Date-Time Data Analysis in R

Habtamu Bizuayehu

Table of contents

# 1. **Overview of Presentation Contents**

This presentation introduces essential techniques for analyzing date and time data in health research using R. Key topics include:

* **Importance of date/time analysis**
* **Core R packages** for handling dates (e.g., lubridate, tidyverse)
* **Date conversion and formatting**
* **Calculating durations and age**
* **Extracting date components** (e.g., month, weekday, season)
* **Summary and Visualize time-based trends**

**1. Importance of Date/Time Analysis in Health Data**

Date-time data is crucial in healthcare because it allows us to:

* Calculate accurate patient ages, service time
* Measure time intervals between critical events (e.g., diagnosis to treatment, age vaccination)
* Identify seasonal patterns in disease occurrence
* Track vaccine effectiveness over time
* Determine follow-up periods for clinical studies

# 2. **Load required libraries**

library(lubridate) # Simplifies the manipulation of dates and times in R (e.g., formatting, extracting components)  
library(dplyr) # Provides tools for data manipulation, helpful for filtering and summarizing date-based data  
library(stringr) # Simplifies string manipulation  
  
library(zoo) # For working with time series data, including rolling calculations and handling missing data in time series  
library(tsibble) # Provides tools for handling time series data, including date-time indexes and features for forecasting  
  
library(readr) # For reading date-time data from CSV  
library(haven) # data from other statistical software formats (SPSS, SAS, Stata)  
library(forecast) # Useful for time series forecasting, especially when working with seasonal or trend-based data  
library(ggplot2) # For visualizing date-time trends in data (e.g., time series plots)  
library(janitor) # Tabulation, cleaning column names, adding totals and proportions  
library (plotly) # Interactive visualizations  
  
# Documentation/reporting  
library(prettydoc) # Pretty document templates  
library(flexdashboard)# Interactive dashboards  
library(knitr) # Dynamic report generation  
library(quarto) # For rendering and publishing documents with the Quarto framework  
library(rmarkdown) # R Markdown document processing  
library(yaml) # YAML document processing  
  
# Tabulation  
library(flextable) # Flexible table formatting  
library(gt) # Grammar of tables  
library(reactable) # Interactive tables

setwd("C:/Users/User/Desktop/Materials\_ Course and proposals/Course Related/DataCamp/Data/Synthetic\_data")  
  
patients <- read.csv("patients.csv")  
vaccination <- read.csv("immunizations.csv")

# Convert column names to lowercase  
colnames(patients) <- tolower(colnames(patients))  
colnames(vaccination) <- tolower(colnames(vaccination))  
  
# View data  
# head(patients)  
# head(vaccination)  
  
str(patients)

'data.frame': 106 obs. of 28 variables:  
 $ id : chr "30a6452c-4297-a1ac-977a-6a23237c7b46" "34a4dcc4-35fb-6ad5-ab98-be285c586a4f" "7179458e-d6e3-c723-2530-d4acfe1c2668" "37c177ea-4398-fb7a-29fa-70eb3d673876" ...  
 $ birthdate : chr "1994-02-06" "1968-08-06" "2008-12-21" "1994-01-27" ...  
 $ deathdate : chr "" "2009-12-11" "" "" ...  
 $ ssn : chr "999-52-8591" "999-75-3953" "999-70-1925" "999-27-9779" ...  
 $ drivers : chr "S99996852" "S99993577" "" "S99995100" ...  
 $ passport : chr "X47758697X" "X28173268X" "" "X83694889X" ...  
 $ prefix : chr "Mr." "Mr." "" "Mrs." ...  
 $ first : chr "Joshua658" "Bennie663" "Hunter736" "Carlyn477" ...  
 $ middle : chr "Alvin56" "" "Mckinley734" "Florencia449" ...  
 $ last : chr "Kunde533" "Ebert178" "Gerlach374" "Williamson769" ...  
 $ suffix : logi NA NA NA NA NA NA ...  
 $ maiden : chr "" "" "" "Rogahn59" ...  
 $ marital : chr "M" "D" "" "M" ...  
 $ race : chr "white" "white" "white" "asian" ...  
 $ ethnicity : chr "nonhispanic" "nonhispanic" "nonhispanic" "nonhispanic" ...  
 $ gender : chr "M" "M" "M" "F" ...  
 $ birthplace : chr "Boston Massachusetts US" "Chicopee Massachusetts US" "Spencer Massachusetts US" "Franklin Massachusetts US" ...  
 $ address : chr "811 Kihn Viaduct" "975 Pfannerstill Throughway" "548 Heller Lane" "160 Fadel Crossroad Apt 65" ...  
 $ city : chr "Braintree" "Braintree" "Mattapoisett" "Wareham" ...  
 $ state : chr "Massachusetts" "Massachusetts" "Massachusetts" "Massachusetts" ...  
 $ county : chr "Norfolk County" "Norfolk County" "Plymouth County" "Plymouth County" ...  
 $ fips : int 25021 25021 NA NA NA 25017 25017 25021 25009 NA ...  
 $ zip : int 2184 2184 0 0 0 2138 2476 2184 1835 0 ...  
 $ lat : num 42.2 42.3 41.6 41.8 42.7 ...  
 $ lon : num -71 -71 -70.9 -70.7 -71 ...  
 $ healthcare\_expenses: num 56905 124024 45645 12895 18500 ...  
 $ healthcare\_coverage: num 18020 1075 6155 659952 5494 ...  
 $ income : int 100511 49737 133816 17382 52159 75767 58294 49737 77756 35255 ...

