

Strava Statistics

Jesse Hao

Last updated: 2025-01-14

```
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2     3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
df_raw <- read_csv("activities_12_28_2024.csv")
```

```
## New names:
## Rows: 371 Columns: 94
## -- Column specification
## ----- Delimiter: "," chr
## (6): Activity Date, Activity Name, Activity Type, Activity Description,... dbl
## (51): Activity ID, Elapsed Time...6, Distance...7, Max Heart Rate...8, R... lgl
## (37): Commute...10, Activity Private Note, Athlete Weight, Bike Weight, ...
## i Use 'spec()' to retrieve the full column specification for this data. i
## Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## * 'Elapsed Time' -> 'Elapsed Time...6'
## * 'Distance' -> 'Distance...7'
## * 'Max Heart Rate' -> 'Max Heart Rate...8'
## * 'Relative Effort' -> 'Relative Effort...9'
## * 'Commute' -> 'Commute...10'
## * 'Elapsed Time' -> 'Elapsed Time...16'
## * 'Distance' -> 'Distance...18'
## * 'Max Heart Rate' -> 'Max Heart Rate...31'
## * 'Relative Effort' -> 'Relative Effort...38'
## * 'Commute' -> 'Commute...51'
```

```
df_clean <- df_raw |>
  select(date = "Activity Date",
         name = "Activity Name",
         type = "Activity Type",
         description = "Activity Description",
```

```

    total_time = "Elapsed Time...6",
    time = "Moving Time",
    distance = "Distance...7",
    elevation = "Elevation Gain") |>
mutate(distance = distance * .621,
    total_time = total_time / 60,
    time = time / 60,
    pace = time / distance,
    elevation = if_else(is.na(elevation), 0, elevation),
    elevation = elevation * 3.281)

print(df_clean)

```

```

## # A tibble: 371 x 9
##   date       name type description total_time  time distance elevation  pace
##   <chr>      <chr> <chr> <chr>          <dbl> <dbl>    <dbl>    <dbl> <dbl>
## 1 Jun 8, 202~ Even~ Run  <NA>         50.0  30.1     3.32      454.  9.05
## 2 Jun 10, 20~ Even~ Run  <NA>         32.6  32.0     3.03      379. 10.5
## 3 Jun 11, 20~ Even~ Run  <NA>         18.1  17.9     2.00      268.  8.94
## 4 Jun 12, 20~ Morn~ Run  <NA>          8.62  8.62     1.05      160.  8.21
## 5 Jun 16, 20~ Even~ Run  <NA>         19.8  16.8     1.06         0 15.8
## 6 Jun 16, 20~ Even~ Run  <NA>         19.9  19.6     2.00         0  9.81
## 7 Jun 17, 20~ Even~ Run  <NA>          9.22  7.98     1.00         0  7.98
## 8 Jun 17, 20~ Even~ Run  <NA>          6.32  6.32     0.677         0  9.33
## 9 Jun 17, 20~ Even~ Run  <NA>         14.8  14.5     1.45         0  9.96
## 10 Jun 18, 20~ Even~ Run  <NA>         30.4  30.0     3.05         0  9.86
## # i 361 more rows

