



mlbplotR :: CHEAT SHEET

Create **ggplot2** and **gt** visuals with Major League Baseball logos, cap logos, headshots and team colors — you supply team abbreviations or MLBAM player ids, mlbplotR turns them into images and hex colors. By Camden Kay (with Sebastian Carl); the baseball member of the *plotR family (nflplotR, cfbplotR). Page 1: the ggplot2 layer — geoms, scales, theme elements.

v1.2.0.9000

R ≥ 3.5.0 · ggplot2 · gt
camdenk.github.io/mlbplotR

GET STARTED

```
install.packages("mlbplotR") # CRAN
# development version
remotes::install_github("camdenk/mlbplotR")

library(mlbplotR)
library(ggplot2)
```

Two keys drive everything: `team_abbr` — Baseball Savant team abbreviations — and `player_id` — the player's MLBAM id (the id in his Savant page URL).

TEAM KEYS

```
valid_team_names() # 30 teams + AL
NL MLB
valid_team_names(remove_league_info = TRUE)
clean_team_abbrs(c("STL", "WSN", "CHW"))
# alternates -> Savant abbreviations
clean_team_abbrs(x, keep_non_matches = FALSE)
```

Geoms and scales clean abbreviations for you; `clean_team_abbrs()` is for prepping join / color columns yourself.

LOGO & HEADSHOT GEOMS

Geom	Key aes
<code>geom_mlb_logos()</code>	<code>team_abbr</code>
<code>geom_mlb_scoreboard_logos()</code>	<code>team_abbr</code>
<code>geom_mlb_dot_logos()</code>	<code>team_abbr</code>
<code>geom_mlb_dark_cap_logos()</code>	<code>team_abbr</code>
<code>geom_mlb_light_cap_logos()</code>	<code>team_abbr</code>
<code>geom_mlb_headshots()</code>	<code>player_id</code>
<code>geom_mlb_dot_headshots()</code>	<code>player_id</code>
<code>geom_milb_logos()</code>	<code>team_name</code> (full)
<code>geom_milb_light_cap_logos()</code>	MLB variants
<code>geom_milb_dot_logos()</code>	
<code>geom_milb_dot_headshots()</code>	

SHARED IMAGE AESTHETICS

Aes	Meaning
<code>x, y</code>	position (required)
<code>width / height</code>	size in npc; typical 0.075 / 0.1
<code>alpha</code>	transparency 0-1
<code>colour</code>	colorize image; "b/w" = grayscale
<code>angle</code>	rotation 0-360
<code>hjust, vjust</code>	adjust around x / y
<code>nudge_x, nudge_y</code>	offset labels (param)

REFERENCE LINES & ANY IMAGE

```
ggplot(mtcars, aes(displacement, mpg)) +
  geom_point() +
  geom_median_lines(v_var = displacement, h_var = mpg) +
  geom_mean_lines(v_var = displacement, h_var = mpg, color = "blue")
```

`geom_mean_lines()` / `geom_median_lines()` draw vertical (`v_var`) and horizontal (`h_var`) reference lines at the mean / median of a variable — league-average crosshairs.

```
ggplot(df, aes(x, y)) +
  geom_from_path(aes(path = url), width = 0.1)
```

`geom_from_path()` plots any image from a URL or local path with the same aesthetics.

TEAM COLOR SCALES

```
scale_color_mlb(type = "secondary")
scale_fill_mlb(alpha = 0.4) # type = "primary"
```

Map `team_abbr` values to team hex colors. Args: `type` ("primary" / "secondary"), `values` (override), `alpha`, `guide` = NULL (no legend by default). `scale_colour_mlb()` is an alias.