str(vaccination)

'data.frame': 1619 obs. of 6 variables:  
 $ date : chr "2016-04-10T09:04:48Z" "2016-04-10T09:04:48Z" "2016-04-10T09:04:48Z" "2016-04-10T09:04:48Z" ...  
 $ patient : chr "30a6452c-4297-a1ac-977a-6a23237c7b46" "30a6452c-4297-a1ac-977a-6a23237c7b46" "30a6452c-4297-a1ac-977a-6a23237c7b46" "30a6452c-4297-a1ac-977a-6a23237c7b46" ...  
 $ encounter : chr "0b03e41b-06a6-66fa-b972-acc5a83b134a" "0b03e41b-06a6-66fa-b972-acc5a83b134a" "0b03e41b-06a6-66fa-b972-acc5a83b134a" "0b03e41b-06a6-66fa-b972-acc5a83b134a" ...  
 $ code : int 140 113 43 114 140 140 140 140 140 52 ...  
 $ description: chr "Influenza seasonal injectable preservative free" "Td (adult) 5 Lf tetanus toxoid preservative free adsorbed" "Hep B adult" "meningococcal MCV4P" ...  
 $ base\_cost : num 136 136 136 136 136 136 136 136 136 136 ...

# Check for missing IDs  
patients$id[patients$id == ""] <- NA  
print(sum(is.na(patients$id)))

[1] 0

# Check duplicates  
patients <- patients %>%  
 arrange(id) %>%  
 group\_by(id) %>%  
 mutate(dup = row\_number()) %>%  
 ungroup()  
  
table(patients$dup) # Check if any duplicates exist

1   
106

# Merge with vaccination data  
vaccination <- vaccination %>% mutate(id = patient)  
  
  
vacc\_pt\_merged <- left\_join(patients, vaccination, by = "id" )  
  
#vacc\_pt\_merged <- left\_join(patients, vaccination, by = c("id" = "patient"))  
  
str (vacc\_pt\_merged)

# 3. Converting Character to Date Format

in the dataset birthdate, deathdate, and date (vaccination date) are all stored as character strings. I need to convert these to proper date formats using lubridate::ymd().

# Convert to date format  
str (vacc\_pt\_merged$birthdate)

chr [1:1619] "1975-12-24" "1975-12-24" "1975-12-24" "1975-12-24" ...

str (vacc\_pt\_merged$date) # date is YYYY-MM-DD HH:MM:SS format.

chr [1:1619] "2015-06-24T14:05:28Z" "2016-06-29T14:05:28Z" ...

vacc\_pt\_merged <- vacc\_pt\_merged %>%  
 mutate(  
 birthdate = parse\_date(birthdate, format = "%Y-%m-%d"),  
 vacc\_date = as.Date(strptime(date, format = "%Y-%m-%dT%H:%M:%SZ"))  
 )  
  
# Check structure  
str (vacc\_pt\_merged$birthdate)

Date[1:1619], format: "1975-12-24" "1975-12-24" "1975-12-24" "1975-12-24" "1975-12-24" ...

str (vacc\_pt\_merged$vacc\_date)

Date[1:1619], format: "2015-06-24" "2016-06-29" "2016-06-29" "2017-07-05" "2017-07-05" ...