```

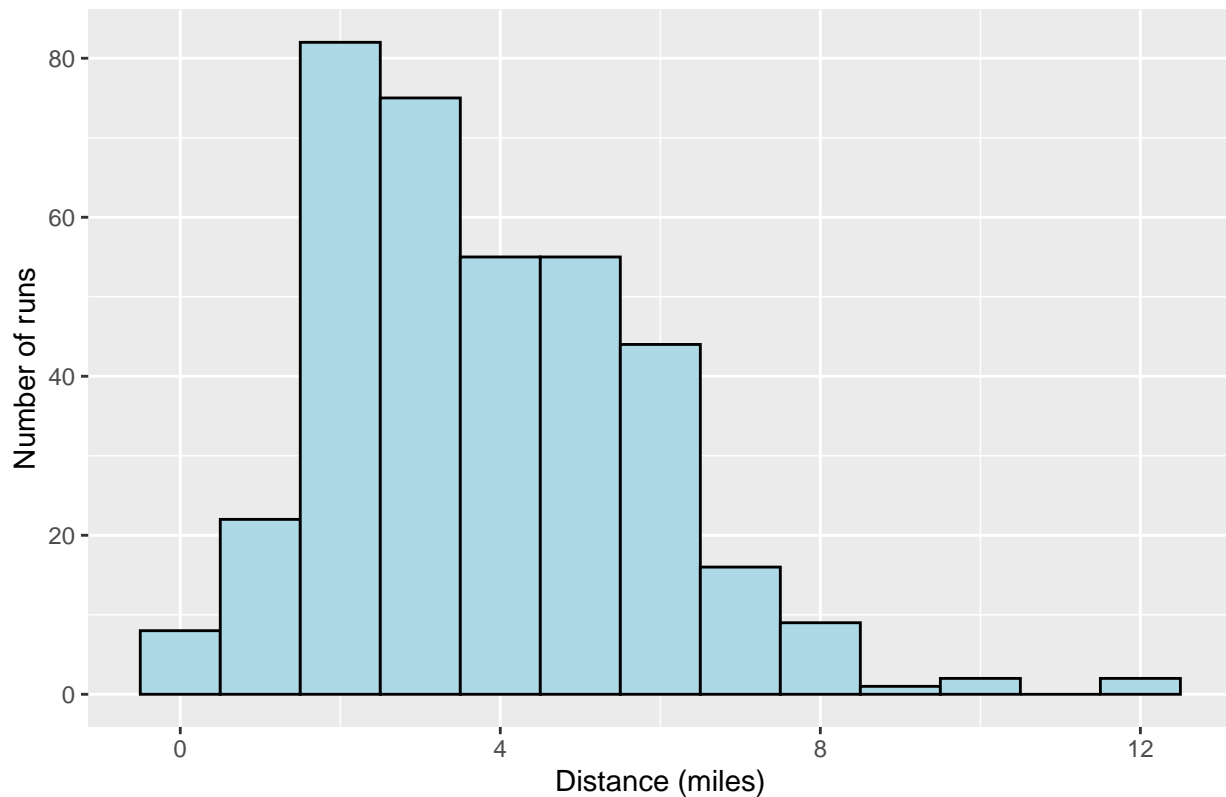
Total Statistics:

```

run_histogram <- df_clean |>
  ggplot(aes(x = distance)) +
  geom_histogram(color = "black", fill = "lightblue", binwidth = 1) +
  labs(x = "Distance (miles)",
    y = "Number of runs",
    title = "Number of runs at each distance")
print(run_histogram)