THEME ELEMENTS (AXIS IMAGES)

Modern way to put logos / headshots in **non-data** plot components — axis text, strips, titles:

```
ggplot(df, aes(x = teams, y = value)) +
  geom_col(aes(fill = teams), width = 0.5) +
  scale_fill_mlb(alpha = 0.4) +
  theme_minimal() +
  theme(axis.text.x = element_mlb_logo())
```

Element	Render s
<code>element_mlb_logo()</code>	team logo
<code>element_mlb_scoreboard_logo()</code>	scoreboard logo
<code>element_mlb_dot_logo()</code>	dot logo
<code>element_mlb_dark_cap_logo()</code>	cap logos
<code>element_mlb_light_cap_logo()</code>	
<code>element_mlb_headshot()</code>	headshot from id
<code>element_mlb_dot_headshot()</code>	
<code>element_milb_logo()</code>	MLB variant s
<code>element_milb_light_cap_logo()</code>	
<code>element_milb_dot_logo()</code>	
<code>element_milb_dot_headshot()</code>	
<code>element_path()</code>	any image URL

Superseded (still exported): `scale_x_mlb()` / `scale_y_mlb()` + `theme_x_mlb()` / `theme_y_mlb()` — html-label axis logos via `ggtext`; prefer the `element_*()` route.

PREVIEW AT PRINT SIZE

```
ggpreview(plot = ggplot2::last_plot(),
  width = 8, height = 4.5, dpi = 300,
  asp = 16/9)
```

Renders a temp png at the exact save dimensions — image sizes only make sense at final aspect ratio.

RECIPE: TEAM SCATTER

```
df <- data.frame(team_abbr = valid_team_names(
  remove_league_info = TRUE),
  x = rnorm(30), y = rnorm(30))
ggplot(df, aes(x, y)) +
  geom_mean_lines(v_var = df$x, h_var = df$y) +
  geom_mlb_logos(aes(team_abbr = team_abbr),
    width = 0.075, alpha = 0.9)
```

RECIPE: PLAYER LEADERBOARD

```
# player_id = MLBAM (Savant) id
lead <- data.frame(
  player_id = c("660271", "592450", "665742"),
  war = c(9.0, 11.2, 7.9))
ggplot(lead, aes(x = war,
  y = reorder(player_id, war))) +
  geom_col(width = 0.4) +
  geom_mlb_headshots(aes(player_id =
    player_id),
    height = 0.2) +
  theme(axis.text.y = element_blank())
```

RECIPE: LOGOS ON THE AXIS

```
ggplot(df, aes(x = team_abbr, y = y)) +
  geom_col(aes(color = team_abbr,
    fill = team_abbr), width = 0.5) +
  scale_color_mlb(type = "secondary") +
  scale_fill_mlb(alpha = 0.4) +
  theme_minimal() +
  theme(axis.text.x = element_mlb_logo())
```

Facet by team instead? Order facets by division with `mlb_team_factor()` (page 2).



GT TABLE HELPERS

LOGOS IN TABLE CELLS

Each takes (`gt_object`, `columns`, `height = 30`, `locations = NULL`) and converts abbreviations from `valid_team_names()` into images. Wrappers around [gtExtras::gt_image_rows\(\)](#) boilerplate.

Function	Image
<code>gt_fmt_mlb_logo()</code>	primary logo
<code>gt_fmt_mlb_scoreboard_logo()</code>	scoreboard logo
<code>gt_fmt_mlb_dot_logo()</code>	dot logo
<code>gt_fmt_milb_logo()</code>	MiLB (full team name)
<code>gt_fmt_milb_dot_logo()</code>	

```
library(gt)
df <- data.frame(team = valid_team_names()
[1:5],
                 logo = valid_team_names()
[1:5])
df %>%
  gt::gt() %>%
  gt_fmt_mlb_logo(columns = "logo")
```

`locations` = accepts `gt::cells_body()`, `cells_stub()`, `cells_column_labels()`, `cells_row_groups()` (or a `list()` of them) to target other cells.

HEADSHOTS IN TABLE CELLS

```
data.frame(player = c("Seager", "deGrom"),
           savant_id = c(608369, 594798)) %>%
  gt::gt() %>%
  gt_fmt_mlb_headshot(columns = "savant_id")

gt_fmt_mlb_headshot() · gt_fmt_mlb_dot_headshot() ·
gt_fmt_milb_dot_headshot() — the column holds
MLBAM ids; non-matching ids render a placeholder.
```

IMAGES AS COLUMN LABELS

```
df %>% gt::gt() %>%
  gt::cols_label(
    BAL = gt_mlb_column_labels("BAL",
  "mlb_logo"),
    LAD = gt_mlb_column_labels("LAD",
  "dot_logo"),
    Trout = gt_mlb_column_labels(545361,
  "headshot"))

gt_mlb_column_labels(value, type, height = 30,
na_headshot_to_logo = TRUE) — type is one of
"mlb_logo", "scoreboard_logo", "dot_logo",
"headshot", "dot_headshot".
```

