sportyR :: CHEAT SHEET

Plot scaled `ggplot2` representations of sports playing surfaces, drawn to rule-book specification, as the base layer for any tracking / shot / event data. Nine `geom_{sport}()` surface plotters share one argument set; `cani_*()` helpers list what can be plotted and colored. Part of the SportsDataverse; Python mirror: **sportypy**.

v2.2.3

R ≥ 3.5 · ggplot2 · grid
sportyR.sportsdataverse.org

GET STARTED

```
install.packages("sportyR")
# dev version
devtools::install_github(
  "sportsdataverse/sportyR")

library(sportyR)
geom_basketball("nba")    # a full NBA court
```

Every plot is a regular `ggplot2` object: add layers with `+`, save with `ggsave()`, animate with **gganimate**.

THE GEOM_{SPORT}() FAMILY

Function	Surface · units arg
`geom_baseball()`	field · `field_units`
`geom_basketball()`	court · `court_units`
`geom_curling()`	sheet · `sheet_units`
`geom_football()`	field · `field_units`
`geom_hockey()`	rink · `rink_units`
`geom_lacrosse()`	field · `field_units`
`geom_soccer()`	pitch · `pitch_units`
`geom_tennis()`	court · `court_units`
`geom_volleyball()`	court · `court_units`

The dimension-override list is named the same way: `field_updates`, `court_updates`, `rink_updates`, `pitch_updates`, `sheet_updates`.

SHARED ARGUMENTS

Arg	Meaning
`league`	required; case-insensitive (`"custom"` allowed)
`display_range`	zoom preset, default `"full"`
`{surface}_updates`	list of dimension overrides
`color_updates`	list of feature colors
`rotation`	degrees, counterclockwise
`x_trans` `y_trans`	shift the origin (surface units)
`{surface}_units`	plot in your data's units
`xlims` `ylims`	hard axis limits (override preset)

SUPPORTED LEAGUES

Sport	Leagues (+ `custom`)	Units
baseball	MLB · MiLB · NCAA · NFHS · Little League · Pony	ft
basketball	NBA · WNBA · NBA G League · NCAA · FIBA · NFHS	ft; FIBA m
curling	WCF · Curling Canada	ft
football	NFL · NCAA · CFL · NFHS11 / 9 / 8 / 6	yd
hockey	NHL · AHL · ECHL · PWHL · PHF · NWHL · IIHF · NCAA · OHL · QMJHL · USHL	ft; IIHF m
lacrosse	NCAAM · NCAAW · NLL · PLL · USAM · USAW · World Lacrosse	yd / m / ft
soccer	FIFA · EPL · MLS · NCAA · NWSL	m / yd
tennis	ATP · WTA · ITF · ITA · NCAA · USTA	ft
volleyball	FIVB · NCAA · USA Volleyball	m

League codes are case-insensitive strings: `geom_hockey("nhl")` == `geom_hockey("NHL")`. Per-league feature dimensions live in the bundled `surface_dimensions` data.

DISCOVERY: CANI_*()

```
cani_plot_league("MLB")
# which geom_{sport}() draws this league?
cani_plot_sport("basketball")
# which leagues does this sport support?
cani_color_league_features("NCAA",
"basketball")
# feature names accepted by color_updates
```

All three print to the console. Pass the sport name to `cani_color_league_features()` when a league (e.g. NCAA) spans several sports.

DISPLAY_RANGE PRESETS

`"full"` and `"in_bounds_only"` work for every sport (baseball: `"full"` / `"infield"`). Spelling is forgiving: `"red zone"`, `"redzone"` and `"red_zone"` all match; `offense` / `offence` both work.