```

Number of runs at each distance



```
total_summary <- df_clean |>
  summarize(
    total_activities = n(),
    total_miles = sum(distance),
    total_hours = sum(time),
    total_elevation = sum(elevation)) |>
  mutate(total_hours = total_hours / 60)

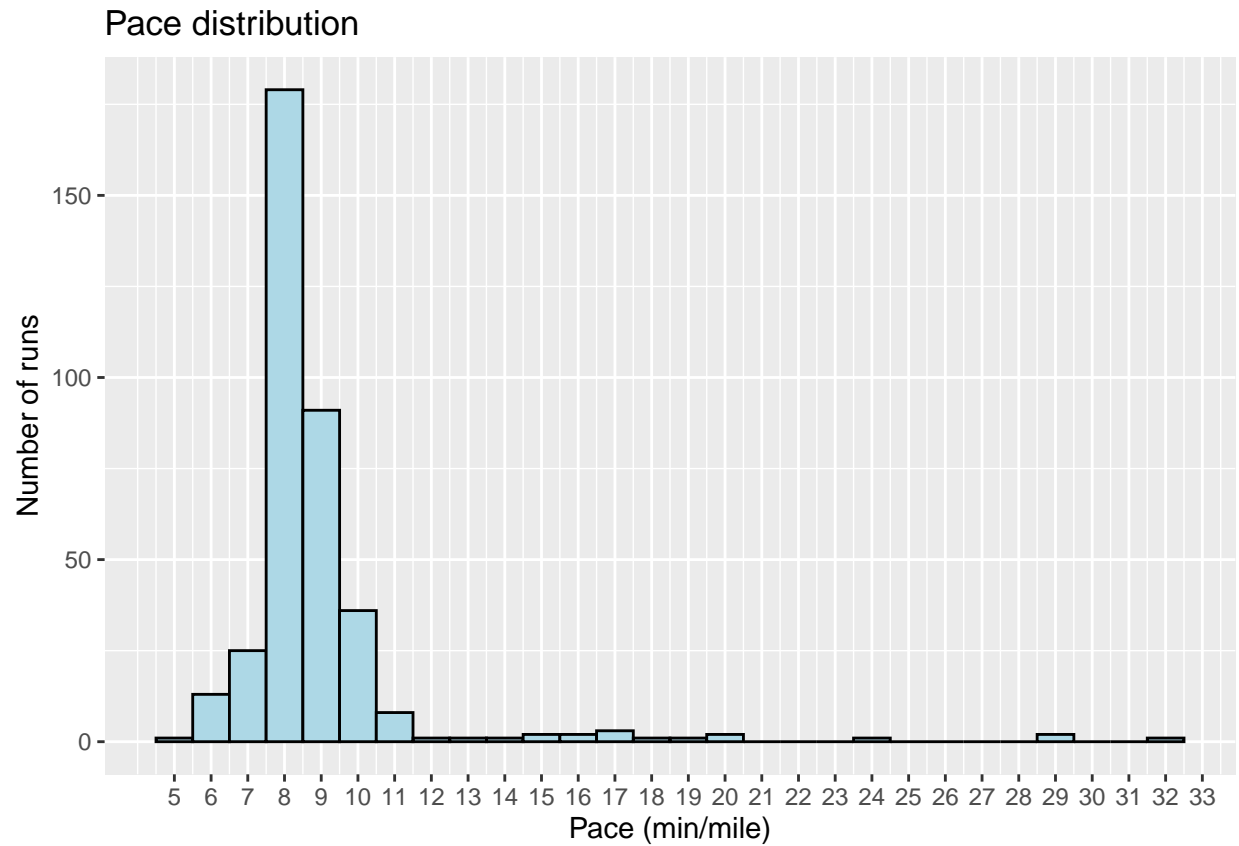
print(total_summary)
```

```
## # A tibble: 1 x 4
##   total_activities total_miles total_hours total_elevation
##   <int>         <dbl>      <dbl>         <dbl>
## 1         371      1413.       204.       77565.
```

Pace histogram:

```
pace_histogram <- df_clean |>
  ggplot(aes(x = pace)) +
  geom_histogram(color = "black", fill = "lightblue", binwidth = 1) +
  labs(x = "Pace (min/mile)",
       y = "Number of runs",
       title = "Pace distribution") +
  scale_x_continuous(breaks = seq(floor(min(df_clean$pace)),
                                  ceiling(max(df_clean$pace)), by = 1))

print(pace_histogram)
```



Adding month and year columns

```
df_dates <- df_clean |>
  mutate(
    year = case_when(
      grepl("2021", date) ~ 2021,
      grepl("2022", date) ~ 2022,
      grepl("2023", date) ~ 2023,
      grepl("2024", date) ~ 2024),
    month = case_when(
      grepl("Jan", date) ~ "Jan",
      grepl("Feb", date) ~ "Feb",
      grepl("Mar", date) ~ "Mar",
      grepl("Apr", date) ~ "Apr",
      grepl("May", date) ~ "May",
      grepl("Jun", date) ~ "Jun",
      grepl("Jul", date) ~ "Jul",
      grepl("Aug", date) ~ "Aug",
      grepl("Sep", date) ~ "Sep",
      grepl("Oct", date) ~ "Oct",
      grepl("Nov", date) ~ "Nov",
      grepl("Dec", date) ~ "Dec",
    )) |>
  select(date, year, month, everything())
```

