

# Standings of the NHL regular season 2025–2026 using R

Kimmo Vehkalahti, Senior University Lecturer, PhD

December 2025 — January 2026

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Libraries for data wrangling and visualization</b>	<b>2</b>
<b>3</b>	<b>Functions for arranging the tables of standings</b>	<b>2</b>
<b>4</b>	<b>Functions for displaying the tables of standings</b>	<b>4</b>
<b>5</b>	<b>Build the NHL data sets of teams and games</b>	<b>9</b>
<b>6</b>	<b>Compute cumulative points and standings</b>	<b>11</b>
<b>7</b>	<b>Add status flags to be joined into the tables</b>	<b>12</b>
<b>8</b>	<b>Render various tables of standings</b>	<b>14</b>
<b>9</b>	<b>Design and display a graph of standings</b>	<b>20</b>
<b>10</b>	<b>Display the code for rendering the tables of standings</b>	<b>23</b>

## 1 Introduction

This report uses R (R Core Team, 2026) to wrangle data and present tables and figures on the NHL 2025–2026 regular-season standings. Team and game data are retrieved with the `nhlscraper` package (Saijo, 2026).

The project had two goals: (1) to follow the NHL freely from a social data scientist’s perspective, and (2) to practice R coding. It began in November 2025 with a hand-drawn sketch of a cumulative standings chart. Access to the right data—thanks to `nhlscraper`—made it feasible. With support from *CurreChat* (see below) for code generation and debugging, the work progressed quickly. I then realized it would serve as material for my upcoming course “Storytelling with Data and Infovis,” starting in autumn 2026, when a new NHL season begins—and the current one will be history, rich with stories for students to explore.

### References:

R Core Team (2026). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

Saijo, Rento (2026). `nhlscraper`: Scraper for National Hockey League Data. R package version 0.4.2.9000, <https://rentosaijo.github.io/nhlscraper/>.

*CurreChat* is a generative AI tool for students and staff at the University of Helsinki. It runs OpenAI’s large language models hosted in the University’s Microsoft Azure EU cloud. Data entered into the service is not used to further train the models, and no identifying user information is sent to the model.

## 2 Libraries for data wrangling and visualization

```
library(rlang)
library(tidyverse)
library(lubridate)
library(kableExtra)

library(ggplot2)
library(patchwork)
library(cowplot)
library(MetBrewer)
```

## 3 Functions for arranging the tables of standings

*These functions implement the following tie-breaking procedure (source: [www.nhl.com/standings](http://www.nhl.com/standings)):*

If two or more clubs are tied in points during the regular season, the standing of the clubs is determined in the following order:

1. The fewer number of games played (i.e., superior points percentage).
2. The greater number of games won, excluding games won in Overtime or by Shootout (i.e., ‘Regulation Wins’). This figure is reflected in the RW column.
3. The greater number of games won, excluding games won by Shootout. This figure is reflected in the ROW column.
4. The greater number of games won by the Club in any manner (i.e., ‘Total Wins’). This figure is reflected in the W column.
5. The greater number of points earned in games against each other among two or more tied clubs. For the purpose of determining standing for two or more Clubs that have not played an even number of games with one or more of the other tied Clubs, the first game played in the city that has the extra game (the ‘odd game’) shall not be included. When more than two Clubs are tied, the percentage of available points earned in games among each other (and not including any ‘odd games’) shall be used to determine standing.
6. The greater differential between goals for and against (including goals scored in Overtime or awarded for prevailing in Shootouts) for the entire regular season. This figure is reflected in the DIFF column.
7. The greater number of goals scored (including goals scored in Overtime or awarded for prevailing in Shootouts) for the entire regular season. This figure is reflected in the GF column.

NOTE: In standings a victory in a shootout counts as one goal for, while a shootout loss counts as one goal against.

```
# Head-to-head points/percentage with odd-game rule (rule #5 above)
#   games_long: long-format games data (one row per team per game), as defined
#   tied_ids: integer vector of TeamId values that are tied on previous criteria
# Returns a tibble TeamId, H2H_Pct among tied_ids following NHL odd-game rule

compute_h2h_pct <- function(games_long, tied_ids) {
  mutual <- games_long %>%
    filter(TeamId %in% tied_ids, OppTeamId %in% tied_ids) %>%
    select(id, gameDate, TeamId, OppTeamId, side, Points, Code)

  # We need to exclude "odd games": for any pair of clubs (A,B),
  # if one city hosted an extra game, drop the first game in that city.
  # Work at game-level; then rebuild team rows.

  # Get distinct game-level info with home/away codes to detect city (venue) hosting
  game_info <- mutual %>%
    group_by(id) %>%
    summarise(
      gameDate = first(gameDate),
      homeId   = TeamId[side == "home"][1],
```

```

    awayId    = TeamId[side == "away"][1],
    .groups   = "drop"
  ) %>%
  # Only keep games where both teams are in tied_ids (already ensured)
  mutate(pair_key = paste(pmin(homeId, awayId), pmax(homeId, awayId), sep = "-"))

# For each pair_key, count games by venue (home team city)
# i.e., count by venue (homeId) within pair
venue_counts <- game_info %>%
  group_by(pair_key, homeId) %>%
  arrange(gameDate, .by_group = TRUE) %>%
  summarise(
    n_city_games = n(),
    first_game_id_in_city = first(id),
    .groups = "drop"
  )

# identify odd venue cases and drop first game in the city with extra game
odd_games_to_drop <- venue_counts %>%
  group_by(pair_key) %>%
  # Determine if any homeId has greater n_city_games than the minimum; drop one earliest for those
  mutate(min_city = min(n_city_games)) %>%
  filter(n_city_games > min_city) %>%
  # NHL says: "the first game played in the city that has the extra game" is excluded
  transmute(pair_key, drop_game_id = first_game_id_in_city) %>%
  distinct()

# Apply the exclusion at game level
game_info_filtered <- game_info %>%
  anti_join(odd_games_to_drop, by = c("pair_key", "id" = "drop_game_id"))

# Rebuild the mutual rows from filtered games
mutual_filtered <- mutual %>%
  semi_join(game_info_filtered %>% select(id), by = "id")

# Now compute points earned vs tied group and percentage of available points
# i.e., summarize per team
h2h <- mutual_filtered %>%
  group_by(TeamId) %>%
  summarise(
    H2H_Games    = n(),                # team-rows; each game contributes 1 for the team
    H2H_Pts      = sum(Points),        # points earned in these mutual games
    H2H_Pct      = ifelse(H2H_Games > 0, round(H2H_Pts / (1 * H2H_Games), 3), NA_real_),
    # Available points for a team are 2 per game; percentage of available points = H2H_Pts / (2 * games)
    H2H_PctAvail = ifelse(H2H_Games > 0, round(H2H_Pts / (2 * H2H_Games), 3), NA_real_),
    .groups = "drop"
  )

# NHL for >2 clubs: "percentage of available points" among each other; for 2 clubs, points vs each other.
# We'll provide both; ordering function will pick H2H_PctAvail for groups of size >=3, else H2H_Pct.
h2h
}

# A stable tie-break ordering function applying 1-7 with #5 included. Resolves ties iteratively:

