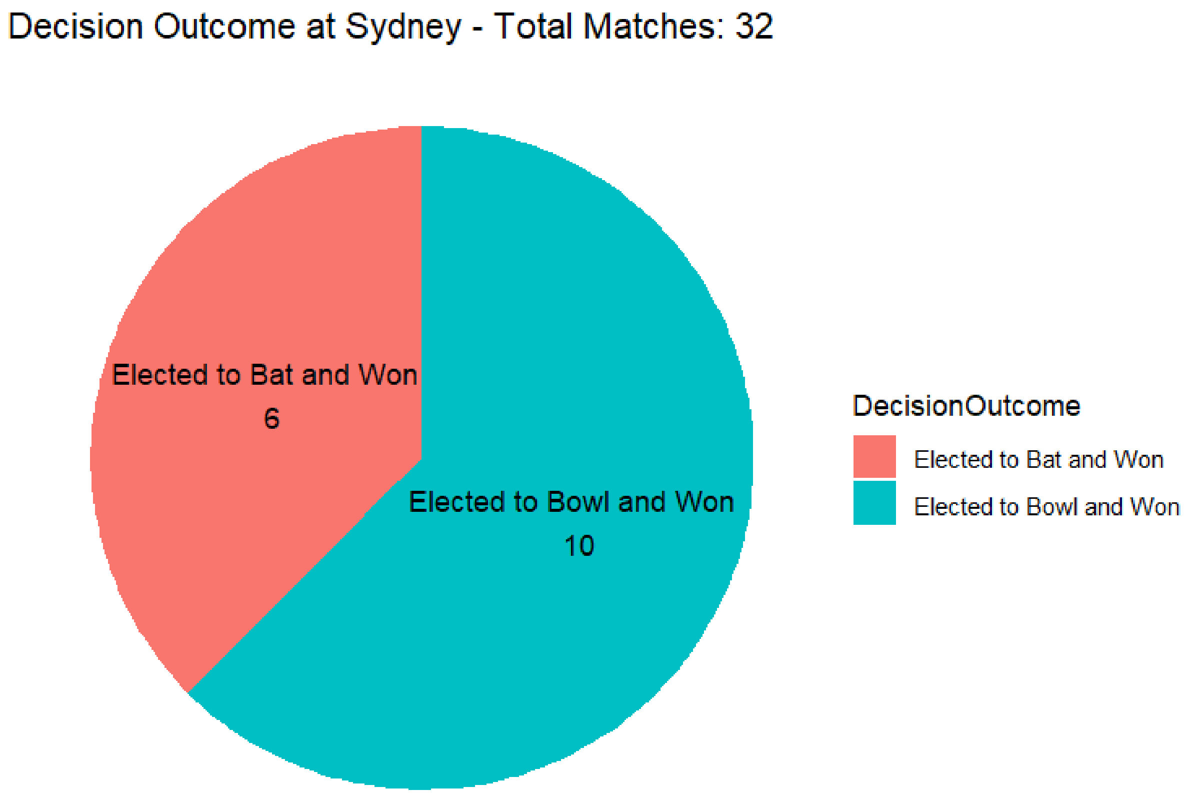



Decision Outcome at Brisbane - Total Matches: 20


Decision Outcome at Geelong - Total Matches: 14


Decision Outcome at Melbourne - Total Matches: 36


Decision Outcome at Hobart - Total Matches: 22



Hide

```
NA
NA
```

Showing win percentage of eachground with their decisions

Hide

```
# Load necessary library
library(dplyr)

# Read the CSV file
df <- read.csv("cleaned_t20.csv")

# Process the data to determine the outcome based on decision and toss
df <- df %>%
  mutate(DecisionWon = case_when(
    Toss == "won" & Decision == "Elected to bowl" & Result == "won" ~ TRUE,
    Toss == "lost" & Decision == "Elected to bowl" & Result == "lost" ~ TRUE,
    Toss == "won" & Decision == "Elected to bat" & Result == "won" ~ TRUE,
    Toss == "lost" & Decision == "Elected to bat" & Result == "lost" ~ TRUE,
    TRUE ~ FALSE
  ))

# Counting the number of wins based on the decision for each ground
wins_df <- df %>%
  group_by(Ground, Decision) %>%
  summarise(Wins = sum(DecisionWon))
```

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

Hide

```
# Counting the total matches played on each ground for each decision
total_matches_df <- df %>%
  group_by(Ground, Decision) %>%
  summarise(Total_Matches = n())
```

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

Hide

```
# Merging the wins and total matches dataframes
merged_df <- merge(wins_df, total_matches_df, by = c("Ground", "Decision"))

# Adding a column for win percentage
merged_df$Win_Percentage <- (merged_df$Wins / merged_df$Total_Matches) * 100

# Ordering the final dataframe by Ground and Win Percentage
final_df <- merged_df[order(merged_df$Ground, -merged_df$Win_Percentage), ]

# Display the final dataframe
print(final_df)
```

	Ground <chr>	Decision <chr>	Wins <int>	Total_Matches <int>	Win_Percentage <dbl>
1	Adelaide	Elected to bowl	1	7	14.28571