STACKED, TEAM-COLORED TEXT

```
load_mlb_teams() %>%
  dplyr::slice(1:5) %>%
  dplyr::select(team_abbr, team_location,
               team_mascot) %>%

  gt::gt() %>%
  gt_merge_stack_team_color(
    col1 = "team_location",
    col2 = "team_mascot",
    team_col = "team_abbr")
```

Merges two columns, stacking `col1` over `col2`, coloring the bottom line with `team_col`'s team color. Tune `font_sizes = c(12, 14)`, `font_weights = c("lighter", "bold")`, `font_variants = "small-caps"`, `color` (top text). Adapted from [gtExtras::gt_merge_stack\(\)](#).

TIER CHARTS

```
mlb_team_tiers(data) # tier_no + team_abbr
mlb_player_tiers(data) # tier_no + player_id
```

Dark tier-maker plots. Args: `title`, `subtitle`, `tier_desc` (named vector), `presort`, `alpha`, `width`, `no_line_below_tier`, `logo_type` ("main" / "scoreboard" / "dot"), `colors`, `devel = TRUE` for fast text-only iteration.

LOOKUPS & UTILITIES

BUNDLED TEAM DATA

```
teams <- load_mlb_teams()
# team_abbr, team_location, team_mascot,
# logo URLs + hex color codes per team
mlb <- load_milb_teams()
# full-name keyed; logo URLs + league info
```

`load_mlb_teams()` ships in the package;
`load_milb_teams()` reads from the `camdenk/mlbplotR`-data GitHub repo.

FACET ORDERING

```
df$team <- mlb_team_factor(df$team)
ggplot(df, aes(x, y)) +
  geom_line(aes(color = team)) +
  scale_color_mlb() +
  facet_wrap(~team, ncol = 5)
```

`mlb_team_factor(teams)` orders an abbreviation vector by division then team — faceted plots come out grouped AL East → NL West.

DEPRECATED

`load_headshots()` (the old id-crosswalk table) is deprecated for lack of support — use [baseballr::chadwick_player_lu\(\)](#) instead (right column).

FUNCTION INDEX

65 exports: 14 `geom_*` · 12 `element_*` · 5 `scale_*` · 10 `gt_*` · 2 `theme_*` · 2 tiers · loaders + helpers · exported `Geom*` ggproto objects (`GeomMLBLogo`, `GeomFromPath`, ...) for extension authors.

THE MLBAM-ID STORY

Headshots key on the **MLBAM id** — the same id in a player's Baseball Savant URL and in MLB's statsapi. Get it from [baseballr](#):

```
library(baseballr)
playerid_lookup("Judge", "Aaron")$mlbam_id
# or the full Chadwick register:
lu <- chadwick_player_lu() # key_mlbam col
```

The register also carries `key_bbrief`, `key_fangraphs`, `key_retro` — one join away from any stat source.

WORKED EXAMPLE: DATA → DISPLAY

```
library(baseballr); library(mlbplotR)
library(dplyr); library(ggplot2)

# 1 · stats with an MLBAM id column
lu <- chadwick_player_lu() %>%
  filter(name_last %in% c("Ohtani", "Judge",
                        "Soto")) %>%
  select(name_last, key_mlbam)
stats <- lu %>%
  mutate(war = c(9.0, 11.2, 7.9),
         player_id = as.character(key_mlbam))

# 2 · headshot leaderboard
ggplot(stats, aes(war,
  reorder(name_last, war))) +
  geom_col(width = 0.4, alpha = 0.5) +
  geom_mlb_headshots(aes(player_id =
player_id),
                    height = 0.25) +
  labs(x = "WAR", y = NULL)

# 3 · same data as a gt table
stats %>%
  select(player_id, name_last, war) %>%
  gt::gt() %>%
  gt_fmt_mlb_headshot(columns = "player_id")
```

Draw the field itself with [sportyR::geom_baseball\("mlb"\)](#); pull Savant pitch data with [baseballr](#) scrapers or [sportsdataverse-py](#)'s `mlb_statcast*`.