```

```

# after each criterion, only reorders within equal groups using the next criterion.
nhl_order_with_h2h <- function(df, games_long) {
  # 1) Initial order
  base_ordered <- df %>%
    arrange(desc(Pts), desc(Pct), desc(RW), desc(ROW), desc(W)) %>%
    # save the original row index for preserving the order
    mutate(.row_base = row_number(),
           group_key = paste(Pts, Pct, RW, ROW, W, sep = "|"))

  # 2) Within group order H2H + DIFF + GF
  refined <- base_ordered %>%
    group_by(group_key) %>%
    group_modify(function(grp, key) {
      if (nrow(grp) <= 1) return(grp)

      tied_ids <- grp$TeamId
      h2h <- compute_h2h_pct(games_long, tied_ids)
      metric_col <- if (length(tied_ids) >= 3) "H2H_PctAvail" else "H2H_Pct"

      grp %>%
        left_join(h2h, by = "TeamId") %>%
        mutate(
          H2H_Pct = coalesce(H2H_Pct, 0),
          H2H_PctAvail = coalesce(H2H_PctAvail, 0)
        ) %>%
        arrange(desc(.data[[metric_col]]), desc(DIFF), desc(GF))
    }) %>%
    ungroup()

  # 3) Preserve the between groups order based on the initial order
  refined %>%
    arrange(.row_base) %>%
    select(-any_of(c("group_key", ".row_base")))
}

# Wrapper to rank after ordering
order_and_rank <- function(df, rank_col, games_long) {
  rank_col <- rlang::ensym(rank_col)
  df %>%
    nhl_order_with_h2h(games_long = games_long) %>%
    mutate(!rank_col := row_number())
}

```

## 4 Functions for displaying the tables of standings

```

# Show point percentage col with 3 decimals
fmt_pct <- function(x) sprintf("%.3f", x)

# DIFF col formatter for LaTeX: use math mode for proper minus & plus
fmt_diff_math <- function(x) {
  ifelse(
    x > 0, paste0("$+", x, "$"),

```

```

    ifelse(x < 0, paste0("$-", abs(x), "$"), "$0$")
  )
}

# Pretty minus helper (Unicode minus for HTML)
unicode_minus <- function(x) {
  sub("^-", "\u2212", x)
}

# DIFF col formatter for HTML: explicit plus, Unicode minus
fmt_diff_plus_html <- function(x) {
  ifelse(
    x > 0, paste0("+", x),
    ifelse(x < 0, unicode_minus(as.character(x)), "0")
  )
}

# Overall title for tables and figures, with English dates

lct <- Sys.getlocale("LC_TIME"); # save the one in use
Sys.setlocale("LC_TIME", "en")

## [1] "en"

overall_title <- sprintf("NHL Standings as %s", format(Sys.Date(), "%B %d, %Y"))
Sys.setlocale("LC_TIME", lct) # restore the earlier one

## [1] "Finnish_Finland.utf8"

# Legends for the marks and col names below the tables

standings_legend_title <- function() {
  paste("Table symbols and abbreviation definitions:")
}

standings_legend_latex <- function() {
  paste(
    "p = Presidents' Trophy (Winner of the Regular Season);",
    "z = clinched conference;",
    "y = clinched division;",
    "x = clinched playoff spot;",
    "e = eliminated from playoffs;",
    "GP = Games Played (W + L + OT);",
    "W = Wins (2pts);",
    "L = Regulation Losses (0pts);",
    "OT = Overtime/Shootout Losses (1pt);",
    "PTS = Points (W $\\times$ 2 + OT);", # obs! 4 backslashes needed :), also below!
    "P\\% = Points Percentage = PTS / (2 $\\times$ GP);",
    "RW = Regulation Wins;",
    "ROW = Regulation + Overtime Wins;",
    "GF = Goals For;",
    "GA = Goals Against;",
    "DIFF = Goal Differential (GF -- GA)."
  )
}

```

```

standings_legend_html <- function() {
  paste(
    "p = Presidents' Trophy (Winner of the Regular Season);",
    "z = clinched conference;",
    "y = clinched division;",
    "x = clinched playoff spot;",
    "e = eliminated from playoffs;",
    "GP = Games Played (W + L + OT);",
    "W = Wins (2pts);",
    "L = Regulation Losses (0pts);",
    "OT = Overtime/Shootout Losses (1pt);",
    "PTS = Points (W × 2 + OT);",
    "P% = Points Percentage = PTS / (2 × GP);",
    "RW = Regulation Wins;",
    "ROW = Regulation + Overtime Wins;",
    "GF = Goals For;",
    "GA = Goals Against;",
    "DIFF = Goal Differential (GF − GA).\"
  )
}

# Attach status mark to Code (optional)
attach_mark <- function(df, status_df = NULL) {
  if (is.null(status_df) || !"TeamId" %in% names(df)) return(df)
  df %>%
    left_join(status_df %>% select(TeamId, Mark), by = "TeamId") %>%
    mutate(Code = if_else(!is.na(Mark) & Mark != "", paste0(Code, " ", Mark), Code)) %>%
    select(-Mark)
}

# Add display aliases for PTS and P% (LaTeX-safe column name for P%)
p_col_name <- function() if (knitr::is_latex_output()) "P\\%" else "P%"

# Shorten also the TeamName to just Team here
add_display_aliases <- function(df) {
  df %>%
    mutate(
      PTS = Pts,
      Team = TeamName,
      !!p_col_name() := Pct
    )
}

# Final column order for league/conference/division tables
num_cols_all <- c("GP", "W", "L", "OT", "PTS", p_col_name(), "RW", "ROW", "GF", "GA", "DIFF")

text_cols_league <- c("Code", "Team", "Conference", "Division")
text_widths_league <- c(30, 95, 40, 70)
final_cols_league <- c(text_cols_league, num_cols_all)

text_cols_conf <- c("Code", "Team", "Division")
text_widths_conf <- c(30, 95, 70)
final_cols_conf <- c(text_cols_conf, num_cols_all)

```

```

text_cols_div <- c("Code","Team","Conference")
text_widths_div <- c(30, 95, 40)
final_cols_div <- c(text_cols_div, num_cols_all)

# Generic render helper: attach marks optionally, add metrics, select columns, display
render_standings <- function(df,
                             cols,
                             text_cols,
                             text_widths_pt,
                             status_df = NULL,
                             caption,
                             label,
                             add_legend = TRUE) {
  df %>%
    attach_mark(status_df) %>%
    add_display_aliases() %>%
    select(all_of(cols)) %>%
    display_table(caption = caption, label = label, add_legend = add_legend)
}

# Set aligns (right/left) for columns (no center aligns)
infer_align <- function(df) {
  types <- vapply(df, function(x) class(x)[1], character(1))
  aligns <- ifelse(types %in% c("integer", "numeric", "double"), "r", "l")
  paste(aligns, collapse = "")
}

# Function for displaying NHL tables in LaTeX and HTML formats
display_table <- function(x,
                          align = NULL,
                          escape = TRUE,
                          label = NULL,
                          font_size = 9,
                          text_cols = NULL,
                          text_widths_pt = NULL,
                          add_legend = TRUE,
                          ...) {
  align_str <- if (is.null(align))
    infer_align(x)
  else {
    if (length(align) == 1) align else paste(align, collapse = "")
  }

##### LaTeX tables #####

  if (knitr::is_latex_output()) {
    x_print <- x %>%
      mutate(
        across(any_of("DIFF"), fmt_diff_math),
        across(any_of(c("Pct", "P%", "P\\%")), fmt_pct)
      )
  }