	Ground	Decision	Wins	Total_Matches	Win_Percentage
	<chr>	<chr>	<int>	<int>	<dbl>
2	Brisbane	Elected to bat	2	2	100.00000
3	Brisbane	Elected to bowl	0	2	0.00000
4	Geelong	Elected to bat	2	2	100.00000
5	Geelong	Elected to bowl	0	4	0.00000
6	Hobart	Elected to bat	1	1	100.00000
7	Hobart	Elected to bowl	0	7	0.00000
8	Melbourne	Elected to bat	1	1	100.00000
9	Melbourne	Elected to bowl	1	2	50.00000
10	Perth	Elected to bat	1	1	100.00000

1-10 of 13 rows Previous **1** 2 Next

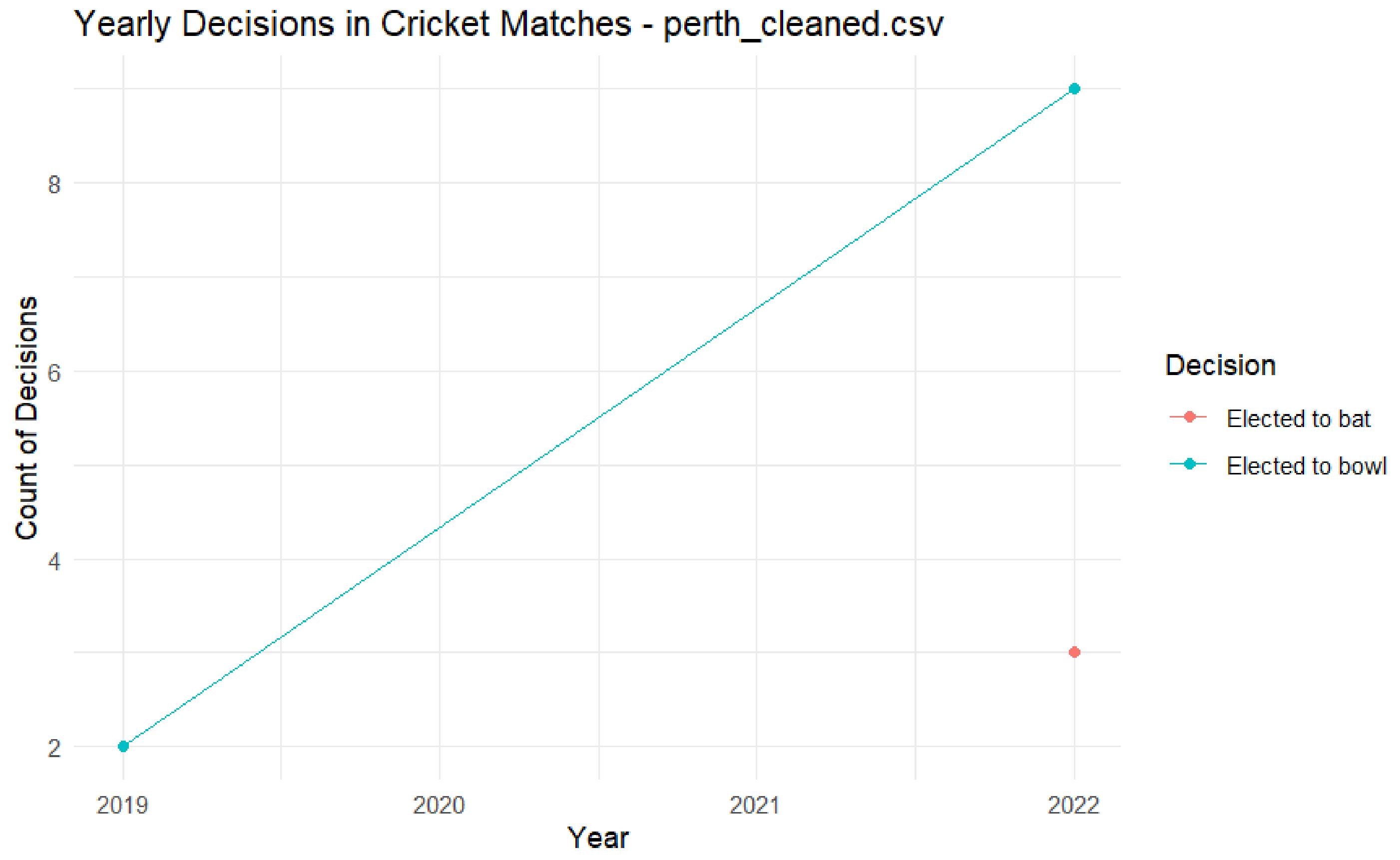
Hide