Yearly Summaries

```
yearly_summaries <- df_dates |>
  group_by(year) |>
  summarize(
    total_activities = n(),
    total_miles = sum(distance),
    total_hours = sum(time),
    total_elevation_feet = sum(elevation)) |>
  mutate(total_hours = total_hours / 60)

print(yearly_summaries)
```

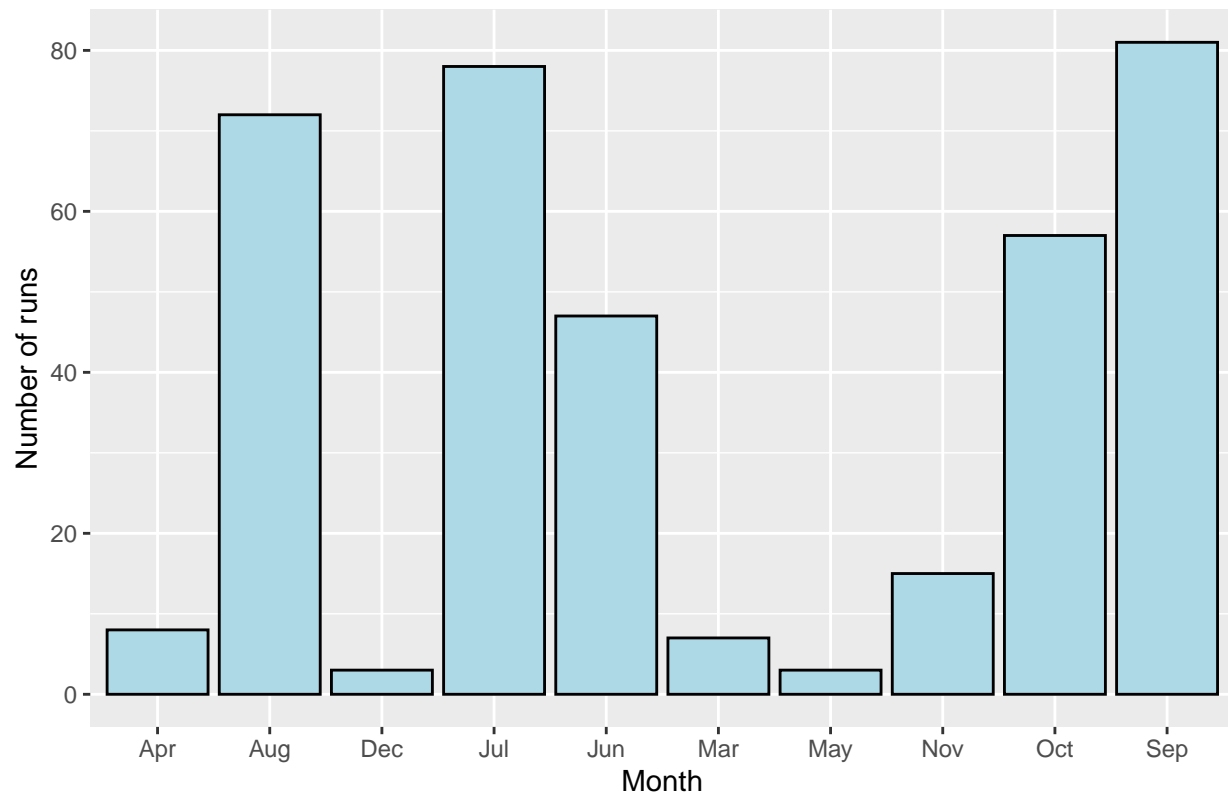
```
## # A tibble: 4 x 5
##   year total_activities total_miles total_hours total_elevation_feet
##   <dbl>         <int>         <dbl>         <dbl>         <dbl>
## 1  2021             139           543.           81.9           38009.
## 2  2022             115           349.           46.6           16829.
## 3  2023             110           505.           73.3           22350.
## 4  2024              7           15.6            2.25            376.
```