# Check converted formats via glimpse  
# glimpse(vacc\_pt\_merged)  
  
# no missing values on dates  
sum(is.na(vacc\_pt\_merged$birthdate))

[1] 0

sum(is.na(vacc\_pt\_merged$vacc\_date))

[1] 0

# Check date ranges  
summary(vacc\_pt\_merged$birthdate)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
"1914-03-03" "1969-12-09" "1999-02-24" "1990-12-03" "2012-09-02" "2023-03-01"

summary(vacc\_pt\_merged$vacc\_date)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
"1962-12-09" "2017-05-21" "2020-01-04" "2019-04-29" "2022-01-26" "2024-10-26"

Now date columns are in Date format (YYYY-MM-DD), ready for analysis!

# 4. **Age and Service year Calculation**

Age at the time of vaccination is essential. Here’s how I calculate it in **years**.

vacc\_pt\_merged <- vacc\_pt\_merged %>%  
 mutate(  
 age\_years = floor (as.numeric(difftime(Sys.Date(), birthdate, units = "days")) / 365.25),  
   
 # Calculate vaccination service year by extracting the year from vacc\_date  
 vacc\_service\_year = year(vacc\_date)  
 )  
  
  
summary(vacc\_pt\_merged$age\_years)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 2.00 12.00 26.00 33.86 55.00 111.00

summary(vacc\_pt\_merged$vacc\_service\_year)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 1962 2017 2020 2019 2022 2024

# tabulation  
table(vacc\_pt\_merged$age\_years)

2 5 7 8 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 26   
 70 64 66 34 102 38 59 21 21 17 64 36 36 36 34 48 13 12 15 32   
 27 28 29 30 31 33 34 36 38 40 41 42 43 44 45 46 48 49 50 51   
 17 30 9 24 14 4 7 6 14 8 49 16 9 8 9 28 30 30 15 13   
 52 53 54 55 56 57 58 61 62 63 64 65 68 69 70 71 73 74 76 79   
 15 15 16 30 18 12 14 26 49 26 26 14 14 42 15 15 27 15 14 12   
 80 85 98 111   
 13 12 20 1

table(vacc\_pt\_merged$vacc\_service\_year)

1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 2001 2002 2003 2004 2005 2006   
 1 3 1 1 1 2 1 1 1 1 1 2 1 1 2 1   
2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022   
 3 6 2 3 2 4 3 22 153 139 137 151 161 137 263 120   
2023 2024   
 170 122

As shown above, the **average age at vaccination** was approximately 33.9 years, with a range from 2 to 111 years.

The **highest frequency of vaccinations** occurred in the year 2021, suggesting a peak in service delivery during that period.

# 5. **Calculating Time Differences**

To calculate time differences, I use interval() or simple subtraction between date objects.

Example: Years between birth and vaccination gives the **number of years from birth to vaccination**.

vacc\_pt\_merged <- vacc\_pt\_merged %>%  
 mutate(  
 days\_to\_vax = as.numeric(vacc\_date - birthdate), # Difference in days  
 weeks\_to\_vax = as.numeric(difftime(vacc\_date, birthdate, units = "weeks")), # Difference in weeks  
 months\_to\_vax = as.numeric(difftime(vacc\_date, birthdate, units = "days")) / 30.44, # Approximate months (average days per month)  
 years\_to\_vax = floor(as.numeric(difftime(vacc\_date, birthdate, units = "days")) / 365.25) # Approximate years (including leap years)  
 )  
  
  
# View summary statistics   
summary(vacc\_pt\_merged$days\_to\_vax)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 0 2548 7742 10375 18263 36813

summary(vacc\_pt\_merged$weeks\_to\_vax)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 0 364 1106 1482 2609 5259

summary(vacc\_pt\_merged$months\_to\_vax)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 0.00 83.71 254.34 340.82 599.97 1209.36

summary(vacc\_pt\_merged$years\_to\_vax)

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 0.00 6.00 21.00 28.02 50.00 100.00

# 6. **Extracting Date Components**

We will perform the following extractions for both the birthdate and vaccination date:

1. **Day**: The day of the month is extracted using the day() function.
2. **Month**: The month of the date is obtained with the month() function. This provides a numeric value (1 for January, 12 for December).
3. **Year**: The year component is retrieved using the year() function, returning a four-digit value for the year.
4. **Quarter**: The quarter of the year is extracted using the quarter() function, providing a numeric value from 1 to 4, representing the first through fourth quarter of the year.
5. **Season**: Based on the month, we assign a corresponding season (Summer, Autumn, Winter, or Spring) using a conditional case\_when() function:

These components are useful for segmenting the data, performing seasonal analysis, and understanding how different periods influence the patterns observed in the data. For instance, analyzing vaccination trends by quarter or season can provide valuable insights into public health strategies.