Sport	Extra ranges
baseball	`infield`
basketball	`offense defense offensive_key defensive_key offensive_paint defensive_paint`
curling	`house`
football	`offense defense red_zone oredzone dredzone`
hockey	`ozone nzone dzone offense defense neutral`
lacrosse / soccer / volleyball	`offense defense` (half field / pitch / court)
tennis	`serve receive serving_half receiving_half`

```
geom_football("nfl", display_range = "red
zone")
geom_curling("wcf", display_range = "house")
```

UNITS & CONVERSION

Surfaces draw in the rule-book unit (table left). Have data in other units? Either re-draw the surface in your units or convert the data:

```
# draw an NHL rink in meters
geom_hockey("nhl", rink_units = "m")

# or convert data cols: mm cm m in ft yd
convert_units(tracking_data, "yd", "m",
              conversion_columns = c("x",
"y"))
convert_units(1, "in", "cm")   # scalar too
```

COORDINATE SYSTEM

Origin `(0, 0)` is always the **center of the surface**. Match a data feed whose origin is a corner by shifting with `x_trans` / `y_trans`; rotate with `rotation` (degrees, CCW). Data-frame twins: `rotate_coords(df, angle = 90)` and `reflect(df, over_y = TRUE)` (need `x`, `y` columns).

QUICK PLOTS

```
# half court, TV-right, rotated upright
geom_basketball("nba",
  display_range = "offense", rotation = 270)

# MLB infield only
geom_baseball("mlb", display_range =
"infield",
              rotation = 270)

# non-regulation FIFA pitch (100m x 75m)
geom_soccer("fifa", pitch_updates = list(
  pitch_length = 100, pitch_width = 75))
```

OVERLAY TRACKING / SHOT DATA

```
# PHF shots: data origin = lower-left corner,
# so shift plot origin by half rink (200x85
ft)
rink <- geom_hockey("phf",
  x_trans = 100, y_trans = 42.5)
rink +
  geom_point(data = shots, aes(x, y),
             color = "#2251b8")

# NFL Big Data Bowl field (120x53.3 yd)
field <- geom_football("nfl",
  x_trans = 60, y_trans = 26.6667)
field + geom_point(data = frame, aes(x, y))
```

Basketball pbp from **hoopR** / **wehoop** ships `coordinate_x` / `coordinate_y` — shift the same way, then `+ geom_point(aes(coordinate_x, coordinate_y))`. Add `geom_segment()` for passes, **gganimate** `transition_time(frame_id)` for movement.

SPORTSDATAVERSE

Coordinate-bearing data sources that plug straight in: **hoopR** · **wehoop** (ESPN pbp shot coords), **fastRhockey** (`x_coord` / `y_coord`), **baseballr** (Statcast `hc_x` / `hc_y`), NFL Big Data Bowl tracking. Python mirror **sportypy** has the same surfaces + league codes (sportypy.sportsdataverse.org).

Per-sport reference :: leagues, features, custom surfaces

Feature names below feed `color_updates`; dimension keys feed `{surface}_updates`. Full lists: `cani_color_league_features()` · `surface_dimensions`

COURT SPORTS

BASKETBALL

NBA WNBA "nba g league" NCAA FIBA NFHS · ft (FIBA m)

Colors: `offensive_half_court defensive_half_court court_apron center_circle_outline / _fill division_line endline sideline two_point_range three_point_line painted_area lane_boundary free_throw_circle_outline / _fill / _dash lane_space_mark restricted_arc backboard basket_ring net`

Dims: `court_length court_width lane_length lane_width basket_center_to_three_point_arc` …

```
geom_basketball("wnba",
  display_range = "offensive_key")
```

TENNIS

ATP WTA ITF ITA NCAA USTA · ft

Colors: `baseline singles_sideline doubles_sideline serviceline center_serviceline center_mark ad_court deuce_court backcourt doubles_alley court_apron net`

```
geom_tennis("usta",
  display_range = "serving_half")
```

VOLLEYBALL

FIVB NCAA "usa volleyball" · m

Colors: `free_zone front_zone defensive_backcourt offensive_backcourt court_apron end_line sideline attack_line center_line service_zone_mark substitution_zone`

```
geom_volleyball("ncaa",
  display_range = "offense")
```