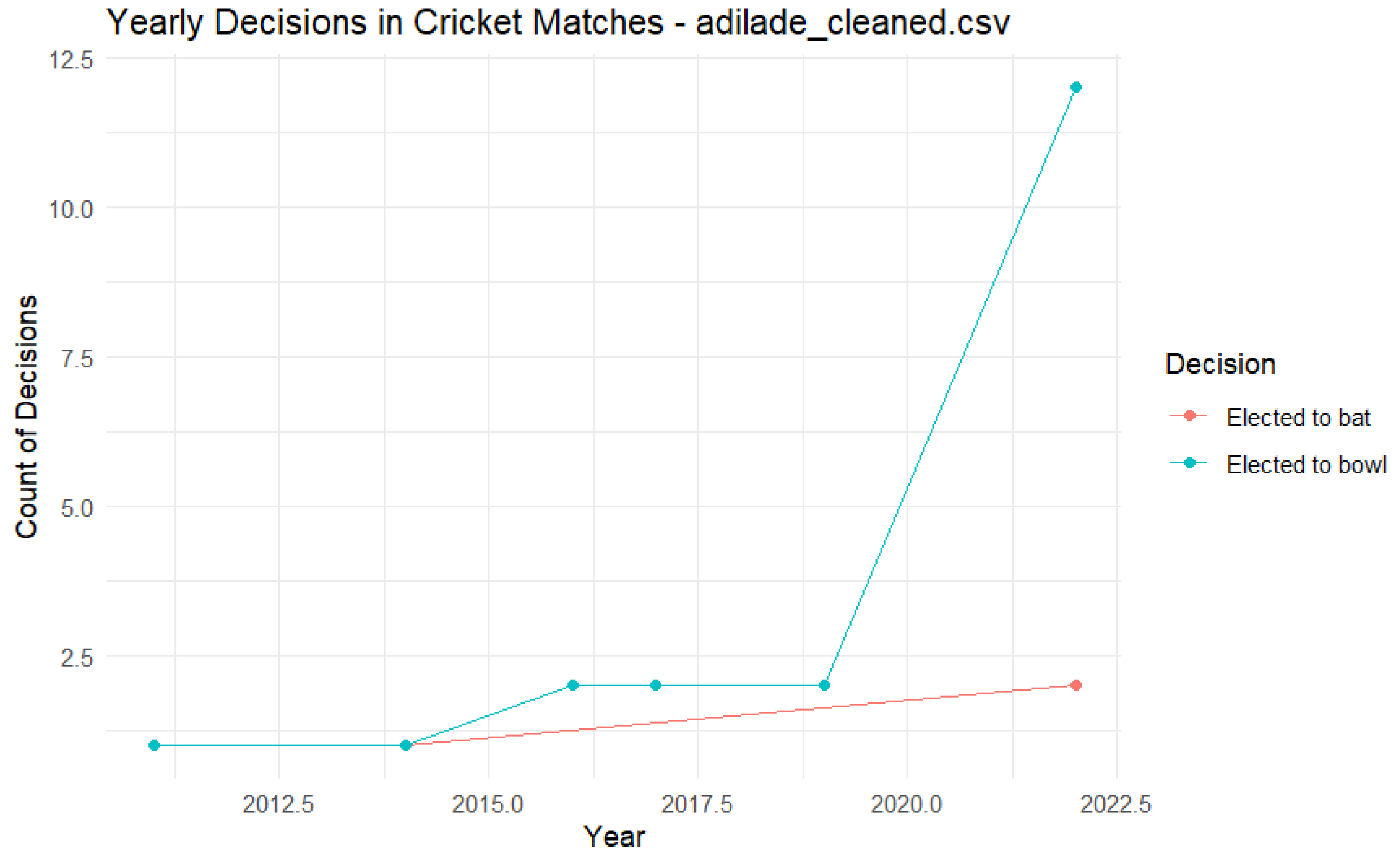
```
NA
NA
NA
```

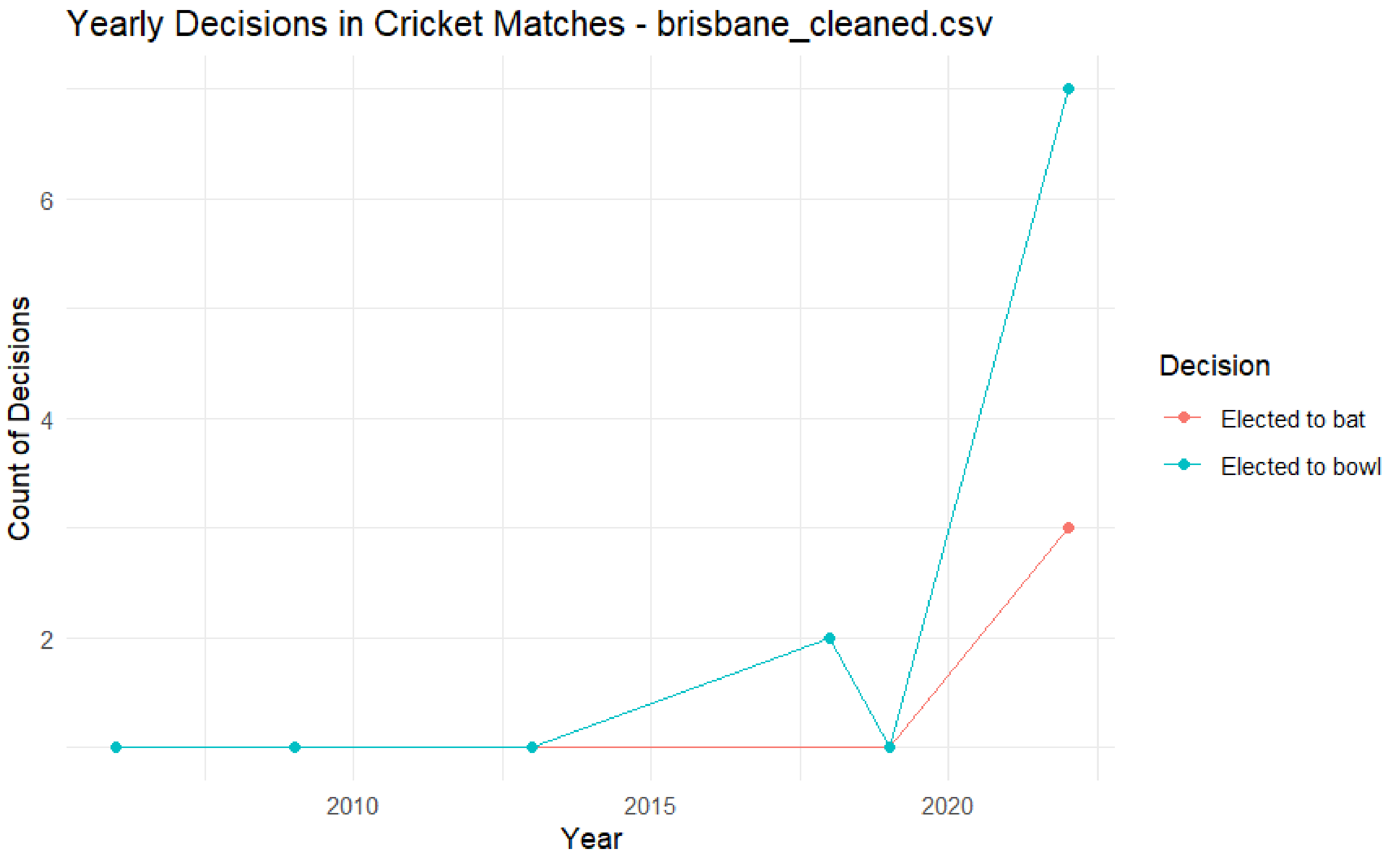
Time series anlysis on each ground

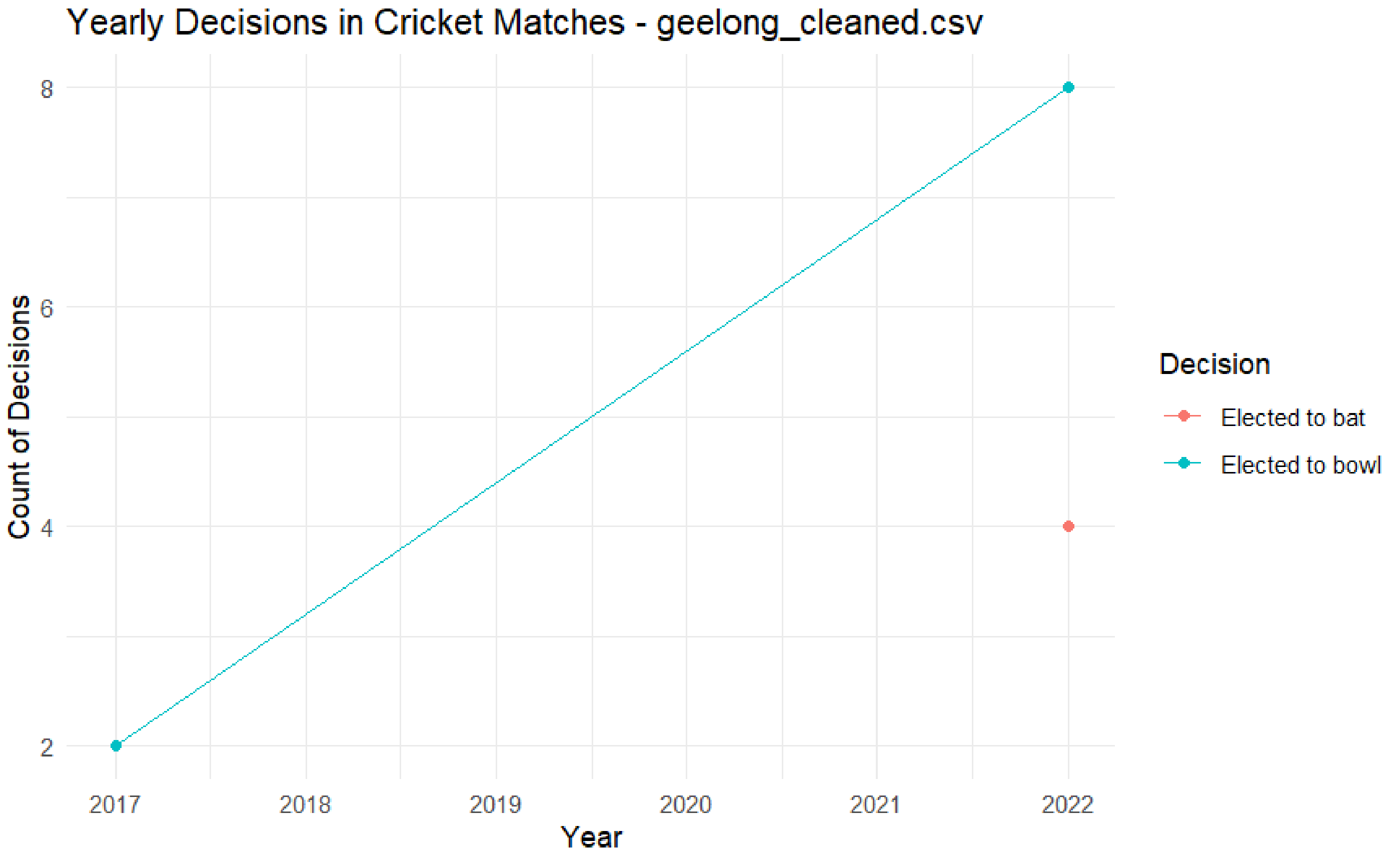
Hide

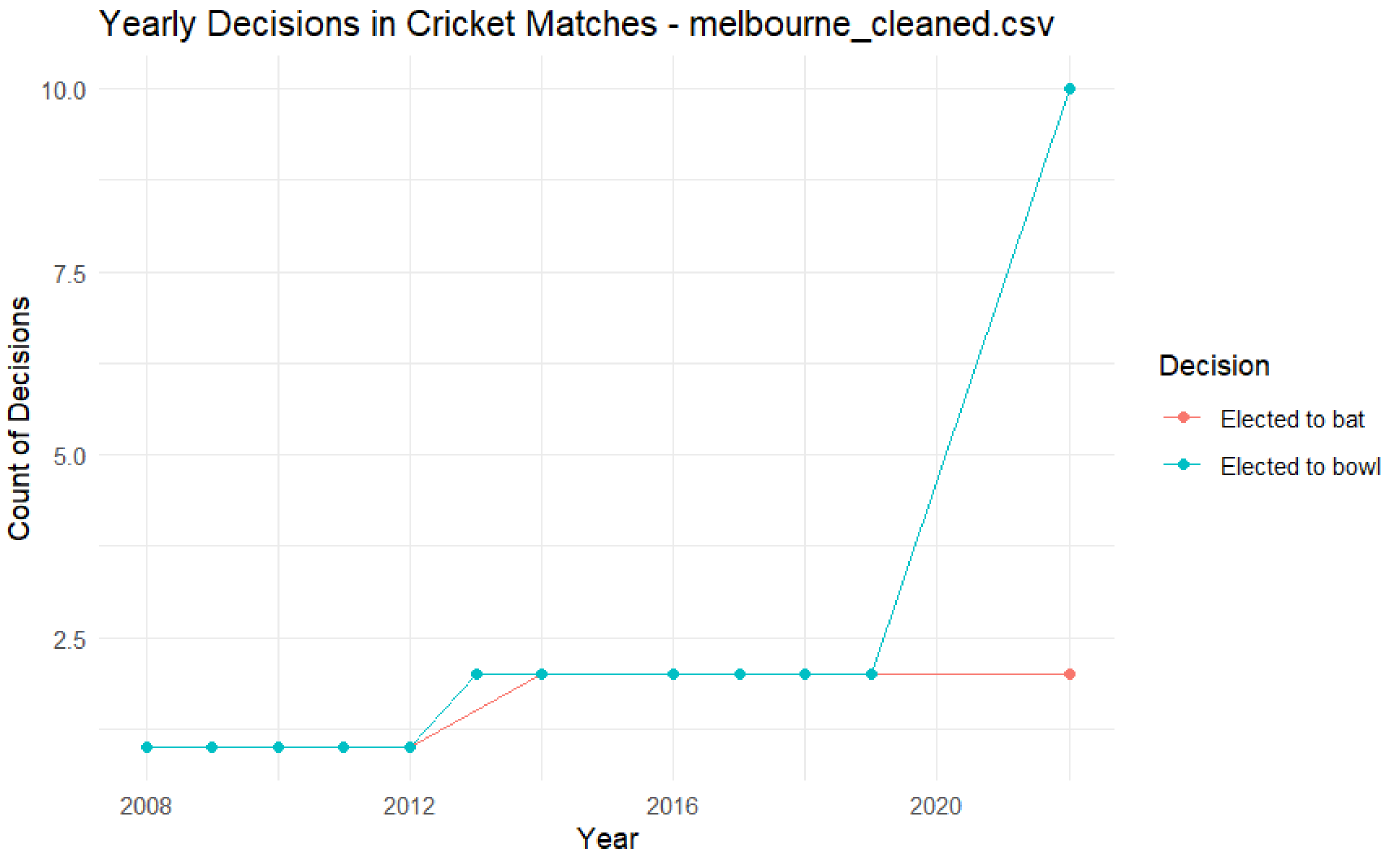
```
# Load necessary libraries
library(dplyr)
library(lubridate)
library(ggplot2)

# Define the paths to the CSV files
file_paths <- c("perth_cleaned.csv", "adilade_cleaned.csv", "brisbane_cleaned.csv",
                "geelong_cleaned.csv", "melbourne_cleaned.csv", "hobart_cleaned.csv",
                "sydney_cleaned.csv")

# Loop through each file
for (file_path in file_paths) {

  # Read the CSV file
  data <- read.csv(file_path)

  # Process the data
  data <- data %>%
    mutate(Decision.Maker = ifelse(Toss == "lost", "Team2", "Team1"),
           Match.Winner = ifelse(Result == "won", "Team1", "Team2"),
           Success = Decision.Maker == Match.Winner) %>%
    mutate(Date = dmy(Date),
           Year = year(Date))

  # Time series analysis for the most taken decisions yearly
yearly_decisions <- data %>%
  group_by(Year, Decision) %>%
  summarise(Count = n(), .groups = 'drop') %>% # Added .groups argument here
  arrange(Year, desc(Count))

# Plotting the line graph for decisions
  plot_decision <- ggplot(yearly_decisions, aes(x = Year, y = Count, group = Decision, color = D
ecision)) +
    geom_line() +
    geom_point() +
    theme_minimal() +
    labs(title = paste("Yearly Decisions in Cricket Matches -", basename(file_path)),
         x = "Year",
         y = "Count of Decisions",
         color = "Decision")

  # Print the plot
  print(plot_decision)

}
```

