Mini Project 2

Breckin Hadley and Kobe Kirk

One team function

```
# creates a function to extract text from HTML nodes
safe_get <- function(page, css_selector, index = 1) {
  vals <- page |> # selects HTML nodes from page
    html_nodes(css_selector) |> # extracts nodes using CSS selector
    html_text(trim = TRUE) # extracts and trims text
# returns NA if selector doesn't return enough values, otherwise returns the desired value
  if (length(vals) < index || length(vals) == 0) return(NA_character_)
  vals[index]
}
# function to scrape team advanceed stats
seahawks <- function(url) {
  Sys.sleep(2) # this pauses between requests, to be polite
  session <- bow(url, force = TRUE) # bows to create a session, also to be polite
  page <- scrape(session) # scrapes page HTML
# extracts team and specific stats using selector gadget
  team <- "SEA"
  pass_attempts <- safe_get(page, "tfoot .right:nth-child(8)")
  rush_attempts <- safe_get(page, "#adv_rushing tfoot .right:nth-child(7)")
  indendend_air_yard <- safe_get(page, "tfoot .right:nth-child(9)")
  on_target <- safe_get(page, "tfoot .right:nth-child(24)")
  rush_before_contact <- safe_get(page, "tfoot .right:nth-child(11)")
  win_pct <- 0.75

# returns the results as a tibble
  tibble(
    team,
    pass_attempts,
    rush_attempts,
    indendend_air_yard,
    on_target,
```

```
    rush_before_contact,
    win_pct

  )
}
# runs seahawks function to scrape data and create tibble from provided url
seahawks("https://www.pro-football-reference.com/teams/sea/2025_advanced.htm")
```

```
No encoding supplied: defaulting to UTF-8.
```

```
# A tibble: 1 x 7
  team  pass_attempts rush_attempts indendend_air_yard on_target
  <chr> <chr>         <chr>         <chr>              <chr>
1 SEA   220           226           1979               80.7
# i 2 more variables: rush_before_contact <chr>, win_pct <dbl>
```

One Team Function 2

```
# creates a function to extract text from HTML nodes
safe_get <- function(page, css_selector, index = 1) {
  vals <- page |>
    html_nodes(css_selector) |>
    html_text(trim = TRUE)
  # returns NA if selector doesn't return enough values, otherwise returns the desired value
  if (length(vals) < index || length(vals) == 0) return(NA_character_)
  vals[index]
}
# function to scrape team record and points scored
seahawks2 <- function(url) {
  Sys.sleep(2)
  session <- bow(url, force = TRUE)
  page <- scrape(session)
# extracts team, and record and points scored using selector gadget
  team <- "SEA"
  record <- safe_get(page, "tfoot .left+ .right")
  points_scored <- safe_get(page, "tfoot .right:nth-child(9)")

# returns the results as a tibble
  tibble(
    team,
    record,
```

```
    points_scored
  )
}
# runs seahawks2 function to scrape data and create tibble from provided url
seahawks2("https://www.pro-football-reference.com/teams/sea/2024/gamelog/")
```

```
# A tibble: 1 x 3
  team  record points_scored
  <chr> <chr>  <chr>
1 SEA   10-7   375
```

For loop

```
library(tidyverse)
library(glue)
library(rvest)
library(polite)

# creates a function to extract text from HTML nodes
safe_get <- function(page, css_selector, index = 1) {
  vals <- page |>
    html_nodes(css_selector) |>
    html_text(trim = TRUE)
  # returns NA if selector doesn't return enough values, otherwise returns the desired value
  if (length(vals) < index || length(vals) == 0) return(NA_character_)
  vals[index]
}

# extracts team advanced stats
team_stats <- function(team_abbr, year = 2025) {
  url <- glue("https://www.pro-football-reference.com/teams/{team_abbr}/{year}_advanced.htm")

  Sys.sleep(2) # this pauses between requests, to be polite
  session <- bow(url, force = TRUE) # bows to create a session, also to be polite
  page <- scrape(session) # scrapes page HTML
# fixes hidden HTML comments so tables can be read correctly
  html_raw <- as.character(page)
  page_fixed <- read_html(gsub("<!--|-->", "", html_raw))
# extracts team abbreviation and specific stats using selector gadget
  team <- toupper(team_abbr)
  pass_attempts <- safe_get(page_fixed, "tfoot .right:nth-child(8)")
```

```
  rush_attempts <- safe_get(page_fixed, "#adv_rushing tfoot .right:nth-child(7)")
  indendend_air_yard <- safe_get(page_fixed, "tfoot .right:nth-child(9)")
  on_target <- safe_get(page_fixed, "tfoot .right:nth-child(24)")
  rush_before_contact <- safe_get(page_fixed, "#adv_rushing tfoot .right:nth-child(11)")
# returns results as a tibble
  tibble(
    team,
    year,
    pass_attempts,
    rush_attempts,
    indendend_air_yard,
    on_target,
    rush_before_contact
  )
}

# creates a vector of all NFL team abbreviations
nfl_teams <- c(
  "crd", "atl", "rav", "buf", "car", "chi", "cin", "cle",
  "dal", "den", "det", "gnb", "htx", "clt", "jax", "kan",
  "rai", "sdg", "ram", "mia", "min", "nwe", "nor", "nyg",
  "nyj", "phi", "pit", "sfo", "sea", "tam", "oti", "was"
)

# defines the years to scrape
years <- 2020:2024
teams <- nfl_teams
# generates all team-year combinations and scrapes stats
nfl_tbl <- expand_grid(team_abbr = teams, year = years) |> # all team-year combinations
  mutate(data = map2(team_abbr, year, team_stats)) |> # scrape each page
  pull(data) |>  # extracts scraped data
  list_rbind()  # combines all of the data into one tibble
```

```
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
```

```
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
```

```
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
```

```
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
```

```
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
No encoding supplied: defaulting to UTF-8.
```

```
nfl_tbl # view the data set
```

```
# A tibble: 160 x 7
   team   year pass_attempts rush_attempts indendend_air_yard on_target
   <chr> <int> <chr>         <chr>         <chr>              <chr>
 1 CRD    2020 575           479           4460               77.3
 2 CRD    2021 591           496           4459               80.0
 3 CRD    2022 664           434           4599               76.5
 4 CRD    2023 555           471           4193               70.9
 5 CRD    2024 543           463           3673               79.2
 6 ATL    2020 628           409           5412               75.5
 7 ATL    2021 573           393           4127               77.1
 8 ATL    2022 415           559           4011               71.9
 9 ATL    2023 530           522           4345               73.9
10 ATL    2024 559           495           4468               75.7
# i 150 more rows
# i 1 more variable: rush_before_contact <chr>
```