By extracting these components, we are preparing the data for more granular analysis, including cohort analysis, seasonal trend assessments, and other forms of time-based grouping.

vacc\_pt\_merged <- vacc\_pt\_merged %>%  
 mutate(  
 # Basic date components  
 vacc\_day = day(vacc\_date),  
 vacc\_month = month(vacc\_date),  
 vacc\_month\_name = month(vacc\_date, label = TRUE, abbr = FALSE), # Full month name  
 vacc\_year = year(vacc\_date),  
 vacc\_weekday = wday(vacc\_date, label = TRUE, abbr = FALSE), # Full weekday name  
 vacc\_week = week(vacc\_date), # Week of year  
 vacc\_quarter = quarter(vacc\_date),  
   
 # Fiscal year (assuming June start)  
 vacc\_fy = ifelse(vacc\_month >= 6, vacc\_year, vacc\_year - 1),  
   
 # Season (Northern Hemisphere)  
 vacc\_season = case\_when(  
 vacc\_month %in% 3:5 ~ "Spring",  
 vacc\_month %in% 6:8 ~ "Summer",  
 vacc\_month %in% 9:11 ~ "Autumn",  
 TRUE ~ "Winter" # December-February  
 ),  
   
 # Weekend flag  
 is\_weekend = wday(vacc\_date) %in% c(1, 7), # 1=Sunday, 7=Saturday  
   
 # Days since specific reference date (e.g., pandemic start)  
 days\_since\_ref = floor(as.numeric(vacc\_date - as.Date("2020-01-01")))  
 )

# 7. Filtering Between Date Ranges

Filtering data based on specific date ranges is a common operation in data analysis, particularly when focusing on subsets of data that fall within particular periods. In R, the dplyr package provides an intuitive way to filter rows within a specific date range.

Below is an example of how to filter rows based on dates that fall between two specified dates using the filter() function:

# Filter only vaccinations in 2022  
vacc\_2022 <- vacc\_pt\_merged %>%  
 filter(vacc\_date >= ymd("2022-01-01") & vacc\_date <= ymd("2022-12-31"))  
  
# Filter vaccinations between 2015 and 2024  
vacc\_1524 <- vacc\_pt\_merged %>%  
 filter(vacc\_service\_year >= 2015 & vacc\_service\_year <= 2024)  
   
  
table(vacc\_1524$vacc\_year)

2015 2016 2017 2018 2019 2020 2021 2022 2023 2024   
 153 139 137 151 161 137 263 120 170 122

table(vacc\_1524$race)

asian black native other white   
 111 94 13 29 1306

# table(vacc\_1524$ethnicity)

# ordering variables for visual checking   
vacc\_1524 <- vacc\_1524 %>%   
 select(id, gender, race, birthdate, date, vacc\_date, age\_years, vacc\_service\_year, days\_to\_vax, weeks\_to\_vax,   
 months\_to\_vax, years\_to\_vax, vacc\_day, vacc\_month, vacc\_month\_name, vacc\_year, vacc\_weekday, vacc\_week,   
 vacc\_quarter, vacc\_fy, vacc\_season, is\_weekend, days\_since\_ref, everything())

# 8. Summary of vaccination pattern: Tabulation ———————————–

# Recode gender  
vacc\_1524 <- vacc\_1524 %>%  
 mutate(gender = case\_when(  
 gender == "F" ~ "Female",  
 gender == "M" ~ "Male",  
 TRUE ~ gender  
 ))  
  
# Convert time variables to character for consistency  
vacc\_1524 <- vacc\_1524 %>%  
 mutate(across(c(vacc\_service\_year, vacc\_quarter, vacc\_weekday), as.character))  
  
# Create and label each table  
tab\_year <- vacc\_1524 %>%  
 tabyl(vacc\_service\_year, gender) %>%  
 adorn\_totals("row") %>%  
 mutate(Category = "Year", .before = 1) %>%  
 rename(Label = vacc\_service\_year)  
  
tab\_quarter <- vacc\_1524 %>%  
 tabyl(vacc\_quarter, gender) %>%  
 adorn\_totals("row") %>%  
 mutate(Category = "Quarter", .before = 1) %>%  
 rename(Label = vacc\_quarter)  
  
tab\_weekday <- vacc\_1524 %>%  
 tabyl(vacc\_weekday, gender) %>%  
 adorn\_totals("row") %>%  
 mutate(Category = "Weekday", .before = 1) %>%  
 rename(Label = vacc\_weekday)  
  