FIELD SPORTS

FOOTBALL

NFL NCAA CFL NFHS11 / 9 / 8 / 6 · yd

Colors: `field_apron offensive_half defensive_half offensive_endzone defensive_endzone end_line sideline field_border red_zone_border major_yard_line goal_line minor_yard_line directional_arrow try_mark yardage_marker restricted_area coaching_box team_bench_area`

Dims: `field_length field_width endzone_length` …

```
geom_football("cfl",
  display_range = "red_zone")
```

SOCCER

FIFA EPL MLS NCAA NWSL · m (FIFA EPL) / yd (MLS NCAA NWSL)

Colors: `touchline goal_line halfway_line center_circle center_mark penalty_box goal_box penalty_mark corner_defensive_mark goal`

Dims: `pitch_length pitch_width penalty_box_length goal_width` …

```
geom_soccer("nwsl",
  display_range = "offense")
```

LACROSSE

NCAAM NCAAW NLL PLL USAM USAW "world lacrosse" · yd (NCAAM PLL USAM) / m (NCAAW USAW) / ft (NLL WL)

Colors: `defensive_zone neutral_zone offensive_zone boards end_line sideline center_line wing_line restraining_line goal_circle goal_arc goal_fan goal_mouth goal_frame goal_net center_face_off_marker` …

```
geom_lacrosse("pll")
```

DIAMOND · ICE · SHEET

BASEBALL

MLB MiLB NCAA NFHS "little league" Pony · ft

Colors: `infield_dirt infield_grass pitchers_mound base pitchers_plate batters_box catchers_box foul_line running_lane` + `plot_background`

Dims: `left_field_distance baseline_distance pitchers_mound_center_to_home_plate` …

```
geom_baseball("milb",
  display_range = "infield")
```

HOCKEY

NHL AHL ECHL PWHL PHF NWHL IIHF NCAA OHL QMJHL USHL · ft (IIHF m)

Colors: `boards ozone_ice nzone_ice dzone_ice center_line zone_line goal_line restricted_trapezoid goal_crease_outline / _fill referee_crease center_faceoff_spot faceoff_spot_ring faceoff_line goal_frame goal_fill team_a_bench team_b_penalty_box off_ice_officials_box` …

```
geom_hockey("iihf",
  display_range = "ozone")
```

CURLING

WCF "curling canada" · ft

Colors: `end_1 end_2 centre_zone sheet_apron centre_line tee_line back_line hog_line hack_line courtesy_line hack button house_rings`

```
geom_curling("wcf",
  display_range = "house")
```

CUSTOM LEAGUE / DIMENSIONS

Every sport accepts `league = "custom"`: supply the full parameterization through `{surface}_updates`. Or start from a real league and override only what differs:

```
# backyard court in meters
geom_basketball("custom",
  court_updates = list(
    court_length = 20, court_width = 12,
    court_units = "m"))

# shrink a real league instead
geom_soccer("fifa", pitch_updates = list(
  pitch_length = 100, pitch_width = 75))
```

Dimension key names per sport: `names(surface_dimensions[["basketball"]][["nba"]])`.

THEMING

```
geom_basketball("ncaa", color_updates = list(
  plot_background = "#ffffff00",
  offensive_half_court = "#e8e0d7",
  court_apron = "#e84a27",
  three_point_line = c("#13294b", "#ffffff"),
  painted_area = c("#e84a27", "#ffffff00")))
```

Keys come from `cani_color_league_features()`. Features drawn as line + fill take a length-2 color vector; 8-digit hex gives transparency (`"#ffffff00"` = fully transparent). `plot_background` colors the figure behind the surface.

PYTHON MIRROR & DATA HOOKS

sportypy mirrors the API: classes like `sportypy.surfaces.hockey.NHLRink().draw()` over matplotlib — same leagues, same feature names (sportypy.sportsdataverse.org).