```

```

kb <- knitr::kable(
  x_print,
  format = "latex",
  booktabs = TRUE,
  longtable = FALSE,
  align = align_str,
  escape = FALSE,          # allow $...$ for math mode (DIFF)
  linesep = "",
  label = label,
  ...
) %>%
  kableExtra::kable_styling(
    latex_options = c("striped"),
    full_width = FALSE,
    font_size = 9
  )

# Widths of text columns
txt_idx <- which(names(x_print) %in% text_cols)
for (i in seq_along(txt_idx)) {
  kb <- kb %>% kableExtra::column_spec(txt_idx[i], width = paste0(text_widths_pt[i], "pt"))
}

if (add_legend) {
  kb <- kb %>%
    kableExtra::footnote(
      general = standings_legend_latex(),
      general_title = standings_legend_title(),
      threeparttable = TRUE,
      escape = FALSE
    )
}
kb

```

##### HTML tables #####

```

} else {
  x_print <- x %>%
    mutate(
      across(any_of("DIFF"), fmt_diff_plus_html),
      across(any_of(c("Pct", "P%", "P\\\\")), fmt_pct)
    )

kb <- knitr::kable(
  x_print,
  format = "html",
  align = align_str,
  escape = TRUE,
  label = label,
  ...
) %>%
  kableExtra::kable_styling(
    full_width = FALSE,

```



```

    font_size = 12
  )

  if (add_legend) {
    kb <- kb %>%
      kableExtra::footnote(
        general = standings_legend_html(),
        general_title = standings_legend_title(),
        escape = TRUE
      )
  }
  kb
}
}

```

## 5 Build the NHL data sets of teams and games

```

library(nhlscraper) # this is everything - see the citation in the beginning!
all_games <- games() # scores of ALL the NHL games since 1917
all_teams <- teams() # including ALL the teams of the history

games <- all_games %>%
  filter(season == "20252026") %>% # regular season: 32*82/2 = 1312 games
  filter(gameType == 2) %>% # regular season games (no pre-season or playoff)
  filter(gameStateId == 7) # only those games that have been played already

TeamIds <- games %>%
  group_by(homeTeamId) %>% summarise() %>% rename(teamId = homeTeamId)

teams <- all_teams %>% filter(id %in% TeamIds$teamId) %>% rename(teamId = id)

lookup <- tibble(
  triCode = c(
    "CAR", "CBJ", "NJD", "NYI", "NYR", "PHI", "PIT", "WSH", # Metropolitan (E)
    "BOS", "BUF", "DET", "FLA", "MTL", "OTT", "TBL", "TOR", # Atlantic (E)
    "ANA", "CGY", "EDM", "LAK", "SJS", "SEA", "VAN", "VGK", # Pacific (W)
    "CHI", "COL", "DAL", "MIN", "NSH", "STL", "UTA", "WPG" # Central (W)
  ), Conference = c(
    rep("Eastern", 16), rep("Western", 16)
  ), Division = c(
    rep("Metropolitan", 8), rep("Atlantic", 8),
    rep("Pacific", 8), rep("Central", 8)
  )
)

teams <- teams %>%
  left_join(lookup, by = "triCode") %>%
  select(-franchiseId, -leagueId, -rawTricode)

games_enriched <- games %>%
  left_join(teams, by = c("homeTeamId" = "teamId")) %>%
  rename(

```

```

    triCode_home      = triCode,
    fullName_home     = fullName,
    Division_home     = Division,
    Conference_home   = Conference
  ) %>%
  left_join(teams, by = c("visitingTeamId" = "teamId")) %>%
  rename(
    triCode_away      = triCode,
    fullName_away     = fullName,
    Division_away     = Division,
    Conference_away   = Conference
  ) %>%
  mutate(
    homeTeamCode      = triCode_home,
    visitingTeamCode  = triCode_away,
    homeTeamName      = fullName_home,
    visitingTeamName  = fullName_away,
    homeTeamDivision  = Division_home,
    visitingTeamDivision = Division_away,
    homeTeamConference = Conference_home,
    visitingTeamConference = Conference_away
  )

# Select only those columns that will be needed
games_played <- games_enriched %>%
  select(gameId,
    gameDate,      # "YYYY-MM-DD" (needed for plotting)
    homeTeamId,
    homeTeamCode,
    homeTeamName,
    homeTeamDivision,
    homeTeamConference,
    visitingTeamId,
    visitingTeamCode,
    visitingTeamName,
    visitingTeamDivision,
    visitingTeamConference,
    homeScore,
    visitingScore,
    period # game finished in 3: regular time / 4: overtime / 5: shootout
  )

# Calculate the points based on the scores
games_played <- games_played %>%
  mutate(
    homeTeamPoints = case_when(
      homeScore > visitingScore & period == 3 ~ 2L, # regulation win
      homeScore > visitingScore & period > 3 ~ 2L, # OT/SO win
      homeScore < visitingScore & period == 3 ~ 0L, # regulation loss
      homeScore < visitingScore & period > 3 ~ 1L  # OT/SO loss
    ),
    visitingTeamPoints = case_when(
      visitingScore > homeScore & period == 3 ~ 2L, # regulation win
      visitingScore > homeScore & period > 3 ~ 2L, # OT/SO win

```

```

    visitingScore <- homeScore & period == 3 ~ OL, # regulation loss
    visitingScore <- homeScore & period > 3 ~ 1L # OT/SO loss
  )
)

```

## 6 Compute cumulative points and standings

```

games_played <- games_played %>%
  mutate(gameDate = lubridate::ymd(gameDate)) # "YYYY-MM-DD"

home_long <- games_played %>%
  transmute(
    gameId, gameDate,
    TeamId      = homeTeamId,
    Code        = homeTeamCode,
    TeamName    = homeTeamName,
    Division    = homeTeamDivision,
    Conference  = homeTeamConference,
    Points      = homeTeamPoints,
    side        = "home",
    OppTeamId   = visitingTeamId,
    OppTeamCode = visitingTeamCode,
    OppTeamName = visitingTeamName,
    GF_total    = homeScore,
    GA_total    = visitingScore,
    period      = period,
    Win         = as.integer(homeScore > visitingScore),
    RegWin      = as.integer(homeScore > visitingScore & period == 3),
    OtWin       = as.integer(homeScore > visitingScore & period == 4),
    SoWin       = as.integer(homeScore > visitingScore & period == 5),
    RegOrOtWin  = as.integer(homeScore > visitingScore & period != 5),
  )

away_long <- games_played %>%
  transmute(
    gameId, gameDate,
    TeamId      = visitingTeamId,
    Code        = visitingTeamCode,
    TeamName    = visitingTeamName,
    Division    = visitingTeamDivision,
    Conference  = visitingTeamConference,
    Points      = visitingTeamPoints,
    side        = "away",
    OppTeamId   = homeTeamId,
    OppTeamCode = homeTeamCode,
    OppTeamName = homeTeamName,
    GF_total    = visitingScore,
    GA_total    = homeScore,
    period      = period,
    Win         = as.integer(visitingScore > homeScore),
    RegWin      = as.integer(visitingScore > homeScore & period == 3),
    OtWin       = as.integer(visitingScore > homeScore & period == 4),
    SoWin       = as.integer(visitingScore > homeScore & period == 5),
  )

