Jan-July23_data_aggration.R

vinay_bijalwan

2023-08-24

```
#Date: 24-08-2023

#this Script

library('data.table')


#Set Default path for my file in System

setwd("C:\\Users\\vinay_bijalwan.PATANJALI\\Desktop\\data_analysis_with_R")


##Read Csav file

dt <- fread("JanJuly2023_04.csv")

dim(dt)
```

```
## [1] 65966    39
```

```
# [1] 65966    38 ---->here is total patient in csv file is 65966


## Data Aggration

# 1. Total Number of Visits per Diagnosis:

diagnosis_summary <- dt[, .(TotalVisits = .N), by = category]
print(diagnosis_summary)
```

```
##                                                                 category
##   1:              MUSCULO-SKELETAL DISORDERS|MUSCULO-SKELETAL DISORDERS
##   2:                           RENAL DISORDERS|GENITO-URINARY DISORDERS
##   3:                                       MUSCULO-SKELETAL DISORDERS
##   4:                                              EYE & ENT DISORDERS
##   5:                                                    MISCELLANEOUS
##  ---
## 889:                                    GIT DISORDERS|SEXUAL DISORDERS
## 890:                                      CANCER|RESPIRATORY DISORDERS
## 891:                                  CANCER|CANCER|ENDOCRINE DISORDERS
## 892: MUSCULO-SKELETAL DISORDERS|MUSCULO-SKELETAL DISORDERS|MISCELLANEOUS
## 893:                                       MISCELLANEOUS|CANCER|CANCER
##      TotalVisits
##   1:         694
##   2:          14
##   3:        9760
##   4:        7007
##   5:        4329
##  ---
## 889:           1
## 890:           1
## 891:           1
## 892:           1
## 893:           1
```

```
#saving the file

fwrite(diagnosis_summary, "diagnosis_summary23.csv")


disease_summary <- dt[, .(TotalVisits = .N), by = disease]

fwrite(disease_summary, "disease.csv")


# 2. Average Number of Visits per Patient:

patient_avg_visits <- dt[, .(AvgVisits = .N / length(unique(visit_date))), b