Number of runs in each month

```
monthly_runs <- df_dates |>
  group_by(year) |>
  ggplot(aes(x = month)) +
  geom_bar(color = "black", fill = "lightblue") +
  labs(x = "Month",
       y = "Number of runs",
       title = "Number of runs each month across all years")

print(monthly_runs)
```

Number of runs each month across all years



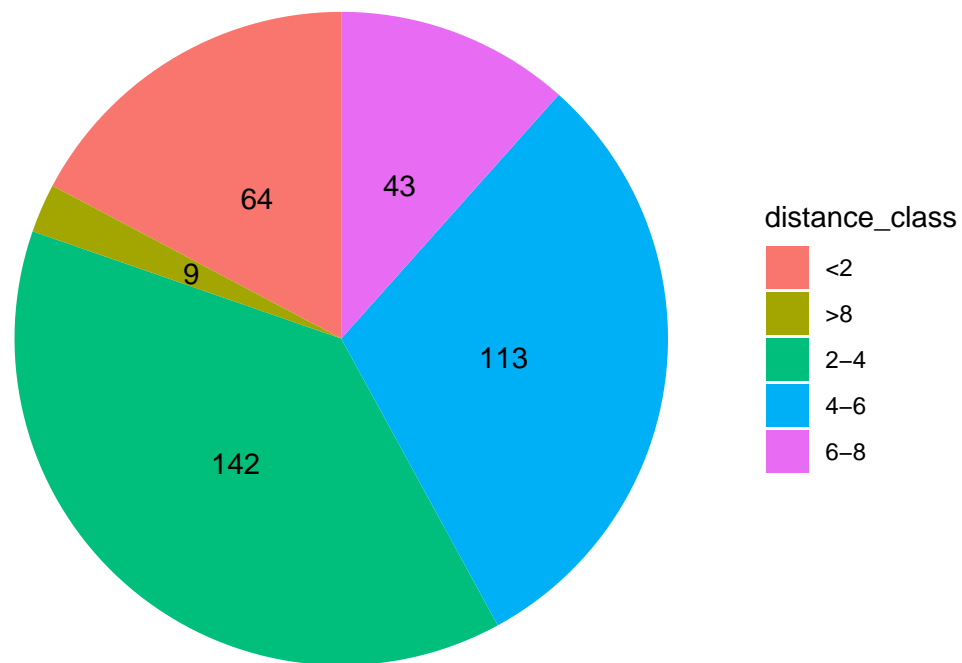
Pie Chart

```
distance_pie <- df_clean |>
  mutate(distance_class = case_when(
    distance >= 0 & distance <= 2 ~ "<2",
    distance > 2 & distance <= 4 ~ "2-4",
    distance > 4 & distance <= 6 ~ "4-6",
    distance > 6 & distance <= 8 ~ "6-8",
    distance > 8 ~ ">8"
  )) |>
  group_by(distance_class) |>
  summarize(count = n()) |>

  ggplot(aes(x = "", y = count, fill = distance_class)) +
    geom_bar(stat = "identity") +
    geom_text(aes(label = count),
              position = position_stack(vjust = 0.5)) +
    coord_polar(theta = "y") +
    theme_void() +
    labs(title = "Number of runs in each distance class")

print(distance_pie)
```

Number of runs in each distance class



Splitting by year

```
df_2021 <- df_clean |>
  filter(grepl(2021, date))

df_2022 <- df_clean |>
  filter(grepl(2022, date))

df_2023 <- df_clean |>
  filter(grepl(2023, date))

df_2024 <- df_clean |>
  filter(grepl(2024, date))

# Decided this was unnecessary, but these are here in case I want to do
# anything in the future
```