```

```

    RegOrOtWin    = as.integer(visitingScore > homeScore & period != 5),
  )
games_long <- bind_rows(home_long, away_long)

team_stats <- games_long %>%
  group_by(TeamId, Code, TeamName, Conference, Division) %>%
  summarise(
    GP      = n(), # note: GP = W + L + OT, where W = RW + ROW (see below)
    Pts     = sum(Points),
    Pct     = round(Pts / (GP * 2), digits = 3), # NHL points percentage
    W       = sum(Win),
    L       = sum(GA_total > GF_total & period == 3), # Regulation Loss
    OT      = sum(GA_total > GF_total & period > 3), # OT/SO Loss
    RW      = sum(RegWin),
    ROW     = sum(RegOrOtWin),
    GF      = sum(GF_total),
    GA      = sum(GA_total),
    DIFF    = GF - GA,
    .groups = "drop"
  )

east_ordered <- team_stats %>% filter(Conference == "Eastern") %>% order_and_rank(ConfRank, games_long)
west_ordered <- team_stats %>% filter(Conference == "Western") %>% order_and_rank(ConfRank, games_long)

division_tables <- list(
  Atlantic      = team_stats %>% filter(Division == "Atlantic") %>% order_and_rank(DivRank, games_long),
  Metropolitan  = team_stats %>% filter(Division == "Metropolitan") %>% order_and_rank(DivRank, games_long),
  Central       = team_stats %>% filter(Division == "Central") %>% order_and_rank(DivRank, games_long),
  Pacific       = team_stats %>% filter(Division == "Pacific") %>% order_and_rank(DivRank, games_long)
)

auto_east <- bind_rows(division_tables[c("Atlantic", "Metropolitan")]) %>% filter(DivRank <= 3)
auto_west <- bind_rows(division_tables[c("Central", "Pacific")]) %>% filter(DivRank <= 3)

east_remaining <- east_ordered %>% anti_join(select(auto_east, TeamId), by = "TeamId")
west_remaining <- west_ordered %>% anti_join(select(auto_west, TeamId), by = "TeamId")

east_wildcards <- east_remaining %>% arrange(ConfRank) %>% slice_head(n = 2)
west_wildcards <- west_remaining %>% arrange(ConfRank) %>% slice_head(n = 2)

wildcard_codes <- bind_rows(east_wildcards, west_wildcards) %>% pull(Code)

```

## 7 Add status flags to be joined into the tables

- compute status flags (p, z, y, x, e) based on points and maximum attainable points
- join these flags into tables by modifying the Code with the highest-priority mark
- the possible marks (in the priority order) are as follows:
  - p = Presidents' Trophy (Winner of the Regular Season)
  - z = clinched conference (*see below*)
  - y = clinched division (*see below*)
  - x = clinched playoff spot (*see below*)
  - e = eliminated from playoffs (*see below*)

NOTE: In the NHL, to “clinch” means a team has mathematically secured a spot in the Stanley Cup Playoffs, a division title, or a top seed, making it impossible for them to miss that position regardless of the outcome of their remaining games. A team clinches when their total points or win percentage guarantee a berth (a secured spot), even if they lose all remaining games. Key details:

- Playoff Berth: Securing a spot in the top 3 of a division or one of the two wild-card spots per conference.
- Division/Conference Title: Guaranteeing the highest rank in their division or conference, often securing home-ice advantage.
- “Magic Number”: A, often unofficial, number of points a team needs to gain (or opposing teams need to lose) to clinch.
- Alternative Clinch: A team can clinch a playoff spot on a day they do not play, if another team’s loss prevents them from catching up.
- Significance: It signifies that a team has successfully qualified for the post-season, eliminating the possibility of finishing outside the top 16 positions.

(Source: Google AI Overview)

```
# NOTE: This is a practical approximation; it does not fully simulate NHL tie-break scenarios.
```

```
# Extend team stats with remaining games and maximum points
```

```
team_ext <- team_stats %>%  
  mutate(  
    GP_remaining = 82L - GP,  
    MaxPts       = Pts + 2L * GP_remaining  
  )
```

```
# p, z, y flags: compare Pts to "others' max MaxPts"
```

```
best_flags <- team_ext %>%  
  rowwise() %>%  
  mutate(  
    p_flag = Pts >= max(team_ext$MaxPts[team_ext$TeamId != TeamId]),  
    z_flag = Pts >= max(team_ext$MaxPts[team_ext$Conference == Conference & team_ext$TeamId != TeamId]),  
    y_flag = Pts >= max(team_ext$MaxPts[team_ext$Division == Division & team_ext$TeamId != TeamId])  
  ) %>%  
  ungroup() %>%  
  select(TeamId, Code, Pts, MaxPts, Conference, Division, p_flag, z_flag, y_flag)
```

```
# e flag: eliminated if max attainable < current 8th-place Pts in conference
```

```
east_ord <- east_ordered %>% select(TeamId, Code, Pts, Conference)  
west_ord <- west_ordered %>% select(TeamId, Code, Pts, Conference)  
  
east_top8_pts <- east_ord %>% arrange(desc(Pts)) %>% slice(8) %>% pull(Pts)  
west_top8_pts <- west_ord %>% arrange(desc(Pts)) %>% slice(8) %>% pull(Pts)
```

```
conf_status <- team_ext %>%  
  select(TeamId, Code, Conference, Pts, MaxPts) %>%  
  mutate(conf_top8_pts = if_else(Conference == "Eastern", east_top8_pts, west_top8_pts),  
         e_flag       = MaxPts < conf_top8_pts)
```

```
# x flag: clinched playoff spot if Pts >= max MaxPts among non-top-8 teams in conference
```

```
east_not_top8_max <- east_ord %>%  
  arrange(desc(Pts)) %>%  
  slice(9:n()) %>%  
  left_join(team_ext %>% select(TeamId, MaxPts), by = "TeamId") %>%  
  summarise(max_not8_maxpts = max(MaxPts)) %>%  
  pull(max_not8_maxpts)
```

```
west_not_top8_max <- west_ord %>%  
  arrange(desc(Pts)) %>%
```

```

slice(9:n()) %>%
left_join(team_ext %>% select(TeamId, MaxPts), by = "TeamId") %>%
summarise(max_not8_maxpts = max(MaxPts)) %>%
pull(max_not8_maxpts)

conf_status <- conf_status %>%
  mutate(
    x_flag = case_when(
      Conference == "Eastern" ~ Pts >= coalesce(east_not_top8_max, -Inf),
      Conference == "Western" ~ Pts >= coalesce(west_not_top8_max, -Inf),
      TRUE ~ FALSE
    )
  )

status_all <- best_flags %>%
  select(TeamId, p_flag, z_flag, y_flag) %>%
  left_join(conf_status %>% select(TeamId, x_flag, e_flag), by = "TeamId") %>%
  mutate(
    Mark = case_when(
      p_flag ~ "p",
      z_flag ~ "z",
      y_flag ~ "y",
      x_flag ~ "x",
      e_flag ~ "e",
      TRUE ~ ""
    )
  )