y = patient_id]
print(patient_avg_visits)
```

```
##        patient_id AvgVisits
##     1:          1         1
##     2:          2         1
##     3:          3         1
##     4:          4         1
##     5:          5         1
##    ---
## 65962:      65962         1
## 65963:      65963         1
## 65964:      65964         1
## 65965:      65965         1
## 65966:      65966         1
```

```
# Time Series Analysis:

# Convert visit_date to a Date object
dt[, visit_date := as.Date(visit_date)]



# Count of Visits Over Time:

# Aggregate and count visits per day
daily_visits <- dt[, .(TotalVisits = .N), by = visit_date]
print(daily_visits)
```

```
##      visit_date TotalVisits
##   1: 0029-03-20         247
##   2: 0018-03-20         471
##   3: 0001-01-20         105
##   4: 0002-01-20         332
##   5: 0003-01-20         315
##  ---
## 207: 0028-07-20         317
## 208: 0029-07-20         470
## 209: 0030-07-20         225
## 210: 0031-07-20         444
## 211: 0030-06-20         351
```

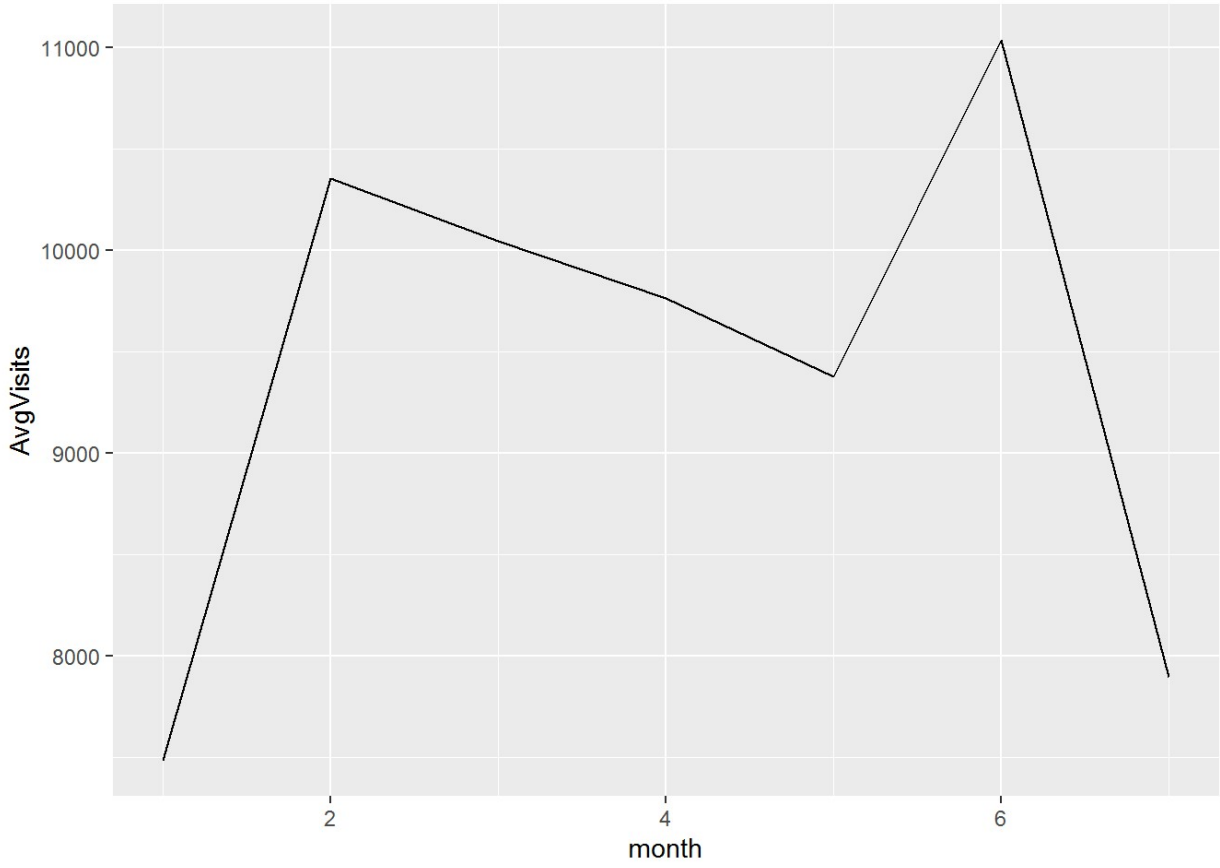
```
#Saving the data in csv file

fwrite(daily_visits, "daily_visits.csv")


# Average Number of Visits Per Month:

# Extract month and year from visit_date
dt[, month := month(visit_date)]
dt[, year := year(visit_date)]


dt[, .N, by = month]
```

```
##    month     N
## 1:     3 10045
## 2:     1  7486
## 3:     2 10355
## 4:     4  9765
## 5:     5  9380
## 6:     6 11037
## 7:     7  7898
```

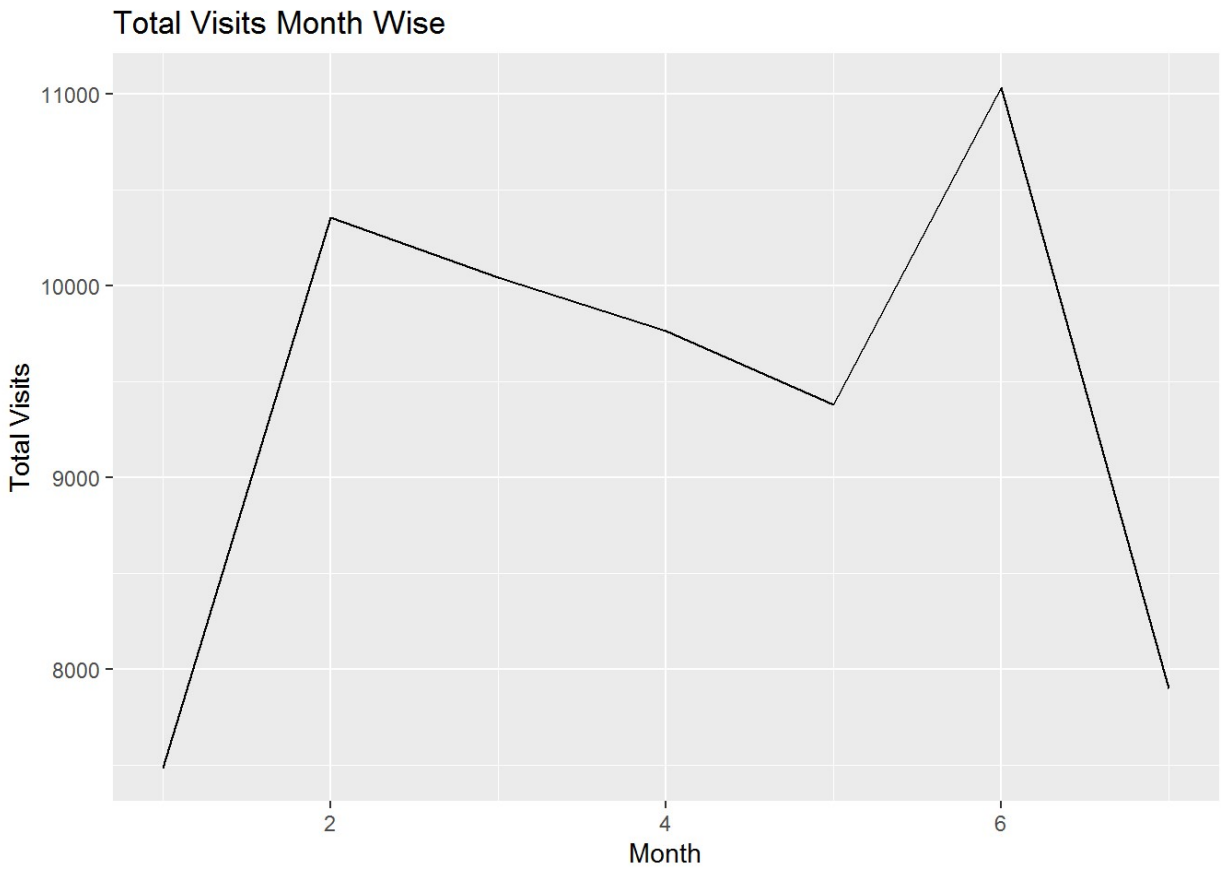
```
# Aggregate and calculate average visits per month
avg_visits_per_month <- dt[, .(AvgVisits = mean(.N)), by = .(month)]
print(avg_visits_per_month)
```

```
##    month AvgVisits
## 1:     3     10045
## 2:     1      7486
## 3:     2     10355
## 4:     4      9765
## 5:     5      9380
## 6:     6     11037
## 7:     7      7898
```