# Combine all  
tab\_combined <- bind\_rows(tab\_year, tab\_quarter, tab\_weekday)  
  
# Calculate column proportions within each Category group (excluding "Total" row)  
tab\_combined <- tab\_combined %>%  
 group\_by(Category) %>%  
 mutate(  
 Female\_Prop = round(100 \* Female / sum(Female[Label != "Total"]), 1),  
 Male\_Prop = round(100 \* Male / sum(Male[Label != "Total"]), 1)  
 ) %>%  
 ungroup()  
  
# Format with flextable  
ft <- flextable(tab\_combined) %>%  
 set\_header\_labels(  
 Category = "Time Unit",  
 Label = "Category",  
 Female = "Female Count",  
 Male = "Male Count",  
 Female\_Prop = "Female %",  
 Male\_Prop = "Male %"  
 ) %>%  
 add\_header\_lines(values = "Table: Vaccination counts and gender proportions (%) across Year, Quarter, and Weekday") %>%  
 theme\_box() %>%  
 autofit()  
  
ft

| **Table: Vaccination counts and gender proportions (%) across Year, Quarter, and Weekday** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Time Unit** | **Category** | **Female Count** | **Male Count** | **Female %** | **Male %** |
| Year | 2015 | 74 | 79 | 9.3 | 10.5 |
| Year | 2016 | 77 | 62 | 9.6 | 8.2 |
| Year | 2017 | 65 | 72 | 8.1 | 9.5 |
| Year | 2018 | 97 | 54 | 12.2 | 7.2 |
| Year | 2019 | 78 | 83 | 9.8 | 11.0 |
| Year | 2020 | 61 | 76 | 7.6 | 10.1 |
| Year | 2021 | 146 | 117 | 18.3 | 15.5 |
| Year | 2022 | 61 | 59 | 7.6 | 7.8 |
| Year | 2023 | 82 | 88 | 10.3 | 11.7 |
| Year | 2024 | 57 | 65 | 7.1 | 8.6 |
| Year | Total | 798 | 755 | 100.0 | 100.0 |
| Quarter | 1 | 174 | 229 | 21.8 | 30.3 |
| Quarter | 2 | 230 | 146 | 28.8 | 19.3 |
| Quarter | 3 | 205 | 262 | 25.7 | 34.7 |
| Quarter | 4 | 189 | 118 | 23.7 | 15.6 |
| Quarter | Total | 798 | 755 | 100.0 | 100.0 |
| Weekday | Friday | 69 | 111 | 8.6 | 14.7 |
| Weekday | Monday | 117 | 60 | 14.7 | 7.9 |
| Weekday | Saturday | 171 | 82 | 21.4 | 10.9 |
| Weekday | Sunday | 153 | 124 | 19.2 | 16.4 |
| Weekday | Thursday | 118 | 135 | 14.8 | 17.9 |
| Weekday | Tuesday | 79 | 86 | 9.9 | 11.4 |
| Weekday | Wednesday | 91 | 157 | 11.4 | 20.8 |
| Weekday | Total | 798 | 755 | 100.0 | 100.0 |

📝 **Summary of Vaccination Uptake by Gender and Time Period**

The table Aabove presents vaccination counts and gender proportions across different time units—**year, quarter, and weekday**—for individuals aged 15–24 years.

#### 8.0.0.1 **By Year**

* **2021** recorded the **highest vaccination uptake** for both **females (146; 18.3%)** and **males (117; 15.5%)**, indicating a peak in vaccine distribution during that year.
* The **lowest uptake** was observed in **2017 for females (65; 8.1%)** and in **2016 for males (62; 8.2%)**.
* Overall, vaccination was more evenly distributed across years, but there was a noticeable increase in 2021, likely due to public health initiatives during the COVID-19 pandemic.

#### 8.0.0.2 **By Quarter**

* For **females**, the highest uptake was in **Quarter 2 (230; 28.8%)**, followed by **Quarter 3 (205; 25.7%)**.
* For **males**, **Quarter 3 (262; 34.7%)** had the highest uptake, while **Quarter 2 (146; 19.3%)** was lower in comparison.
* **Quarter 1** showed the **lowest female uptake proportionally (21.8%)**, while **Quarter 4** had the lowest for males (15.6%).