```

## 8 Render various tables of standings

Table 1: NHL Standings as February 08, 2026: League

Code	Team	Conference	Division	GP	W	L	OT	PTS	P%	RW	ROW	GF	GA	DIFF
COL	Colorado Avalanche	Western	Central	55	37	9	9	83	0.755	34	36	212	138	+74
TBL	Tampa Bay Lightning	Eastern	Atlantic	55	37	14	4	78	0.709	28	33	199	140	+59
CAR	Carolina Hurricanes	Eastern	Metropolitan	57	36	15	6	78	0.684	26	32	197	163	+34
MIN	Minnesota Wild	Western	Central	58	34	14	10	78	0.672	21	30	196	169	+27
DAL	Dallas Stars	Western	Central	57	34	14	9	77	0.675	28	30	193	160	+33
MTL	Montréal Canadiens	Eastern	Atlantic	57	32	17	8	72	0.632	21	30	199	187	+12
DET	Detroit Red Wings	Eastern	Atlantic	58	33	19	6	72	0.621	23	31	174	175	−1
PIT	Pittsburgh Penguins	Eastern	Metropolitan	56	29	15	12	70	0.625	25	28	192	169	+23
BUF	Buffalo Sabres	Eastern	Atlantic	57	32	19	6	70	0.614	26	29	195	176	+19
BOS	Boston Bruins	Eastern	Atlantic	57	32	20	5	69	0.605	24	30	195	182	+13
NYI	New York Islanders	Eastern	Metropolitan	58	32	21	5	69	0.595	22	28	169	162	+7
VGK	Vegas Golden Knights	Western	Pacific	57	27	16	14	68	0.596	20	26	191	179	+12
CBJ	Columbus Blue Jackets	Eastern	Metropolitan	56	29	20	7	65	0.580	20	24	179	177	+2
WSH	Washington Capitals	Eastern	Metropolitan	59	29	23	7	65	0.551	24	28	189	177	+12
UTA	Utah Mammoth	Western	Central	57	30	23	4	64	0.561	23	30	183	157	+26
EDM	Edmonton Oilers	Western	Pacific	58	28	22	8	64	0.552	21	28	198	194	+4
SEA	Seattle Kraken	Western	Pacific	56	27	20	9	63	0.562	21	26	162	165	−3
ANA	Anaheim Ducks	Western	Pacific	56	30	23	3	63	0.562	18	24	185	195	−10
OTT	Ottawa Senators	Eastern	Atlantic	57	28	22	7	63	0.553	22	25	191	184	+7
TOR	Toronto Maple Leafs	Eastern	Atlantic	57	27	21	9	63	0.553	19	26	188	194	−6
PHI	Philadelphia Flyers	Eastern	Metropolitan	56	25	20	11	61	0.545	18	20	167	180	−13
FLA	Florida Panthers	Eastern	Atlantic	57	29	25	3	61	0.535	23	25	174	191	−17
LAK	Los Angeles Kings	Western	Pacific	56	23	19	14	60	0.536	14	19	146	161	−15
NSH	Nashville Predators	Western	Central	57	26	24	7	59	0.518	19	23	169	198	−29
SJS	San Jose Sharks	Western	Pacific	55	27	24	4	58	0.527	16	25	171	193	−22
NJD	New Jersey Devils	Eastern	Metropolitan	57	28	27	2	58	0.509	18	25	146	175	−29
CHI	Chicago Blackhawks	Western	Central	57	22	26	9	53	0.465	17	19	154	185	−31
WPG	Winnipeg Jets	Western	Central	56	22	26	8	52	0.464	20	21	160	175	−15
CGY	Calgary Flames	Western	Pacific	56	23	27	6	52	0.464	19	21	142	169	−27
NYR	New York Rangers	Eastern	Metropolitan	57	22	29	6	50	0.439	14	20	151	180	−29
STL	St. Louis Blues	Western	Central	57	20	28	9	49	0.430	19	19	147	203	−56
VAN	Vancouver Canucks	Western	Pacific	57	18	33	6	42	0.368	12	14	149	210	−61

*Table symbols and abbreviation definitions:*

p = Presidents' Trophy (Winner of the Regular Season); z = clinched conference; y = clinched division; x = clinched playoff spot; e = eliminated from playoffs; GP = Games Played (W + L + OT); W = Wins (2pts); L = Regulation Losses (0pts); OT = Overtime/Shootout Losses (1pt); PTS = Points (W × 2 + OT); P% = Points Percentage = PTS / (2 × GP); RW = Regulation Wins; ROW = Regulation + Overtime Wins; GF = Goals For; GA = Goals Against; DIFF = Goal Differential (GF − GA).

Table 2: NHL Standings as February 08, 2026: Western Conference

Code	Team	Division	GP	W	L	OT	PTS	P%	RW	ROW	GF	GA	DIFF
COL	Colorado Avalanche	Central	55	37	9	9	83	0.755	34	36	212	138	+74
MIN	Minnesota Wild	Central	58	34	14	10	78	0.672	21	30	196	169	+27
DAL	Dallas Stars	Central	57	34	14	9	77	0.675	28	30	193	160	+33
VGK	Vegas Golden Knights	Pacific	57	27	16	14	68	0.596	20	26	191	179	+12
UTA	Utah Mammoth	Central	57	30	23	4	64	0.561	23	30	183	157	+26
EDM	Edmonton Oilers	Pacific	58	28	22	8	64	0.552	21	28	198	194	+4
SEA	Seattle Kraken	Pacific	56	27	20	9	63	0.562	21	26	162	165	-3
ANA	Anaheim Ducks	Pacific	56	30	23	3	63	0.562	18	24	185	195	-10
LAK	Los Angeles Kings	Pacific	56	23	19	14	60	0.536	14	19	146	161	-15
NSH	Nashville Predators	Central	57	26	24	7	59	0.518	19	23	169	198	-29
SJS	San Jose Sharks	Pacific	55	27	24	4	58	0.527	16	25	171	193	-22
CHI	Chicago Blackhawks	Central	57	22	26	9	53	0.465	17	19	154	185	-31
WPG	Winnipeg Jets	Central	56	22	26	8	52	0.464	20	21	160	175	-15
CGY	Calgary Flames	Pacific	56	23	27	6	52	0.464	19	21	142	169	-27
STL	St. Louis Blues	Central	57	20	28	9	49	0.430	19	19	147	203	-56
VAN	Vancouver Canucks	Pacific	57	18	33	6	42	0.368	12	14	149	210	-61

Table 3: NHL Standings as February 08, 2026: Eastern Conference

Code	Team	Division	GP	W	L	OT	PTS	P%	RW	ROW	GF	GA	DIFF
TBL	Tampa Bay Lightning	Atlantic	55	37	14	4	78	0.709	28	33	199	140	+59
CAR	Carolina Hurricanes	Metropolitan	57	36	15	6	78	0.684	26	32	197	163	+34
MTL	Montréal Canadiens	Atlantic	57	32	17	8	72	0.632	21	30	199	187	+12
DET	Detroit Red Wings	Atlantic	58	33	19	6	72	0.621	23	31	174	175	-1
PIT	Pittsburgh Penguins	Metropolitan	56	29	15	12	70	0.625	25	28	192	169	+23
BUF	Buffalo Sabres	Atlantic	57	32	19	6	70	0.614	26	29	195	176	+19
BOS	Boston Bruins	Atlantic	57	32	20	5	69	0.605	24	30	195	182	+13
NYI	New York Islanders	Metropolitan	58	32	21	5	69	0.595	22	28	169	162	+7
CBJ	Columbus Blue Jackets	Metropolitan	56	29	20	7	65	0.580	20	24	179	177	+2
WSH	Washington Capitals	Metropolitan	59	29	23	7	65	0.551	24	28	189	177	+12
OTT	Ottawa Senators	Atlantic	57	28	22	7	63	0.553	22	25	191	184	+7
TOR	Toronto Maple Leafs	Atlantic	57	27	21	9	63	0.553	19	26	188	194	-6
PHI	Philadelphia Flyers	Metropolitan	56	25	20	11	61	0.545	18	20	167	180	-13
FLA	Florida Panthers	Atlantic	57	29	25	3	61	0.535	23	25	174	191	-17
NJD	New Jersey Devils	Metropolitan	57	28	27	2	58	0.509	18	25	146	175	-29
NYR	New York Rangers	Metropolitan	57	22	29	6	50	0.439	14	20	151	180	-29

*Table symbols and abbreviation definitions:*

p = Presidents' Trophy (Winner of the Regular Season); z = clinched conference; y = clinched division; x = clinched playoff spot; e = eliminated from playoffs; GP = Games Played (W + L + OT); W = Wins (2pts); L = Regulation Losses (0pts); OT = Overtime/Shootout Losses (1pt); PTS = Points (W × 2 + OT); P% = Points Percentage = PTS / (2 × GP); RW = Regulation Wins; ROW = Regulation + Overtime Wins; GF = Goals For; GA = Goals Against; DIFF = Goal Differential (GF - GA).