mlbplotR :: CHEAT SHEET

Create **ggplot2** and **gt** visuals with Major League Baseball logos, cap logos, headshots and team colors — you supply team abbreviations or MLBAM player ids, mlbplotR turns them into images and hex colors. By Camden Kay (with Sebastian Carl); the baseball member of the *plotR family (nflplotR, cfbplotR). Page 1: the ggplot2 layer — geoms, scales, theme elements.

v1.2.0.9000

R ≥ 3.5.0 · ggplot2 · gt
camdenk.github.io/mlbplotR

GET STARTED

```
install.packages("mlbplotR") # CRAN
# development version
remotes::install_github("camdenk/mlbplotR")
```

```
library(mlbplotR)
library(ggplot2)
```

Two keys drive everything: `team_abbr` — Baseball Savant team abbreviations — and `player_id` — the player's MLBAM id (the id in his Savant page URL).

TEAM KEYS

```
valid_team_names() # 30 teams + AL
NL MLB
valid_team_names(remove_league_info = TRUE)
clean_team_abbrs(c("STL", "WSN", "CHW"))
# alternates -> Savant abbreviations
clean_team_abbrs(x, keep_non_matches = FALSE)
```

Geoms and scales clean abbreviations for you; `clean_team_abbrs()` is for prepping join / color columns yourself.

LOGO & HEADSHOT GEOMS

Geom	Key aes
<code>geom_mlb_logos()</code>	<code>team_abbr</code>
<code>geom_mlb_scoreboard_logos()</code>	<code>team_abbr</code>
<code>geom_mlb_dot_logos()</code>	<code>team_abbr</code>
<code>geom_mlb_dark_cap_logos()</code>	<code>team_abbr</code>
<code>geom_mlb_light_cap_logos()</code>	<code>team_abbr</code>
<code>geom_mlb_headshots()</code>	<code>player_id</code>
<code>geom_mlb_dot_headshots()</code>	<code>player_id</code>
<code>geom_milb_logos()</code>	<code>team_name</code> (full)
<code>geom_milb_light_cap_logos()</code> <code>geom_milb_dot_logos()</code> <code>geom_milb_dot_headshots()</code>	MiLB variants

SHARED IMAGE AESTHETICS

Aes	Meaning
<code>x, y</code>	position (required)
<code>width / height</code>	size in npc; typical 0.075 / 0.1
<code>alpha</code>	transparency 0-1
<code>colour</code>	colorize image; "b/w" = grayscale
<code>angle</code>	rotation 0-360
<code>hjust, vjust</code>	adjust around x / y
<code>nudge_x, nudge_y</code>	offset labels (param)

REFERENCE LINES & ANY IMAGE

```
ggplot(mtcars, aes(displ, mpg)) +
  geom_point() +
  geom_median_lines(v_var = 400, h_var = 15) +
  geom_mean_lines(v_var = 150, h_var = 30,
    color = "blue")
```

`geom_mean_lines()` / `geom_median_lines()` draw vertical (`v_var`) and horizontal (`h_var`) reference lines at the mean / median of a variable — league-average crosshairs.

```
ggplot(df, aes(x, y)) +
  geom_from_path(aes(path = url), width = 0.1)
```

`geom_from_path()` plots any image from a URL or local path with the same aesthetics.

TEAM COLOR SCALES

```
scale_color_mlb(type = "secondary")
scale_fill_mlb(alpha = 0.4) # type =
"primary"
```

Map `team_abbr` values to team hex colors. Args: `type` ("primary" / "secondary"), `values` (override), `alpha`, `guide` = NULL (no legend by default). `scale_colour_mlb()` is an alias.