SportsDataverse coordinate columns to overlay: **hoopR** / **wehoop** ESPN pbp `coordinate_x coordinate_y`; **fastRhockey** pbp `x_coord y_coord`; **baseballr** Statcast `hc_x hc_y`; NFL Big Data Bowl tracking `x y` (see page 1 for the `x_trans` / `y_trans` alignment recipe).

sportyR :: CHEAT SHEET

Plot scaled `ggplot2` representations of sports playing surfaces, drawn to rule-book specification, as the base layer for any tracking / shot / event data. Nine `geom_{sport}()` surface plotters share one argument set; `cani_*()` helpers list what can be plotted and colored. Part of the SportsDataverse; Python mirror: **sportypy**.

v2.2.3

R ≥ 3.5 · ggplot2 · grid
sportyR.sportsdataverse.org

GET STARTED

```
install.packages("sportyR")
# dev version
devtools::install_github(
  "sportsdataverse/sportyR")

library(sportyR)
geom_basketball("nba")    # a full NBA court
```

Every plot is a regular `ggplot2` object: add layers with `+`, save with `ggsave()`, animate with **gganimate**.

THE GEOM_{SPORT}() FAMILY

Function	Surface · units arg
`geom_baseball()`	field · `field_units`
`geom_basketball()`	court · `court_units`
`geom_curling()`	sheet · `sheet_units`
`geom_football()`	field · `field_units`
`geom_hockey()`	rink · `rink_units`
`geom_lacrosse()`	field · `field_units`
`geom_soccer()`	pitch · `pitch_units`
`geom_tennis()`	court · `court_units`
`geom_volleyball()`	court · `court_units`

The dimension-override list is named the same way: `field_updates`, `court_updates`, `rink_updates`, `pitch_updates`, `sheet_updates`.

SHARED ARGUMENTS

Arg	Meaning
`league`	required; case-insensitive (`"custom"` allowed)
`display_range`	zoom preset, default `"full"`
`{surface}_updates`	list of dimension overrides
`color_updates`	list of feature colors
`rotation`	degrees, counterclockwise
`x_trans` `y_trans`	shift the origin (surface units)
`{surface}_units`	plot in your data's units
`xlims` `ylims`	hard axis limits (override preset)

SUPPORTED LEAGUES

Sport	Leagues (+ `custom`)	Units
baseball	MLB · MiLB · NCAA · NFHS · Little League · Pony	ft
basketball	NBA · WNBA · NBA G League · NCAA · FIBA · NFHS	ft; FIBA m
curling	WCF · Curling Canada	ft
football	NFL · NCAA · CFL · NFHS11 / 9 / 8 / 6	yd
hockey	NHL · AHL · ECHL · PWHL · PHF · NWHL · IIHF · NCAA · OHL · QMJHL · USHL	ft; IIHF m
lacrosse	NCAAM · NCAAW · NLL · PLL · USAM · USAW · World Lacrosse	yd / m / ft
soccer	FIFA · EPL · MLS · NCAA · NWSL	m / yd
tennis	ATP · WTA · ITF · ITA · NCAA · USTA	ft
volleyball	FIVB · NCAA · USA Volleyball	m

League codes are case-insensitive strings: `geom_hockey("nhl") == geom_hockey("NHL")`. Per-league feature dimensions live in the bundled `surface_dimensions` data.

DISCOVERY: CANI_*()

```
cani_plot_league("MLB")
# which geom_{sport}() draws this league?
cani_plot_sport("basketball")
# which leagues does this sport support?
cani_color_league_features("NCAA",
"basketball")
# feature names accepted by color_updates
```

All three print to the console. Pass the sport name to `cani_color_league_features()` when a league (e.g. NCAA) spans several sports.

DISPLAY_RANGE PRESETS

`"full"` and `"in_bounds_only"` work for every sport (baseball: `"full"` / `"infield"`). Spelling is forgiving: `"red zone"`, `"redzone"` and `"red_zone"` all match; `offense` / `offence` both work.