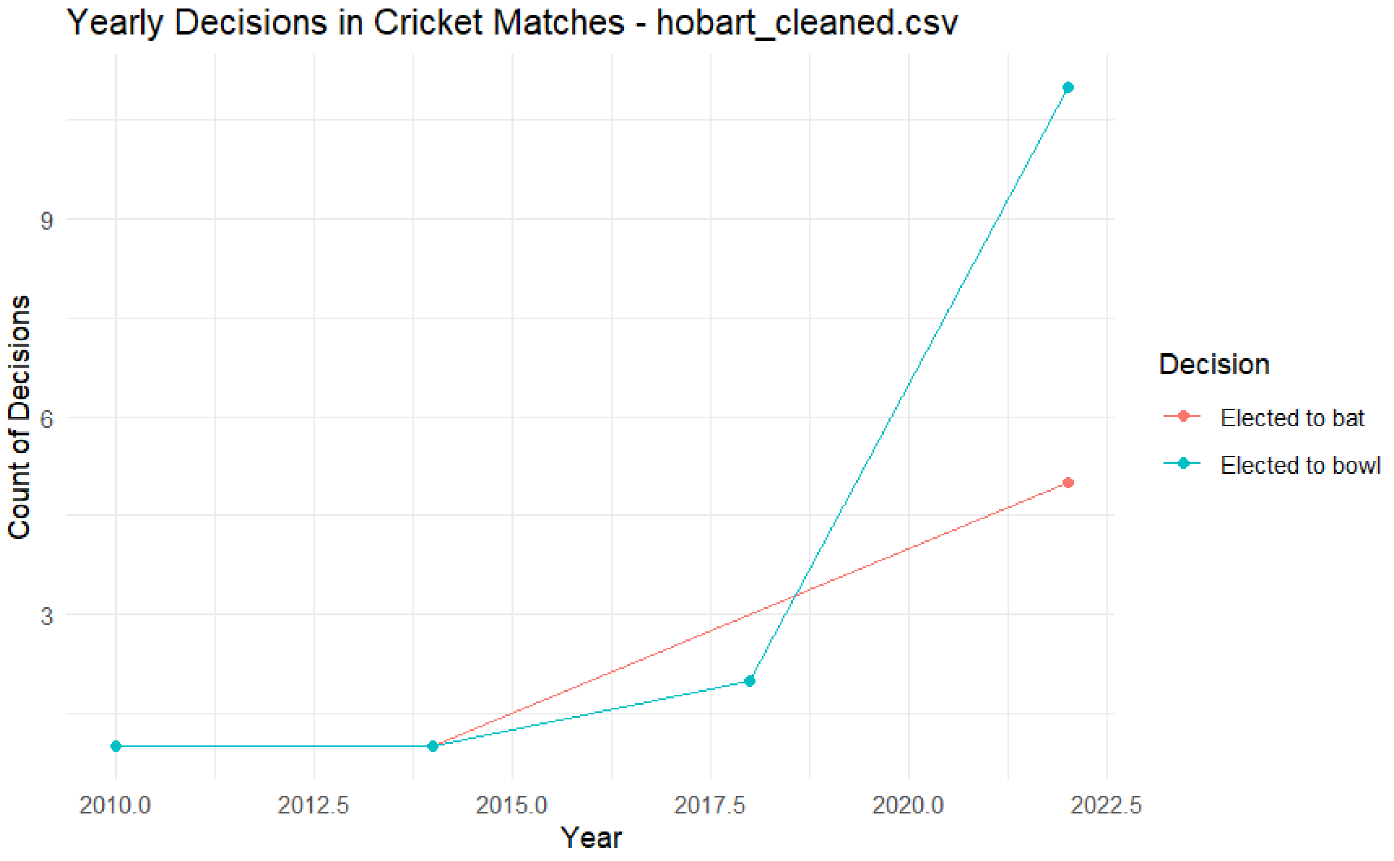

Yearly Decisions in Cricket Matches - perth_cleaned.csv
Count of Decisions
8
6
4
2
2019
2020
2021
2022
Year
Decision
Elected to bat
Elected to bowl