#### 8.0.0.3 **By Weekday**

* **Wednesday (157; 20.8%)** was the peak for males.
* **Friday (14.7%)** and **Thursday (17.9%)** were more popular among males.
* **Monday had the lowest male uptake (7.9%)**, while **Tuesday (9.9%)** was the lowest for females.

# 9. 📊 Visualizing Vaccination Trends

Time-based visualizations help us uncover trends, seasonality, and variation in vaccination data. Below are some examples of how to visualize date information using the vacc\_pt\_merged dataset. These plots support evidence-based insights and can guide public health decision-making.

#### 9.0.0.1 **Overall Vaccination Patterns by by Gender and Race**

* **2021** was a notable year across most racial groups, with **peaks in vaccination uptake**, likely due to intensified COVID-19 vaccine rollouts.
* The **“Other” racial group** showed the **most fluctuation** with a sharp spike in 2023.
* The **White group** had the **most consistent trend**, while **Asian and Black groups** showed spikes at specific periods.
* **Females showed a sharper peak** than males in 2021 but also had a **steeper decline afterward**.
* **Males had more consistent trends overall**, especially from 2015–2020 and 2022–2024.
* The **post-2021 decline** in both groups may suggest reduced urgency or campaign intensity in recent years
* Post-2021, a general **decline across all groups** suggests possible vaccine fatigue or the end of mass campaign periods.
* Variations of vaccination uptake by gender and race are evident, suggesting differing levels of access, trust, or outreach effectiveness across communities.

### 9.0.1 Monthly Vaccination Trend in each Year

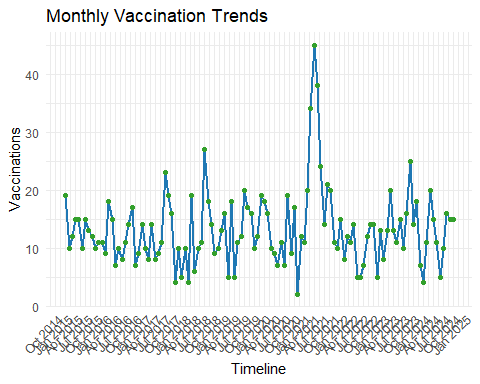
Use a line plot to visualize how the number of vaccinations changes over time.

The time series reveals longitudinal patterns, showing potential campaign effects (peaks) or service interruptions (troughs). The upward trend in mid 2021 suggests successful program for vaccination.

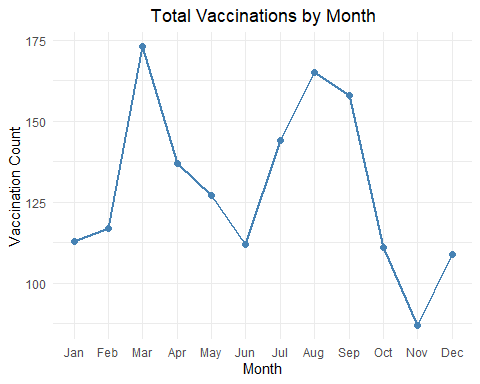
As shown in the monthly vaccination pattern,

* A **bi-modal pattern** is evident with two peaks (March and August) and two noticeable dips (June and November).
* **March and August** are peak months — possibly due to public health campaigns.
* **November** was the lowest — possibly due to holiday distractions, reduced outreach, or end-of-year slowdowns.

vacc\_1524 %>%  
 count(vacc\_year, vacc\_month) %>%  
 ggplot(aes(x = as.Date(paste(vacc\_year, vacc\_month, "01", sep = "-")),   
 y = n)) +  
 geom\_line(color = "#1f78b4", linewidth = 1) +  
 geom\_point(color = "#33a02c") +  
 labs(title = "Monthly Vaccination Trends",  
 x = "Timeline", y = "Vaccinations") +  
 scale\_x\_date(date\_labels = "%b %Y", date\_breaks = "3 months") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