Sport	Extra ranges
baseball	`infield`
basketball	`offense defense offensive_key defensive_key offensive_paint defensive_paint`
curling	`house`
football	`offense defense red_zone oredzone dredzone`
hockey	`ozone nzone dzone offense defense neutral`
lacrosse / soccer / volleyball	`offense defense` (half field / pitch / court)
tennis	`serve receive serving_half receiving_half`

```
geom_football("nfl", display_range = "red
zone")
geom_curling("wcf", display_range = "house")
```

UNITS & CONVERSION

Surfaces draw in the rule-book unit (table left). Have data in other units? Either re-draw the surface in your units or convert the data:

```
# draw an NHL rink in meters
geom_hockey("nhl", rink_units = "m")

# or convert data cols: mm cm m in ft yd
convert_units(tracking_data, "yd", "m",
              conversion_columns = c("x", "y"))
convert_units(1, "in", "cm")    # scalar too
```

COORDINATE SYSTEM

Origin `(0, 0)` is always the **center of the surface**. Match a data feed whose origin is a corner by shifting with `x_trans` / `y_trans`; rotate with `rotation` (degrees, CCW). Data-frame twins: `rotate_coords(df, angle = 90)` and `reflect(df, over_y = TRUE)` (need `x`, `y` columns).

QUICK PLOTS

```
# half court, TV-right, rotated upright
geom_basketball("nba",
  display_range = "offense", rotation = 270)

# MLB infield only
geom_baseball("mlb", display_range = "infield",
              rotation = 270)

# non-regulation FIFA pitch (100m x 75m)
geom_soccer("fifa", pitch_updates = list(
  pitch_length = 100, pitch_width = 75))
```

OVERLAY TRACKING / SHOT DATA

```
# PHF shots: data origin = lower-left corner,
# so shift plot origin by half rink (200x85 ft)
rink <- geom_hockey("phf",
  x_trans = 100, y_trans = 42.5)
rink +
  geom_point(data = shots, aes(x, y),
             color = "#2251b8")

# NFL Big Data Bowl field (120x53.3 yd)
field <- geom_football("nfl",
  x_trans = 60, y_trans = 26.6667)
field + geom_point(data = frame, aes(x, y))
```

Basketball pbp from **hoopR** / **wehoop** ships `coordinate_x` / `coordinate_y` — shift the same way, then `+ geom_point(aes(coordinate_x, coordinate_y))`. Add `geom_segment()` for passes, **gganimate** `transition_time(frame_id)` for movement.

SPORTSDATAVERSE

Coordinate-bearing data sources that plug straight in: **hoopR** · **wehoop** (ESPN pbp shot coords), **fastRhockey** (`x_coord` / `y_coord`), **baseballr** (Statcast `hc_x` / `hc_y`), NFL Big Data Bowl tracking. Python mirror **sportypy** has the same surfaces + league codes (sportypy.sportsdataverse.org).

Per-sport reference :: leagues, features, custom surfaces

Feature names below feed `color_updates`; dimension keys feed `{surface}_updates`. Full lists: `cani_color_league_features()` · `surface_dimensions`

COURT SPORTS

BASKETBALL

NBA WNBA "nba g league" NCAA FIBA NFHS · ft (FIBA m)

Colors: `offensive_half_court defensive_half_court court_apron center_circle_outline / _fill division_line endline sideline two_point_range three_point_line painted_area lane_boundary free_throw_circle_outline / _fill / _dash lane_space_mark restricted_arc backboard basket_ring net`

Dims: `court_length court_width lane_length lane_width basket_center_to_three_point_arc` …

```
geom_basketball("wnba",
  display_range = "offensive_key")
```

TENNIS

ATP WTA ITF ITA NCAA USTA · ft

Colors: `baseline singles_sideline doubles_sideline serviceline center_serviceline center_mark ad_court deuce_court backcourt doubles_alley court_apron net`

```
geom_tennis("usta",
  display_range = "serving_half")
```

VOLLEYBALL

FIVB NCAA "usa volleyball" · m

Colors: `free_zone front_zone defensive_backcourt offensive_backcourt court_apron end_line sideline attack_line center_line service_zone_mark substitution_zone`

```
geom_volleyball("ncaa",
  display_range = "offense")
```