For loop 2

```r
# creates a function to extract text from HTML nodes
safe_get <- function(page, css_selector, index = 1) {
  vals <- page |>
    html_nodes(css_selector) |>
    html_text(trim = TRUE)
 # returns NA if selector doesn't return enough values, otherwise returns the desired value
  if (length(vals) < index || length(vals) == 0) return(NA_character_)
  vals[index]
}
# extracts team stats
team_stats2 <- function(team_abbr, year = 2024) {
  url <- glue("https://www.pro-football-reference.com/teams/{team_abbr}/{year}/gamelog/") |>

  Sys.sleep(2) # this pauses between requests, to be polite
  session <- bow(url, force = TRUE) # bows to create a session, also to be polite
  page <- scrape(session) # scrapes page HTML
# fixes hidden HTML comments so tables can be read correctly
  html_raw <- as.character(page)
  page_fixed <- read_html(gsub("<!--|-->", "", html_raw))
# extracts team abbreviation and specific stats using selector gadget
  team <- toupper(team_abbr)
  record <- safe_get(page, "tfoot .left+ .right")
  points_scored <- safe_get(page, "tfoot .right:nth-child(9)")
# returns results as a tibble
  tibble(
    team,
    year,
    record,
    points_scored
  )
}
# creates a vector of all NFL team abbreviations
nfl_teams <- c(
  "crd", "atl", "rav", "buf", "car", "chi", "cin", "cle",
  "dal", "den", "det", "gnb", "htx", "clt", "jax", "kan",
  "rai", "sdg", "ram", "mia", "min", "nwe", "nor", "nyg",
  "nyj", "phi", "pit", "sfo", "sea", "tam", "oti", "was"
)
# defines the years to scrape
years <- 2020:2024
teams <- nfl_teams
# generates all team-year combinations and scrapes stats
```

```r
nfl_tbl2 <- expand_grid(team_abbr = teams, year = years) |> # all team-year combinations
  mutate(data = map2(team_abbr, year, team_stats2)) |> # scrape each page
  pull(data) |> # extracts scraped data
  list_rbind() # combines all of the data into one tibble

nfl_tbl2 # view the data set
```

```
# A tibble: 160 x 4
   team   year record points_scored
   <chr> <int> <chr>  <chr>
 1 CRD    2020 8-8    410
 2 CRD    2021 11-6   449
 3 CRD    2022 4-13   340
 4 CRD    2023 4-13   330
 5 CRD    2024 8-9    400
 6 ATL    2020 4-12   396
 7 ATL    2021 7-10   313
 8 ATL    2022 7-10   365
 9 ATL    2023 7-10   321
10 ATL    2024 8-9    389
# i 150 more rows
```

```r
# merges the two tables by team name
full_table <- left_join(nfl_tbl,nfl_tbl2, join_by(team,year))
full_table # view final data set
```

```
# A tibble: 160 x 9
   team   year pass_attempts rush_attempts indendend_air_yard on_target
   <chr> <int> <chr>         <chr>         <chr>               <chr>
 1 CRD    2020 575           479           4460                77.3
 2 CRD    2021 591           496           4459                80.0
 3 CRD    2022 664           434           4599                76.5
 4 CRD    2023 555           471           4193                70.9
 5 CRD    2024 543           463           3673                79.2
 6 ATL    2020 628           409           5412                75.5
 7 ATL    2021 573           393           4127                77.1
 8 ATL    2022 415           559           4011                71.9
 9 ATL    2023 530           522           4345                73.9
10 ATL    2024 559           495           4468                75.7
# i 150 more rows
# i 3 more variables: rush_before_contact <chr>, record <chr>,
#   points_scored <chr>
```