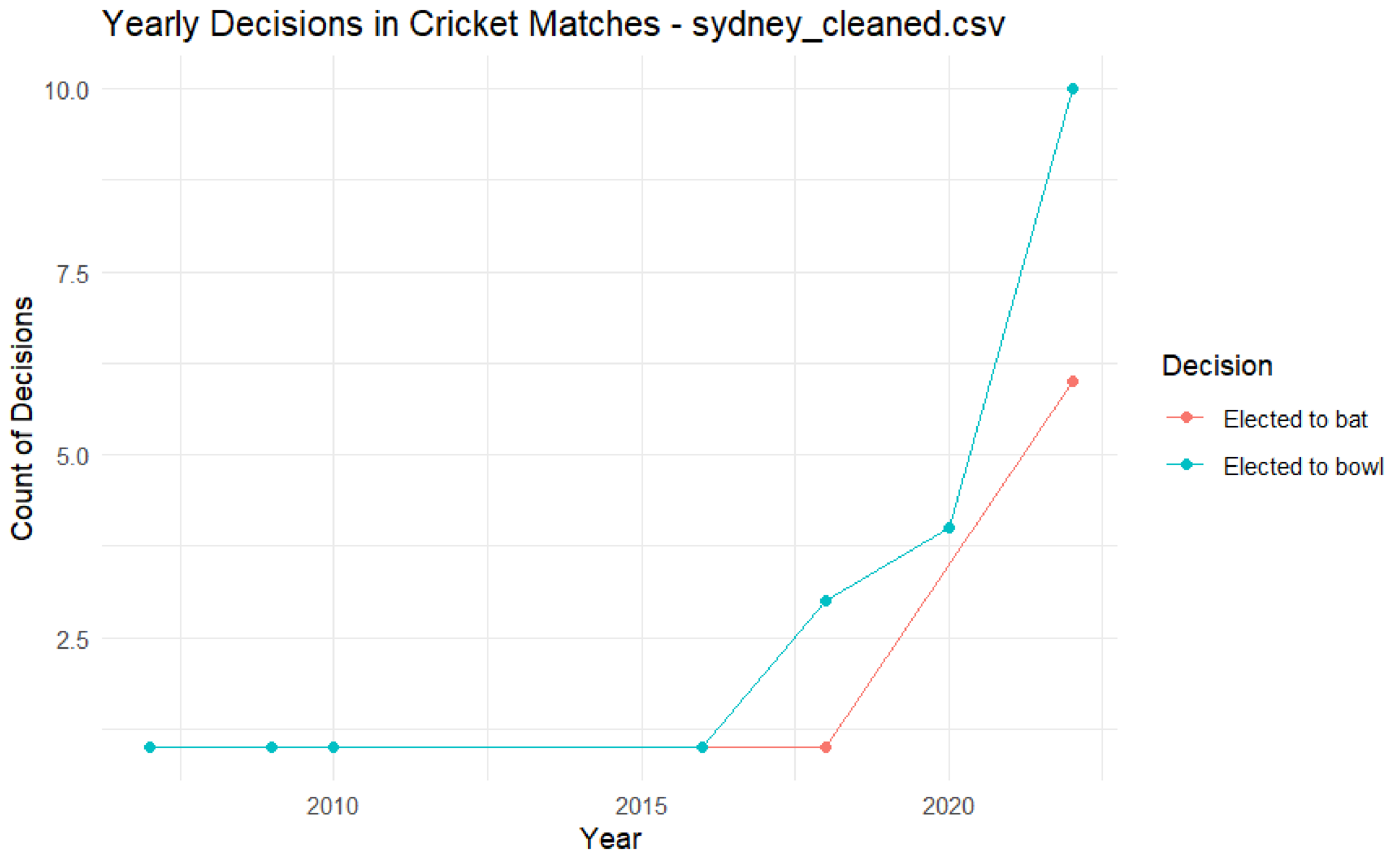

Yearly Decisions in Cricket Matches - adilade_cleaned.csv
Count of Decisions
12.5
10.0
7.5
5.0
2.5
2012.5
2015.0
2017.5
2020.0
2022.5
Year
Decision
Elected to bat
Elected to bowl


Yearly Decisions in Cricket Matches - brisbane_cleaned.csv
6
4
2
Count of Decisions
2010
2015
2020
Year
Decision
Elected to bat
Elected to bowl