FIELD SPORTS

FOOTBALL

NFL NCAA CFL NFHS11 / 9 / 8 / 6 · yd

Colors: `field_apron offensive_half defensive_half offensive_endzone defensive_endzone end_line sideline field_border red_zone_border major_yard_line goal_line minor_yard_line directional_arrow try_mark yardage_marker restricted_area coaching_box team_bench_area`

Dims: `field_length field_width endzone_length` …

```
geom_football("cfl",
  display_range = "red_zone")
```

SOCCER

FIFA EPL MLS NCAA NWSL · m (FIFA EPL) / yd (MLS NCAA NWSL)

Colors: `touchline goal_line halfway_line center_circle center_mark penalty_box goal_box penalty_mark corner_defensive_mark goal`

Dims: `pitch_length pitch_width penalty_box_length goal_width` …

```
geom_soccer("nwsl",
  display_range = "offense")
```

LACROSSE

NCAAM NCAAW NLL PLL USAM USAW "world lacrosse" · yd (NCAAM PLL USAM) / m (NCAAW USAW) / ft (NLL WL)

Colors: `defensive_zone neutral_zone offensive_zone boards end_line sideline center_line wing_line restraining_line goal_circle goal_arc goal_fan goal_mouth goal_frame goal_net center_face_off_marker` …

```
geom_lacrosse("pll")
```

DIAMOND · ICE · SHEET

BASEBALL

MLB MiLB NCAA NFHS "little league" Pony · ft

Colors: `infield_dirt infield_grass pitchers_mound base pitchers_plate batters_box catchers_box foul_line running_lane + plot_background`

Dims: `left_field_distance baseline_distance pitchers_mound_center_to_home_plate` …

```
geom_baseball("milb",
  display_range = "infield")
```

HOCKEY

NHL AHL ECHL PWHL PHF NWHL IIHF NCAA OHL QMJHL USHL · ft (IIHF m)

Colors: `boards ozone_ice nzone_ice dzone_ice center_line zone_line goal_line restricted_trapezoid goal_crease_outline / _fill referee_crease center_faceoff_spot faceoff_spot_ring faceoff_line goal_frame goal_fill team_a_bench team_b_penalty_box off_ice_officials_box` …

```
geom_hockey("iihf",
  display_range = "ozone")
```

CURLING

WCF "curling canada" · ft

Colors: `end_1 end_2 centre_zone sheet_apron centre_line tee_line back_line hog_line hack_line courtesy_line hack button house_rings`

```
geom_curling("wcf",
  display_range = "house")
```

CUSTOM LEAGUE / DIMENSIONS

Every sport accepts `league = "custom"`: supply the full parameterization through `{surface}_updates`. Or start from a real league and override only what differs:

```
# backyard court in meters
geom_basketball("custom",
  court_updates = list(
    court_length = 20, court_width = 12,
    court_units = "m"))

# shrink a real league instead
geom_soccer("fifa", pitch_updates = list(
  pitch_length = 100, pitch_width = 75))
```

Dimension key names per sport: `names(surface_dimensions[["basketball"]][["nba"]])`.

THEMING

```
geom_basketball("ncaa", color_updates = list(
  plot_background = "#ffffff00",
  offensive_half_court = "#e8e0d7",
  court_apron = "#e84a27",
  three_point_line = c("#13294b", "#ffffff"),
  painted_area = c("#e84a27", "#ffffff00")))
```

Keys come from `cani_color_league_features()`. Features drawn as line + fill take a length-2 color vector; 8-digit hex gives transparency (`"#ffffff00"` = fully transparent). `plot_background` colors the figure behind the surface.

PYTHON MIRROR & DATA HOOKS

sportypy mirrors the API: classes like `sportypy.surfaces.hockey.NHLRink().draw()` over matplotlib — same leagues, same feature names (sportypy.sportsdataverse.org).

SportsDataverse coordinate columns to overlay: **hoopR** / **wehoop** ESPN pbp `coordinate_x coordinate_y`; **fastRhockey** pbp `x_coord y_coord`; **baseballr** Statcast `hc_x hc_y`; NFL Big Data Bowl tracking `x y` (see page 1 for the `x_trans` / `y_trans` alignment recipe).