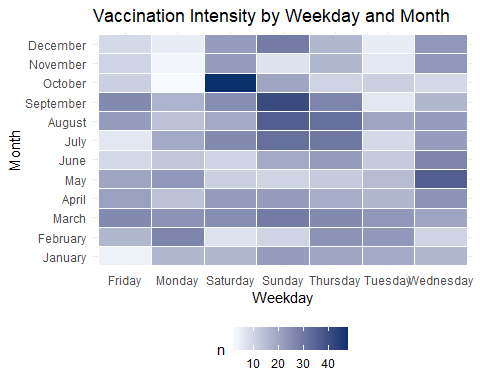
vacc\_1524 %>%  
 mutate(month = month(vacc\_date, label = TRUE, abbr = TRUE)) %>% # Extract month from date  
 group\_by(month) %>%  
 summarise(count = n(), .groups = "drop") %>%  
 ggplot(aes(x = month, y = count, group = 1)) +  
   
 geom\_line(linewidth = 1, color = "steelblue") +  
 geom\_point(size = 2, color = "steelblue") +  
 labs(  
 title = "Total Vaccinations by Month",  
 x = "Month",  
 y = "Vaccination Count"  
 ) +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(hjust = 0.5)  
 )



## 9.1 **Weekday Heatmap**

The heatmap identifies temporal service patterns. Higher intensity on Wednesdays/Thursdays may reflect clinic scheduling norms, while weekend gaps could indicate access barriers.

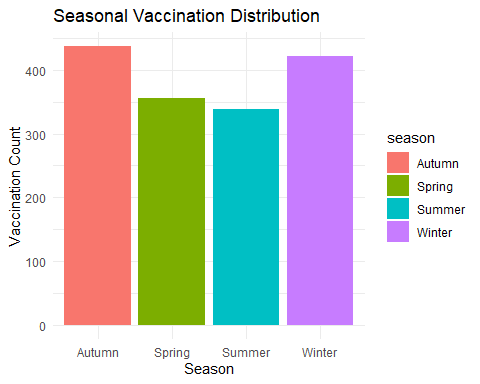
vacc\_1524 %>%  
 count(vacc\_month\_name, vacc\_weekday) %>%  
 ggplot(aes(x = vacc\_weekday, y = vacc\_month\_name, fill = n)) +  
 geom\_tile(color = "white") +  
 scale\_fill\_gradient(low = "#f7fbff", high = "#08306b") +  
 labs(title = "Vaccination Intensity by Weekday and Month",  
 x = "Weekday", y = "Month") +  
 theme\_minimal() +  
 theme(legend.position = "bottom")



## 9.2 **Seasonal Vaccination Distribution**

Explore how vaccination services vary by season.

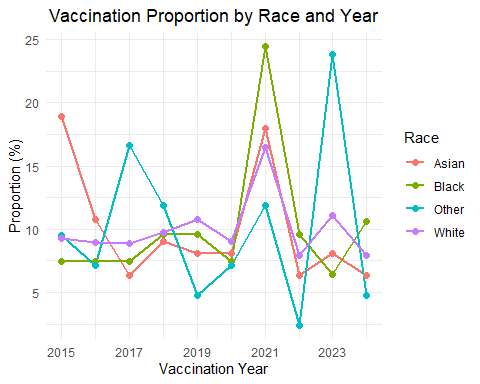
vacc\_1524 %>%  
 mutate(month = as.numeric(format(vacc\_date, "%m")),  
 season = case\_when(  
 month %in% c(12, 1, 2) ~ "Summer",  
 month %in% c(3, 4, 5) ~ "Autumn",  
 month %in% c(6, 7, 8) ~ "Winter",  
 month %in% c(9, 10, 11) ~ "Spring"  
 )) %>%  
 count(season) %>%  
 ggplot(aes(x = season, y = n, fill = season)) +  
 geom\_col() +  
 labs(  
 title = "Seasonal Vaccination Distribution",  
 x = "Season",  
 y = "Vaccination Count"  
 ) +  
 theme\_minimal()



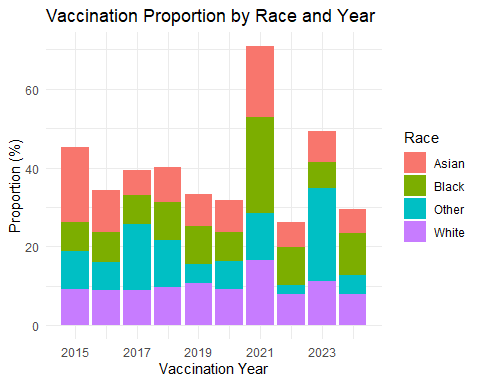
## 9.3 Vaccination proportion by year and race