Yearly Decisions in Cricket Matches - geelong_cleaned.csv
8
6
4
2
Count of Decisions
2017
2018
2019
2020
2021
2022
Year
Decision
Elected to bat
Elected to bowl


Yearly Decisions in Cricket Matches - melbourne_cleaned.csv
10.0
7.5
5.0
2.5
Count of Decisions
2008
2012
2016
2020
Year
Decision
Elected to bat
Elected to bowl


Yearly Decisions in Cricket Matches - hobart_cleaned.csv
9
6
3
Count of Decisions
2010.0
2012.5
2015.0
2017.5
2020.0
2022.5
Year
Decision
Elected to bat
Elected to bowl



Successful decisons made on each ground

Hide

```
# Load necessary libraries
library(dplyr)
library(lubridate)

# Define the paths to the CSV files
file_paths <- c("perth_cleaned.csv", "adilade_cleaned.csv", "brisbane_cleaned.csv",
                "geelong_cleaned.csv", "melbourne_cleaned.csv", "hobart_cleaned.csv",
                "sydney_cleaned.csv")

# Initialize an empty list to store the dataframes
results <- list()

# Loop through each file
for (i in 1:length(file_paths)) {
  file_path <- file_paths[i]

  # Read the CSV file
  data <- read.csv(file_path)

  # Extract ground name from file name
  ground_name <- gsub("_cleaned.csv", "", basename(file_path))

  # Adding the 'Decision Maker' and 'Match Winner' columns
  data <- data %>%
    mutate(Decision.Maker = ifelse(Toss == "lost", "Team2", "Team1"),
           Match.Winner = ifelse(Result == "won", "Team1", "Team2"),
           Success = Decision.Maker == Match.Winner)

  # Ensure that 'Date' is in Date format
  data$Date <- dmy(data$Date)

  # Extract year from Date
  data$Year <- year(data$Date)

  # Calculate the total number of decisions and successful decisions per year
  yearly_stats <- data %>%
    group_by(Year) %>%
    summarise(Total.Decisions = n(),
              Successful.Decisions = sum(Success, na.rm = TRUE))

  # Calculate the success ratio
  yearly_stats$Success.Ratio <- (yearly_stats$Successful.Decisions / yearly_stats$Total.Decision
s) * 100

  # Add the ground name to the results table
  yearly_stats$Ground <- ground_name

  # Store the result in the list with ground name as key
  results[[ground_name]] <- yearly_stats
}

# Display the results with ground name included in each table
```

```
for (ground in names(results)) {
  print(results[[ground]])
}
```

Year	Total.Decisions	Successful.Decisions	Success.Ratio	Ground
<dbl>	<int>	<int>	<dbl>	<chr>
2019	2	2	100	perth
2022	12	6	50	perth

2 rows

Year	Total.Decisions	Successful.Decisions	Success.Ratio	Ground
<dbl>	<int>	<int>	<dbl>	<chr>
2011	2	0	0.00000	adilade
2014	2	0	0.00000	adilade
2016	2	0	0.00000	adilade
2017	2	0	0.00000	adilade
2019	2	0	0.00000	adilade
2022	14	2	14.28571	adilade

6 rows

Year	Total.Decisions	Successful.Decisions	Success.Ratio	Ground
<dbl>	<int>	<int>	<dbl>	<chr>
2006	2	2	100	brisbane
2009	2	0	0	brisbane
2013	2	2	100	brisbane
2018	2	0	0	brisbane
2019	2	0	0	brisbane
2022	10	4	40	brisbane

6 rows

Year	Total.Decisions	Successful.Decisions	Success.Ratio	Ground
<dbl>	<int>	<int>	<dbl>	<chr>
2017	2	2	100.00000	geelong
2022	12	4	33.33333	geelong

2 rows

Year	Total.Decisions	Successful.Decisions	Success.Ratio	Ground
<dbl>	<int>	<int>	<dbl>	<chr>
2008	2	0	0.00000	melbourne
2009	2	2	100.00000	melbourne
2010	2	2	100.00000	melbourne
2011	2	2	100.00000	melbourne
2012	2	0	0.00000	melbourne
2013	2	0	0.00000	melbourne
2014	4	0	0.00000	melbourne
2016	2	0	0.00000	melbourne
2017	2	2	100.00000	melbourne
2018	2	2	100.00000	melbourne

1-10 of 12 rows Previous **1** 2 Next

Year	Total.Decisions	Successful.Decisions	Success.Ratio	Ground
<dbl>	<int>	<int>	<dbl>	<chr>
2010	2	2	100.0	hobart
2014	2	2	100.0	hobart
2018	2	2	100.0	hobart
2022	16	2	12.5	hobart

4 rows

Year	Total.Decisions	Successful.Decisions	Success.Ratio	Ground
<dbl>	<int>	<int>	<dbl>	<chr>
2007	2	2	100	sydney
2009	2	2	100	sydney
2010	2	0	0	sydney
2016	2	0	0	sydney
2018	4	2	50	sydney
2020	4	2	50	sydney
2022	16	8	50	sydney

7 rows

Hide

```
NA
NA
```