Table 4: NHL Standings as February 08, 2026: Central Division

Code	Team	Conference	GP	W	L	OT	PTS	P%	RW	ROW	GF	GA	DIFF
COL	Colorado Avalanche	Western	55	37	9	9	83	0.755	34	36	212	138	+74
MIN	Minnesota Wild	Western	58	34	14	10	78	0.672	21	30	196	169	+27
DAL	Dallas Stars	Western	57	34	14	9	77	0.675	28	30	193	160	+33
UTA	Utah Mammoth	Western	57	30	23	4	64	0.561	23	30	183	157	+26
NSH	Nashville Predators	Western	57	26	24	7	59	0.518	19	23	169	198	-29
CHI	Chicago Blackhawks	Western	57	22	26	9	53	0.465	17	19	154	185	-31
WPG	Winnipeg Jets	Western	56	22	26	8	52	0.464	20	21	160	175	-15
STL	St. Louis Blues	Western	57	20	28	9	49	0.430	19	19	147	203	-56

Table 5: NHL Standings as February 08, 2026: Pacific Division

Code	Team	Conference	GP	W	L	OT	PTS	P%	RW	ROW	GF	GA	DIFF
VGK	Vegas Golden Knights	Western	57	27	16	14	68	0.596	20	26	191	179	+12
EDM	Edmonton Oilers	Western	58	28	22	8	64	0.552	21	28	198	194	+4
SEA	Seattle Kraken	Western	56	27	20	9	63	0.562	21	26	162	165	-3
ANA	Anaheim Ducks	Western	56	30	23	3	63	0.562	18	24	185	195	-10
LAK	Los Angeles Kings	Western	56	23	19	14	60	0.536	14	19	146	161	-15
SJS	San Jose Sharks	Western	55	27	24	4	58	0.527	16	25	171	193	-22
CGY	Calgary Flames	Western	56	23	27	6	52	0.464	19	21	142	169	-27
VAN	Vancouver Canucks	Western	57	18	33	6	42	0.368	12	14	149	210	-61

*Table symbols and abbreviation definitions:*

p = Presidents' Trophy (Winner of the Regular Season); z = clinched conference; y = clinched division; x = clinched playoff spot; e = eliminated from playoffs; GP = Games Played (W + L + OT); W = Wins (2pts); L = Regulation Losses (0pts); OT = Overtime/Shootout Losses (1pt); PTS = Points (W  $\times$  2 + OT); P% = Points Percentage = PTS / (2  $\times$  GP); RW = Regulation Wins; ROW = Regulation + Overtime Wins; GF = Goals For; GA = Goals Against; DIFF = Goal Differential (GF - GA).

Table 6: NHL Standings as February 08, 2026: Atlantic Division

Code	Team	Conference	GP	W	L	OT	PTS	P%	RW	ROW	GF	GA	DIFF
TBL	Tampa Bay Lightning	Eastern	55	37	14	4	78	0.709	28	33	199	140	+59
MTL	Montréal Canadiens	Eastern	57	32	17	8	72	0.632	21	30	199	187	+12
DET	Detroit Red Wings	Eastern	58	33	19	6	72	0.621	23	31	174	175	-1
BUF	Buffalo Sabres	Eastern	57	32	19	6	70	0.614	26	29	195	176	+19
BOS	Boston Bruins	Eastern	57	32	20	5	69	0.605	24	30	195	182	+13
OTT	Ottawa Senators	Eastern	57	28	22	7	63	0.553	22	25	191	184	+7
TOR	Toronto Maple Leafs	Eastern	57	27	21	9	63	0.553	19	26	188	194	-6
FLA	Florida Panthers	Eastern	57	29	25	3	61	0.535	23	25	174	191	-17

Table 7: NHL Standings as February 08, 2026: Metropolitan Division

Code	Team	Conference	GP	W	L	OT	PTS	P%	RW	ROW	GF	GA	DIFF
CAR	Carolina Hurricanes	Eastern	57	36	15	6	78	0.684	26	32	197	163	+34
PIT	Pittsburgh Penguins	Eastern	56	29	15	12	70	0.625	25	28	192	169	+23
NYI	New York Islanders	Eastern	58	32	21	5	69	0.595	22	28	169	162	+7
CBJ	Columbus Blue Jackets	Eastern	56	29	20	7	65	0.580	20	24	179	177	+2
WSH	Washington Capitals	Eastern	59	29	23	7	65	0.551	24	28	189	177	+12
PHI	Philadelphia Flyers	Eastern	56	25	20	11	61	0.545	18	20	167	180	-13
NJD	New Jersey Devils	Eastern	57	28	27	2	58	0.509	18	25	146	175	-29
NYR	New York Rangers	Eastern	57	22	29	6	50	0.439	14	20	151	180	-29

*Table symbols and abbreviation definitions:*

p = Presidents' Trophy (Winner of the Regular Season); z = clinched conference; y = clinched division; x = clinched playoff spot; e = eliminated from playoffs; GP = Games Played (W + L + OT); W = Wins (2pts); L = Regulation Losses (0pts); OT = Overtime/Shootout Losses (1pt); PTS = Points (W × 2 + OT); P% = Points Percentage = PTS / (2 × GP); RW = Regulation Wins; ROW = Regulation + Overtime Wins; GF = Goals For; GA = Goals Against; DIFF = Goal Differential (GF - GA).

Table 8: NHL Standings as February 08, 2026: Western Conference, Auto qualifiers (top 3 per division)

Code	Team	Division	GP	W	L	OT	PTS	P%	RW	ROW	GF	GA	DIFF
COL	Colorado Avalanche	Central	55	37	9	9	83	0.755	34	36	212	138	+74
MIN	Minnesota Wild	Central	58	34	14	10	78	0.672	21	30	196	169	+27
DAL	Dallas Stars	Central	57	34	14	9	77	0.675	28	30	193	160	+33
VGK	Vegas Golden Knights	Pacific	57	27	16	14	68	0.596	20	26	191	179	+12
EDM	Edmonton Oilers	Pacific	58	28	22	8	64	0.552	21	28	198	194	+4
SEA	Seattle Kraken	Pacific	56	27	20	9	63	0.562	21	26	162	165	-3

Table 9: NHL Standings as February 08, 2026: Western Conference, Wild cards (top 2 remaining)

Code	Team	Division	GP	W	L	OT	PTS	P%	RW	ROW	GF	GA	DIFF
UTA	Utah Mammoth	Central	57	30	23	4	64	0.561	23	30	183	157	+26
ANA	Anaheim Ducks	Pacific	56	30	23	3	63	0.562	18	24	185	195	-10

Table 10: NHL Standings as February 08, 2026: Eastern Conference, Auto qualifiers (top 3 per division)

Code	Team	Division	GP	W	L	OT	PTS	P%	RW	ROW	GF	GA	DIFF
TBL	Tampa Bay Lightning	Atlantic	55	37	14	4	78	0.709	28	33	199	140	+59
MTL	Montréal Canadiens	Atlantic	57	32	17	8	72	0.632	21	30	199	187	+12
DET	Detroit Red Wings	Atlantic	58	33	19	6	72	0.621	23	31	174	175	-1
CAR	Carolina Hurricanes	Metropolitan	57	36	15	6	78	0.684	26	32	197	163	+34
PIT	Pittsburgh Penguins	Metropolitan	56	29	15	12	70	0.625	25	28	192	169	+23
NYI	New York Islanders	Metropolitan	58	32	21	5	69	0.595	22	28	169	162	+7