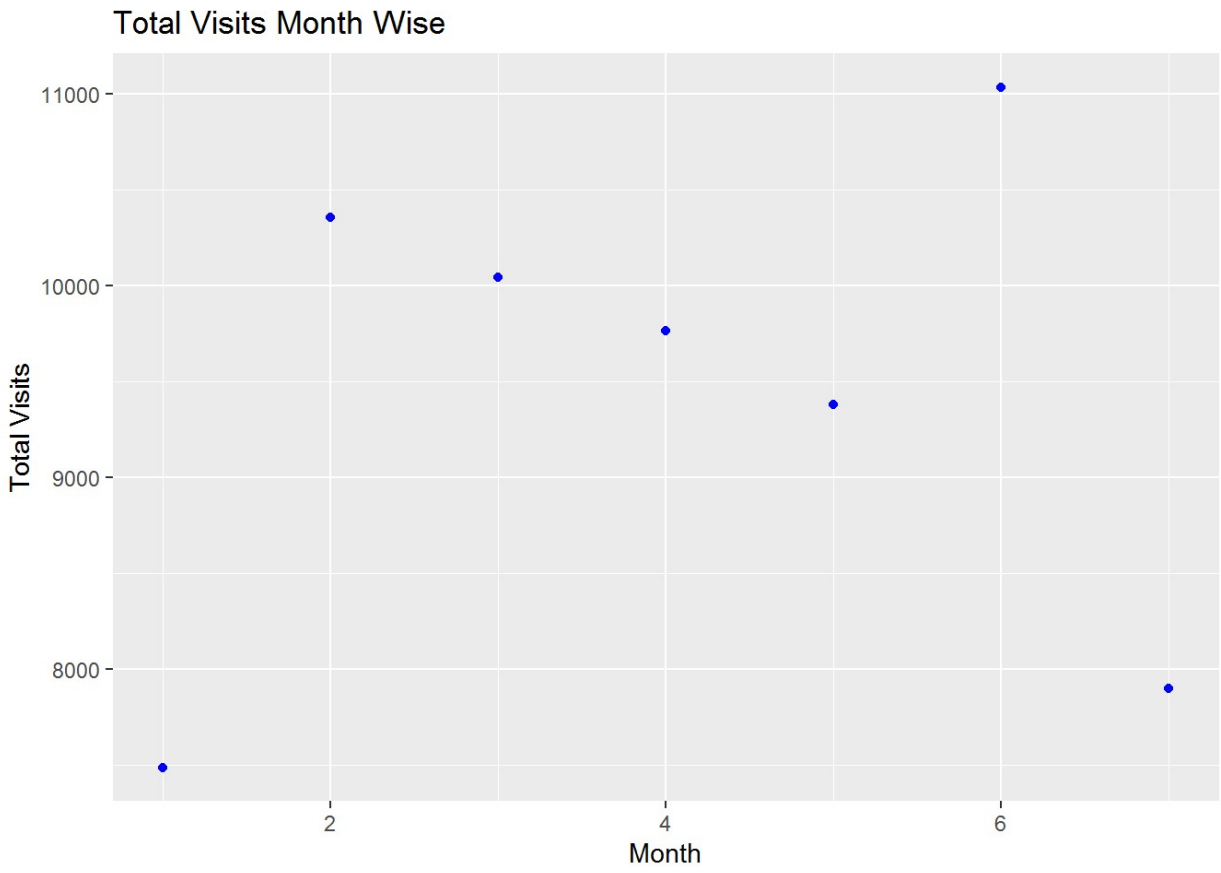
```
#       month AvgVisits
# 1:      3     10045
# 2:      1      7486
# 3:      2     10355
# 4:      4      9765
# 5:      5      9380
# 6:      6     11037
# 7:      7      7898


# Load the ggplot2 library for visualization
library(ggplot2)

##Simple way to use ggplot

# syntax -> ggplot(data = <data>, aes(x = x value, y = "Y value )) + geom_li
ne()

ggplot(avg_visits_per_month, aes(x = month, y = AvgVisits)) +geom_line()
```




```
## Line Chart

ggplot(avg_visits_per_month, aes(x = month, y = AvgVisits)) +
  geom_line() +
  labs(x = "Month", y = "Total Visits", title = "Total Visits Month Wise")
```

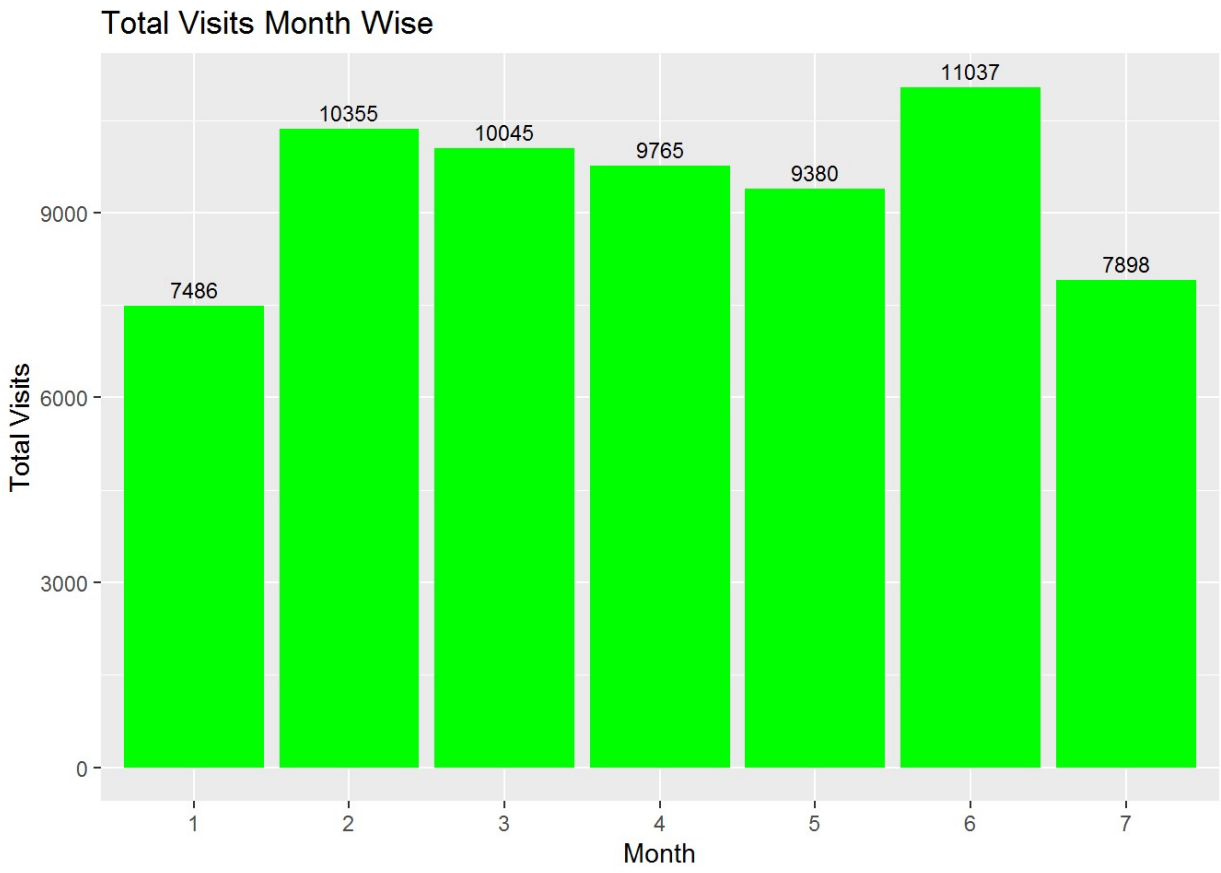



```
## point

ggplot(avg_visits_per_month, aes(x = month, y = AvgVisits)) +
  geom_point(color = "blue") +
  labs(x = "Month", y = "Total Visits", title = "Total Visits Month Wise")
```

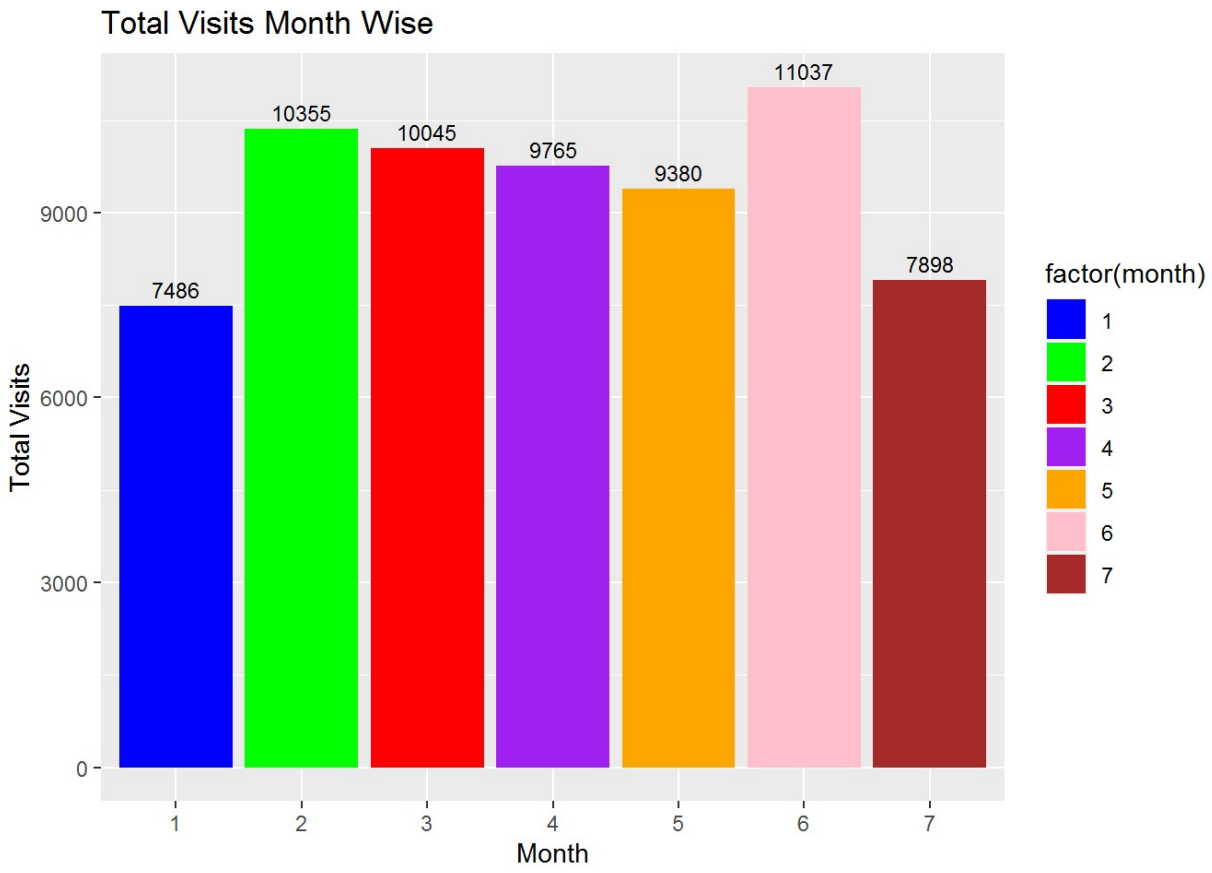



```
## HistBar Chart

ggplot(avg_visits_per_month, aes(x = factor(month), y = AvgVisits)) +
  geom_histogram(stat = "identity", fill = "green") +
  geom_text(aes(label = AvgVisits), vjust = -0.5, color = "black", size =
3) +
  labs(x = "Month", y = "Total Visits", title = "Total Visits Month Wise")
```

```
## Warning in geom_histogram(stat = "identity", fill = "green"): Ignoring un
known
## parameters: `binwidth`, `bins`, and `pad`
```




```
## with different bar color
ggplot(avg_visits_per_month, aes(x = factor(month), y = AvgVisits, fill = fa
ctor(month))) +
  geom_bar(stat = "identity") +
  labs(x = "Month", y = "Total Visits", title = "Total Visits Month Wise") +
  geom_text(aes(label = AvgVisits), vjust = -0.5, color = "black", size = 3)
+
  scale_fill_manual(values = c("blue", "green", "red", "purple", "orange",
"pink", "brown"))  # Match colors to months
```




```
fwrite(avg_visits_per_month, "avg_visits_per_month.csv")
```