All Tables and Graphs

Total Stats

```
print(total_summary)
```

```
## # A tibble: 1 x 4
##   total_activities total_miles total_hours total_elevation
##   <int>         <dbl>         <dbl>         <dbl>
## 1         371       1413.         204.         77565.
```

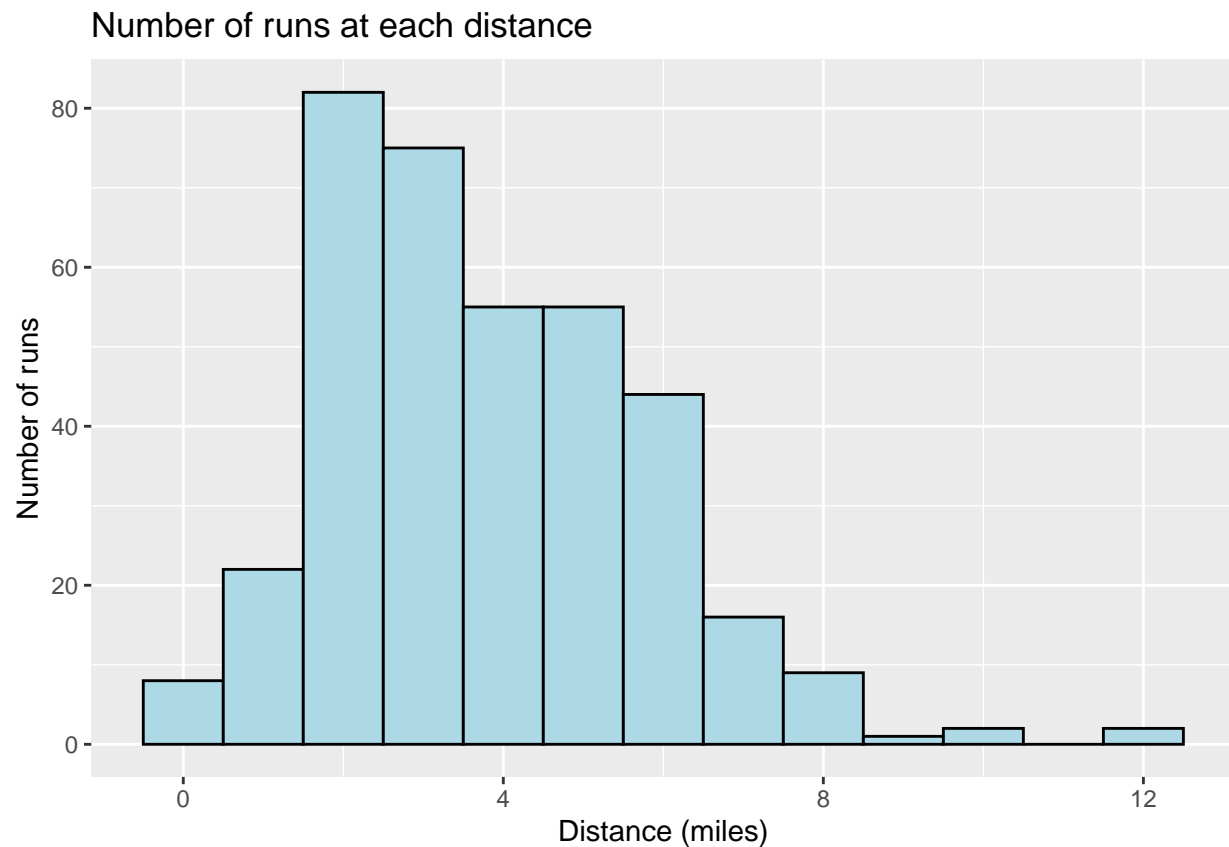
Stats by Year

```
print(yearly_summaries)
```

```
## # A tibble: 4 x 5
##   year total_activities total_miles total_hours total_elevation_feet
##   <dbl>         <int>         <dbl>         <dbl>         <dbl>
## 1  2021             139         543.          81.9         38009.
## 2  2022             115         349.          46.6         16829.
## 3  2023             110         505.          73.3         22350.
## 4  2024              7         15.6           2.25          376.
```