Table 11: NHL Standings as February 08, 2026: Eastern Conference, Wild cards (top 2 remaining)

Code	Team	Division	GP	W	L	OT	PTS	P%	RW	ROW	GF	GA	DIFF
BUF	Buffalo Sabres	Atlantic	57	32	19	6	70	0.614	26	29	195	176	+19
BOS	Boston Bruins	Atlantic	57	32	20	5	69	0.605	24	30	195	182	+13

*Table symbols and abbreviation definitions:*

p = Presidents' Trophy (Winner of the Regular Season); z = clinched conference; y = clinched division; x = clinched playoff spot; e = eliminated from playoffs; GP = Games Played (W + L + OT); W = Wins (2pts); L = Regulation Losses (0pts); OT = Overtime/Shootout Losses (1pt); PTS = Points (W × 2 + OT); P% = Points Percentage = PTS / (2 × GP); RW = Regulation Wins; ROW = Regulation + Overtime Wins; GF = Goals For; GA = Goals Against; DIFF = Goal Differential (GF - GA).

## 9 Design and display a graph of standings

- compute the cumulative points
- define functions for plotting and labeling
- draw cumulative line plots for all Divisions
- legends will show the current order of each Division
- mark wild-cards (= would be in playoffs in addition to 3 best of each Division)

```
# Cumulative points dataset
team_points_cum <- games_long %>%
  arrange(Code, gameDate, gameId) %>%
  group_by(Code, Division, Conference) %>%
  mutate(CumPoints = cumsum(Points)) %>%
  ungroup()

# Compute global axes from the full dataset
# (consistent x-axis dates and y-axis values across plots!)
x_min <- min(team_points_cum$gameDate)
x_max <- max(team_points_cum$gameDate)
y_min <- 0
y_max <- max(team_points_cum$CumPoints)
global_y_scale <- scale_y_continuous(
  limits = c(y_min, y_max),
  expand = expansion(mult = c(0.02, 0.05))
)
global_x_scale <- scale_x_date(
  limits = c(x_min, x_max),
  breaks = scales::breaks_width("1 month"),
  labels = scales::label_date(format = "%b %Y", locale = "en"),
  expand = expansion(mult = c(0.01, 0.05))
)

# pick an 8-color palette, e.g., "Signac", "Hiroshige", "Kandinsky"
div_palette <- MetBrewer::met.brewer(name = "Hiroshige", n = 8, type = "discrete")

make_division_plot_with_wildcard_label <- function(df_div, division_name, x_scale, y_scale, colors = NULL) {
  # Teams that appear in this division's plotting data (factor or character)
  teams_in_plot <- levels(df_div$Code)
  if (is.null(teams_in_plot)) {
    teams_in_plot <- unique(df_div$Code)
  }
  # Build labels: append "*" to wild-card teams
  labels_vec <- setNames(teams_in_plot, teams_in_plot)
  labels_vec[labels_vec %in% wildcard_codes] <- paste0(labels_vec[labels_vec %in% wildcard_codes], " *")

  p <- ggplot(df_div, aes(x = gameDate, y = CumPoints, color = Code, group = Code)) +
    geom_line(linewidth = 0.5, alpha = 0.5) +
    geom_point(aes(fill = side), shape = 21, size = 1.5, stroke = 0.5) +
    scale_color_manual(values = div_palette) + # apply MetBrewer palette
    scale_fill_manual(values = c(home = "white", away = "black"), name = "Side", drop = FALSE) +
    labs(title = paste0(division_name, " Division"), x = NULL, y = "Cumulative points", color = "Team") +
    guides(color = guide_legend(order = 1), fill = guide_legend(order = 2)) +
    x_scale + y_scale +
    theme_minimal() + theme(legend.position = "right")
  # Apply colors and custom labels
  if (!is.null(colors)) {

```

```

# If you have a global named color vector, subset to the teams present
colors_sub <- colors[teams_in_plot]
p <- p + scale_color_manual(values = colors_sub, labels = labels_vec)
} else {
  # Keep default colors but override labels
  p <- p + scale_color_discrete(labels = labels_vec)
}
p
}

div_order_names <- c("Atlantic", "Metropolitan", "Central", "Pacific")
plots_list <- lapply(div_order_names, function(d) {
  df_div_points <- dplyr::filter(team_points_cum, Division == d)
  # Align to standings order for legend consistency
  if (!is.null(division_tables[[d]])) {
    div_levels <- division_tables[[d]]$Code
    df_div_points <- df_div_points %>% dplyr::mutate(Code = factor(Code, levels = div_levels))
  }
  make_division_plot_with_wildcard_label(df_div_points, d, global_x_scale, global_y_scale) # [colors]
})
names(plots_list) <- div_order_names

west_stack <- plot_grid(
  plots_list[["Central"]],
  plots_list[["Pacific"]],
  ncol = 1, align = "v", axis = "lr"
)

east_stack <- plot_grid(
  plots_list[["Atlantic"]],
  plots_list[["Metropolitan"]],
  ncol = 1, align = "v", axis = "lr"
)

west_title <- ggdraw() +
  draw_label("Western Conference", fontface = "bold", size = 16, x = 0.5, hjust = 0.5)
east_title <- ggdraw() +
  draw_label("Eastern Conference", fontface = "bold", size = 16, x = 0.5, hjust = 0.5)

west_panel <- plot_grid(
  west_title, west_stack,
  ncol = 1, rel_heights = c(0.10, 1)
)
east_panel <- plot_grid(
  east_title, east_stack,
  ncol = 1, rel_heights = c(0.10, 1)
)

grid_with_titles <- plot_grid(
  west_panel, east_panel,
  ncol = 2, rel_widths = c(1, 1), align = "hv"
)

final_plot <- (ggdraw() + draw_plot(grid_with_titles)) +

```

```

plot_annotation(
  title = overall_title,
  subtitle = "* indicates current wild-card team",
  theme = theme(
    plot.title = element_text(
      size = 24,
      face = "bold",
      hjust = 0.5,      # center
      margin = margin(b = 12)
    ),
    plot.subtitle = element_text(hjust = 0.5)
  )
)

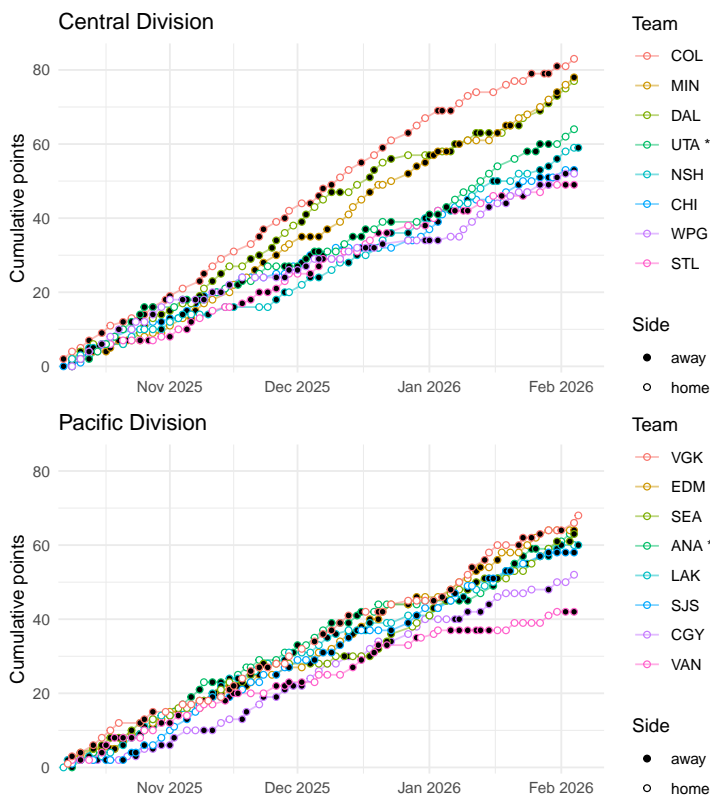
print(final_plot)