THEME ELEMENTS (AXIS IMAGES)

Modern way to put logos / headshots in **non-data** plot components — axis text, strips, titles:

```
ggplot(df, aes(x = teams, y = value)) +
  geom_col(aes(fill = teams), width = 0.5) +
  scale_fill_mlb(alpha = 0.4) +
  theme_minimal() +
  theme(axis.text.x = element_mlb_logo())
```

Element	Render s
<code>element_mlb_logo()</code>	team logo
<code>element_mlb_scoreboard_logo()</code>	scoreboard logo
<code>element_mlb_dot_logo()</code>	dot logo
<code>element_mlb_dark_cap_logo()</code> <code>element_mlb_light_cap_logo()</code>	cap logos
<code>element_mlb_headshot()</code> <code>element_mlb_dot_headshot()</code>	headshot from id
<code>element_milb_logo()</code> <code>element_milb_light_cap_logo()</code> <code>element_milb_dot_logo()</code> <code>element_milb_dot_headshot()</code>	MiLB variant s
<code>element_path()</code>	any image URL

Superseded (still exported): `scale_x_mlb()` / `scale_y_mlb()` + `theme_x_mlb()` / `theme_y_mlb()` — html-label axis logos via `ggtext`; prefer the `element_*`() route.

PREVIEW AT PRINT SIZE

```
ggpreview(plot = ggplot2::last_plot(),
  width = 8, height = 4.5, dpi = 300,
  asp = 16/9)
```

Renders a temp png at the exact save dimensions — image sizes only make sense at final aspect ratio.

RECIPE: TEAM SCATTER

```
df <- data.frame(team_abbr = valid_team_names(
  remove_league_info = TRUE),
  x = rnorm(30), y = rnorm(30))
ggplot(df, aes(x, y)) +
  geom_mean_lines(v_var = df$x, h_var = df$y) +
  geom_mlb_logos(aes(team_abbr = team_abbr),
    width = 0.075, alpha = 0.9)
```

RECIPE: PLAYER LEADERBOARD

```
# player_id = MLBAM (Savant) id
lead <- data.frame(
  player_id = c("660271", "592450", "665742"),
  war = c(9.0, 11.2, 7.9))
ggplot(lead, aes(x = war,
  y = reorder(player_id, war))) +
  geom_col(width = 0.4) +
  geom_mlb_headshots(aes(player_id =
    player_id),
    height = 0.2) +
  theme(axis.text.y = element_blank())
```

RECIPE: LOGOS ON THE AXIS

```
ggplot(df, aes(x = team_abbr, y = y)) +
  geom_col(aes(color = team_abbr,
    fill = team_abbr), width = 0.5)
+
  scale_color_mlb(type = "secondary") +
  scale_fill_mlb(alpha = 0.4) +
  theme_minimal() +
  theme(axis.text.x = element_mlb_logo())
```

Facet by team instead? Order facets by division with `mlb_team_factor()` (page 2).



GT TABLE HELPERS

LOGOS IN TABLE CELLS

Each takes (`gt_object`, `columns`, `height = 30`, `locations = NULL`) and converts abbreviations from `valid_team_names()` into images. Wrappers around `gtExtras::gt_image_rows()` boilerplate.

Function	Image
<code>gt_fmt_mlb_logo()</code>	primary logo
<code>gt_fmt_mlb_scoreboard_logo()</code>	scoreboard logo
<code>gt_fmt_mlb_dot_logo()</code>	dot logo
<code>gt_fmt_milb_logo()</code>	MiLB (full team name)
<code>gt_fmt_milb_dot_logo()</code>	

```
library(gt)
df <- data.frame(team = valid_team_names()
[1:5],
                 logo = valid_team_names()
[1:5])
df %>%
  gt::gt() %>%
  gt_fmt_mlb_logo(columns = "logo")
```

`locations` = accepts `gt::cells_body()`, `cells_stub()`, `cells_column_labels()`, `cells_row_groups()` (or a `list()` of them) to target other cells.

HEADSHOTS IN TABLE CELLS

```
data.frame(player = c("Seager", "deGrom"),
           savant_id = c(608369, 594798)) %>%
  gt::gt() %>%
  gt_fmt_mlb_headshot(columns = "savant_id")

gt_fmt_mlb_headshot() · gt_fmt_mlb_dot_headshot()
· gt_fmt_milb_dot_headshot() — the column holds
MLBAM ids; non-matching ids render a placeholder.
```

IMAGES AS COLUMN LABELS

```
df %>% gt::gt() %>%
  gt::cols_label(
    BAL = gt_mlb_column_labels("BAL",
  "mlb_logo"),
    LAD = gt_mlb_column_labels("LAD",
  "dot_logo"),
    Trout = gt_mlb_column_labels(545361,
  "headshot"))
```

`gt_mlb_column_labels(value, type, height = 30, na_headshot_to_logo = TRUE)` — `type` is one of "mlb_logo", "scoreboard_logo", "dot_logo", "headshot", "dot_headshot".