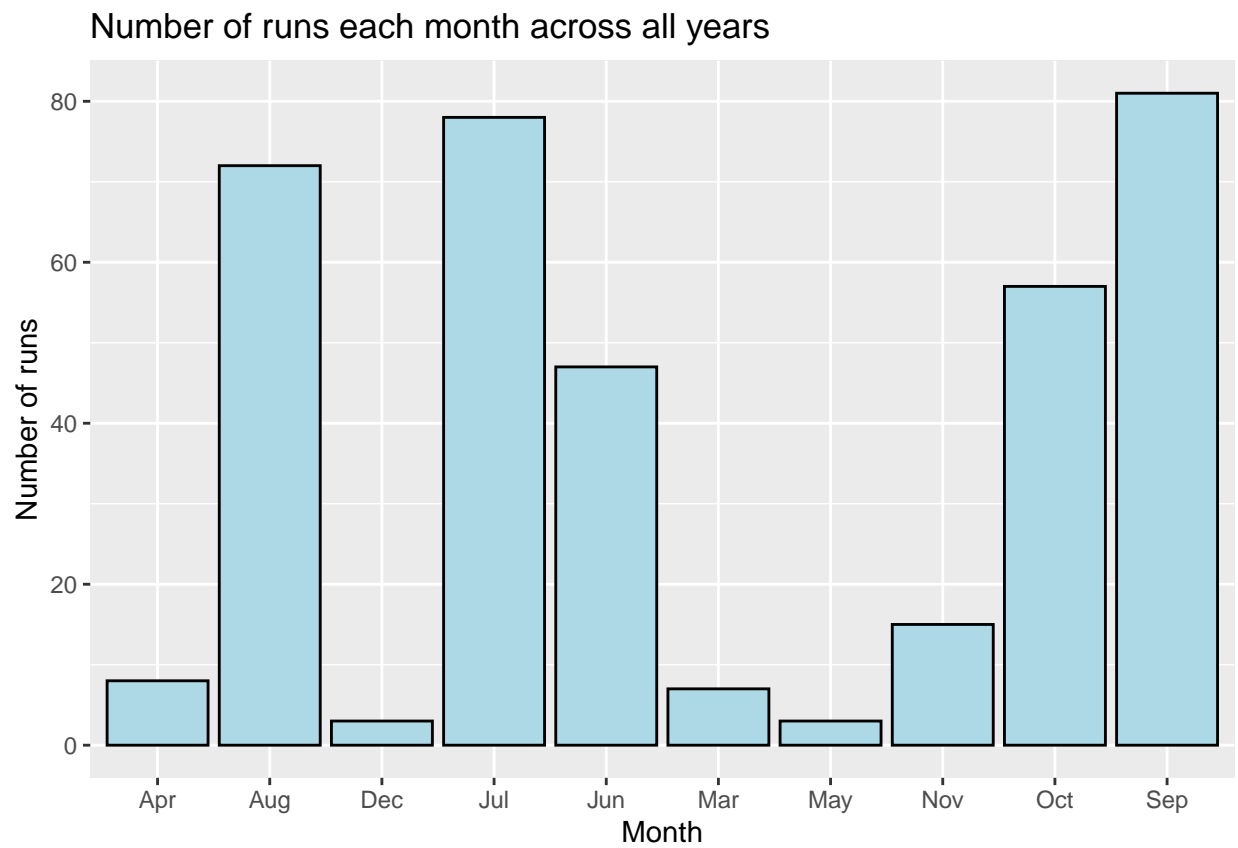
Run Distance Histogram

```
print(run_histogram)
```



Runs in each month

```
print(monthly_runs)
```



Distance Class Pi Chart

```
print(distance_pie)
```

Number of runs in each distance class