```

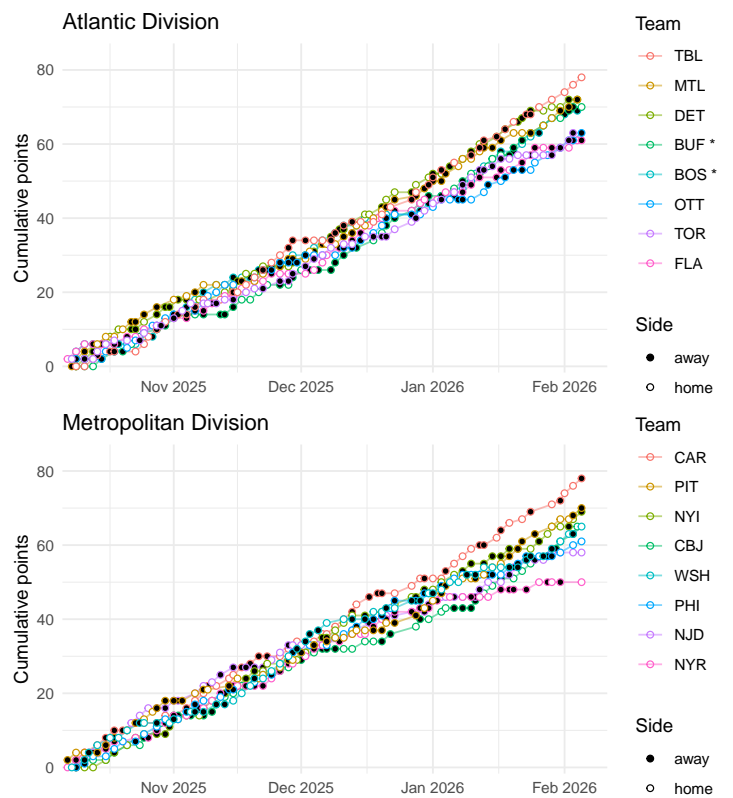
## NHL Standings as February 08, 2026

\* indicates current wild-card team

### Western Conference



### Eastern Conference



```

ggsave(
  filename = paste0(overall_title, ".png"),
  plot = final_plot,
  width = 12, height = 8, dpi = 300, units = "in"
)

```

## 10 Display the code for rendering the tables of standings

```
# League table
tibble(team_stats) %>%
  nhl_order_with_h2h(games_long = games_long) %>%
  render_standings(
    cols = final_cols_league,
    text_cols = text_cols_league,
    text_widths_pt = text_widths_league,
    status_df = status_all, # or NULL if you prefer no marks
    caption = paste0(overall_title, ": League"),
    label = "nhl-standings",
    add_legend = TRUE
  )

if (knitr::is_latex_output()) cat("\n\\clearpage\n") # 'asis' option is needed for these to write code dynamically

# Western Conference table
tibble(west_ordered) %>%
  render_standings(
    cols = final_cols_conf,
    text_cols = text_cols_conf,
    text_widths_pt = text_widths_conf,
    status_df = status_all, # or NULL if you prefer no marks
    caption = paste0(overall_title, ": Western Conference"),
    label = "nhl-west",
    add_legend = FALSE
  )

# Eastern Conference table
tibble(east_ordered) %>%
  render_standings(
    cols = final_cols_conf,
    text_cols = text_cols_conf,
    text_widths_pt = text_widths_conf,
    status_df = status_all, # or NULL if you prefer no marks
    caption = paste0(overall_title, ": Eastern Conference"),
    label = "nhl-east",
    add_legend = TRUE
  )

if (knitr::is_latex_output()) cat("\n\\clearpage\n")

# Divisions (Western Conference)

tibble(division_tables$Central) %>%
  render_standings(
    cols = final_cols_div,
    text_cols = text_cols_div,
    text_widths_pt = text_widths_div,
    status_df = status_all, # or NULL if you prefer no marks
    caption = paste0(overall_title, ": Central Division"),
    label = "nhl-central",
    add_legend = FALSE
  )
```

```

)

tibble(division_tables$Pacific) %>%
  render_standings(
    cols = final_cols_div,
    text_cols = text_cols_div,
    text_widths_pt = text_widths_div,
    status_df = status_all, # or NULL if you prefer no marks
    caption = paste0(overall_title, ": Pacific Division"),
    label = "nhl-pacific",
    add_legend = TRUE
  )

if (knitr::is_latex_output()) cat("\\clearpage\\n")

# Divisions (Eastern Conference)

tibble(division_tables$Atlantic) %>%
  render_standings(
    cols = final_cols_div,
    text_cols = text_cols_div,
    text_widths_pt = text_widths_div,
    status_df = status_all, # or NULL if you prefer no marks
    caption = paste0(overall_title, ": Atlantic Division"),
    label = "nhl-atlantic",
    add_legend = FALSE
  )

tibble(division_tables$Metropolitan) %>%
  render_standings(
    cols = final_cols_div,
    text_cols = text_cols_div,
    text_widths_pt = text_widths_div,
    status_df = status_all, # or NULL if you prefer no marks
    caption = paste0(overall_title, ": Metropolitan Division"),
    label = "nhl-metro",
    add_legend = TRUE
  )

if (knitr::is_latex_output()) cat("\\clearpage\\n")

# Playoff-format tables (no marks)

tibble(auto_west) %>%
  render_standings(
    cols = final_cols_conf,
    text_cols = text_cols_conf,
    text_widths_pt = text_widths_conf,
    status_df = status_all, # or NULL if you prefer no marks
    caption = paste0(overall_title, ": Western Conference, Auto qualifiers (top 3 per division)",
    label = "nhl-west-auto",
    add_legend = FALSE
  )

```



```

tibble(west_wildcards) %>%
  render_standings(
    cols = final_cols_conf,
    text_cols = text_cols_conf,
    text_widths_pt = text_widths_conf,
    status_df = status_all, # or NULL if you prefer no marks
    caption = paste0(overall_title, ": Western Conference, Wild cards (top 2 remaining)"),
    label = "nhl-west-wc",
    add_legend = FALSE
  )

tibble(auto_east) %>%
  render_standings(
    cols = final_cols_conf,
    text_cols = text_cols_conf,
    text_widths_pt = text_widths_conf,
    status_df = status_all, # or NULL if you prefer no marks
    caption = paste0(overall_title, ": Eastern Conference, Auto qualifiers (top 3 per division)"),
    label = "nhl-east-auto",
    add_legend = FALSE
  )

tibble(east_wildcards) %>%
  render_standings(
    cols = final_cols_conf,
    text_cols = text_cols_conf,
    text_widths_pt = text_widths_conf,
    status_df = status_all, # or NULL if you prefer no marks
    caption = paste0(overall_title, ": Eastern Conference, Wild cards (top 2 remaining)"),
    label = "nhl-east-wc",
    add_legend = TRUE
  )

if (knitr::is_latex_output()) cat("\\clearpage\\n")

```