STACKED, TEAM-COLORED TEXT

```
load_mlb_teams() %>%
  dplyr::slice(1:5) %>%
  dplyr::select(team_abbr, team_location,
               team_mascot) %>%

  gt::gt() %>%
  gt_merge_stack_team_color(
    col1 = "team_location",
    col2 = "team_mascot",
    team_col = "team_abbr")
```

Merges two columns, stacking `col1` over `col2`, coloring the bottom line with `team_col`'s team color. Tune `font_sizes = c(12, 14)`, `font_weights = c("lighter", "bold")`, `font_variants = "small-caps"`, `color` (top text). Adapted from `gtExtras::gt_merge_stack()`.

TIER CHARTS

```
mlb_team_tiers(data) # tier_no + team_abbr
mlb_player_tiers(data) # tier_no + player_id
```

Dark tier-maker plots. Args: `title`, `subtitle`, `tier_desc` (named vector), `presort`, `alpha`, `width`, `no_line_below_tier`, `logo_type` ("main" / "scoreboard" / "dot"), `colors`, `devel = TRUE` for fast text-only iteration.

LOOKUPS & UTILITIES

BUNDLED TEAM DATA

```
teams <- load_mlb_teams()
# team_abbr, team_location, team_mascot,
# logo URLs + hex color codes per team
milb <- load_milb_teams()
# full-name keyed; logo URLs + league info
```

`load_mlb_teams()` ships in the package;
`load_milb_teams()` reads from the `camdenk/mlbplotR`-data GitHub repo.

FACET ORDERING

```
df$team <- mlb_team_factor(df$team)
ggplot(df, aes(x, y)) +
  geom_line(aes(color = team)) +
  scale_color_mlb() +
  facet_wrap(~team, ncol = 5)
```

`mlb_team_factor(teams)` orders an abbreviation vector by division then team — faceted plots come out grouped AL East → NL West.

DEPRECATED

`load_headshots()` (the old id-crosswalk table) is deprecated for lack of support — use `baseballr::chadwick_player_lu()` instead (right column).

FUNCTION INDEX

65 exports: 14 `geom_*` · 12 `element_*` · 5 `scale_*` · 10 `gt_*` · 2 `theme_*` · 2 tiers · loaders + helpers · exported `Geom*` ggproto objects (`GeomMLBlogo`, `GeomFromPath`, ...) for extension authors.

THE MLBAM-ID STORY

Headshots key on the **MLBAM id** — the same id in a player's Baseball Savant URL and in MLB's statsapi. Get it from `baseballr`:

```
library(baseballr)
playerid_lookup("Judge", "Aaron")$mlbam_id
# or the full Chadwick register:
lu <- chadwick_player_lu() # key_mlbam col
```

The register also carries `key_bbreif`, `key_fangraphs`, `key_retro` — one join away from any stat source.

WORKED EXAMPLE: DATA → DISPLAY

```
library(baseballr); library(mlbplotR)
library(dplyr); library(ggplot2)

# 1 · stats with an MLBAM id column
lu <- chadwick_player_lu() %>%
  filter(name_last %in% c("Ohtani", "Judge",
                        "Soto")) %>%
  select(name_last, key_mlbam)
stats <- lu %>%
  mutate(war = c(9.0, 11.2, 7.9),
         player_id = as.character(key_mlbam))

# 2 · headshot leaderboard
ggplot(stats, aes(war,
                 reorder(name_last, war))) +
  geom_col(width = 0.4, alpha = 0.5) +
  geom_mlb_headshots(aes(player_id =
                        player_id),
                    height = 0.25) +
  labs(x = "WAR", y = NULL)

# 3 · same data as a gt table
stats %>%
  select(player_id, name_last, war) %>%
  gt::gt() %>%
  gt_fmt_mlb_headshot(columns = "player_id")
```

Draw the field itself with `sportyr::geom_baseball("mlb")`; pull Savant pitch data with `baseballr` scrapers or `sportsdataverse`-py's `mlb_statcast_*`.