If we’d like a stacked bar plot of the proportions

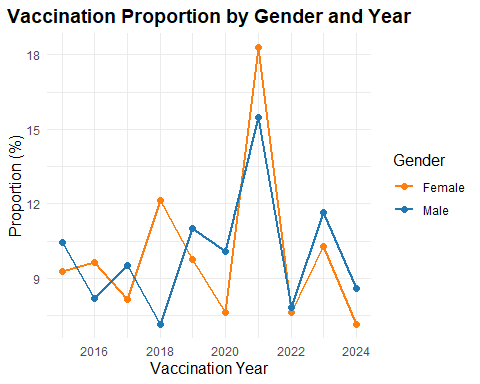
# Proportion by vacc\_year and race with cleaned race labels  
df\_plot\_race <- vacc\_1524 %>%  
 mutate(  
 race = ifelse(race %in% c("native", "other"), "Other", race),  
 race = str\_to\_title(race),  
 vacc\_year = as.integer(floor(as.numeric(vacc\_year))) # Ensure whole number  
 ) %>%  
 group\_by(race, vacc\_year) %>%  
 summarise(count = n(), .groups = "drop") %>%  
 group\_by(race) %>%  
 mutate(  
 total = sum(count),  
 proportion = (count / total) \* 100  
 ) %>%  
 ungroup()  
  
# Line plot  
ggplot(df\_plot\_race, aes(x = vacc\_year, y = proportion, color = race)) +  
 geom\_line(linewidth = 1) +  
 geom\_point(size = 2) +  
 scale\_x\_continuous(breaks = seq(min(df\_plot\_race$vacc\_year), max(df\_plot\_race$vacc\_year), by = 2)) +  
 labs(  
 title = "Vaccination Proportion by Race and Year",  
 x = "Vaccination Year",  
 y = "Proportion (%)",  
 color = "Race"  
 ) +  
 theme\_minimal() +  
 theme(  
 legend.position = "right",  
 plot.title = element\_text(hjust = 0.5)  
 )



# Bar plot  
ggplot(df\_plot\_race, aes(x = vacc\_year, y = proportion, fill = race)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 scale\_x\_continuous(breaks = seq(min(df\_plot\_race$vacc\_year), max(df\_plot\_race$vacc\_year), by = 2)) +  
 labs(  
 title = "Vaccination Proportion by Race and Year",  
 x = "Vaccination Year",  
 y = "Proportion (%)",  
 fill = "Race"  
 ) +  
 theme\_minimal()



# Proportion by vacc\_year and gender  
vacc\_1524 %>%  
 mutate(  
 gender = case\_when(  
 gender == "M" ~ "Male",  
 gender == "F" ~ "Female",  
 TRUE ~ gender  
 ),  
 vacc\_year = as.integer(vacc\_year) # Ensure year is whole number  
 ) %>%  
 group\_by(gender, vacc\_year) %>%  
 summarise(count = n(), .groups = "drop") %>%  
 group\_by(gender) %>%  
 mutate(  
 total = sum(count),  
 proportion = (count / total) \* 100  
 ) %>%  
 ungroup() %>%   
   
 ggplot(aes(x = vacc\_year, y = proportion, color = gender)) +  
 geom\_line(linewidth = 1) +  
 geom\_point(size = 2) +  
 scale\_x\_continuous(breaks = scales::pretty\_breaks()) + # Ensures whole number years  
 labs(  
 title = "Vaccination Proportion by Gender and Year",  
 x = "Vaccination Year",  
 y = "Proportion (%)",  
 color = "Gender"  
 ) +  
 scale\_color\_manual(values = c("Male" = "#1f77b4", "Female" = "#ff7f0e")) +  
 theme\_minimal() +  
 theme(  
 legend.position = "right",  
 plot.title = element\_text(hjust = 0.5, size = 14, face = "bold"),  
 axis.title = element\_text(size = 12),  
 legend.title = element\_text(size = 12)  
 )



**Summary and Key Takeaways**

Key techniques covered:

1. Proper date conversion from character strings
2. Accurate age and time difference calculations
3. Seasonal and temporal pattern analysis
4. Visualization of date-based trends
5. Survival analysis for time-to-event data

Best practices:

* Always validate date conversions
* Document time units clearly (days vs. months vs. years)
* Handle missing dates explicitly
* Consider time zones for multi-center studies

**References**

1. https://CRAN.R-project.org/package=dplyr
2. https://www.jstatsoft.org/article/view/v103i01
3. https://r4ds.hadley.nz/