DSC630_EdrisSafari_Assignment_3.3

edris safari

9/17/2020

Assignment Description

Using **dodgers.csv** dataset, determine ***what night would be the best to run a marketing promotion to increase attendance***. It is up to you if you decide to recommend a specific date (Jan 1, 2020) or if you want to recommend a day of the week (Tuesdays) or Month and day of the week (July Tuesdays). You will want to use TRAIN. As a reminder, the training set is the data we fit our model on. Use a combination of R and Python to accomplish this assignment. It is important to remember, there will be lots of ways to solve this problem. Explain your thought process and how you used various techniques to come up with your recommendation. From this data, at a minimum, you should be able to demonstrate the following:

Box plots

Scatter plots

Regression Model

Description of approach

The night to increase attendance is the night when attendance is lowest. Given the features in this data set, low attendance could be the result of any feature or combination of features. The goal is to estimate the night when attendance is lowest so more marketing can be done on those nights.

Load the dataset

```
getwd()
```

```
## [1] "C:/Users/safar/Documents/GitHub/Safarie1103/Bellevue University/Courses/DSC63
0/Week3"
```

```
setwd(".\\")
getwd()
```

```
## [1] "C:/Users/safar/Documents/GitHub/Safarie1103/Bellevue University/Courses/DSC63
0/Week3"
```

```
dodgers <- read.csv("Data/dodgers.csv")

str(dodgers)
```

```
## 'data.frame':    81 obs. of  12 variables:
##  $ month      : Factor w/ 7 levels "APR","AUG","JUL",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ day        : int  10 11 12 13 14 15 23 24 25 27 ...
##  $ attend     : int  56000 29729 28328 31601 46549 38359 26376 44014 26345 44807
...
##  $ day_of_week: Factor w/ 7 levels "Friday","Monday",..: 6 7 5 1 3 4 2 6 7 1 ...
##  $ opponent   : Factor w/ 17 levels "Angels","Astros",..: 13 13 13 11 11 11 3 3 3 1
0 ...
##  $ temp       : int  67 58 57 54 57 65 60 63 64 66 ...
##  $ skies      : Factor w/ 2 levels "Clear ","Cloudy": 1 2 2 2 2 1 2 2 2 1 ...
##  $ day_night  : Factor w/ 2 levels "Day","Night": 1 2 2 2 2 1 2 2 2 2 ...
##  $ cap        : Factor w/ 2 levels "NO","YES": 1 1 1 1 1 1 1 1 1 1 ...
##  $ shirt      : Factor w/ 2 levels "NO","YES": 1 1 1 1 1 1 1 1 1 1 ...
##  $ fireworks  : Factor w/ 2 levels "NO","YES": 1 1 1 2 1 1 1 1 1 2 ...
##  $ bobblehead : Factor w/ 2 levels "NO","YES": 1 1 1 1 1 1 1 1 1 1 ...
```

```
head(dodgers)
```

```
##   month day attend day_of_week opponent temp  skies day_night cap shirt
## 1   APR  10  56000     Tuesday  Pirates   67 Clear        Day  NO    NO
## 2   APR  11  29729   Wednesday  Pirates   58 Cloudy     Night  NO    NO
## 3   APR  12  28328    Thursday  Pirates   57 Cloudy     Night  NO    NO
## 4   APR  13  31601      Friday   Padres   54 Cloudy     Night  NO    NO
## 5   APR  14  46549    Saturday   Padres   57 Cloudy     Night  NO    NO
## 6   APR  15  38359      Sunday   Padres   65 Clear        Day  NO    NO
##   fireworks bobblehead
## 1        NO         NO
## 2        NO         NO
## 3        NO         NO
## 4       YES         NO
## 5        NO         NO
## 6        NO         NO
```

```
nrow(dodgers)
```

```
## [1] 81
```

Cleanup data

```
missing_values <- dodgers %>% summarize_each(funs(sum(is.na(.))/n()))
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.
```

```
missing_values <- gather(missing_values, key="feature", value="missing_pct")
num_missing <-  sum(missing_values$missing_pct)
num_missing
```

```
## [1] 0
```

```
print(paste0("Number of missing values = ",as.character(num_missing)))
```

```
## [1] "Number of missing values = 0"
```

```
# There are no missing values
```

```
# Encoding categorical data

# Assign month number to month name
dodgers$month <- factor(dodgers$month,
                          levels = c('JAN','FEB','MAR','APR', 'MAY', 'JUN','JUL','AUG',
'SEP','OCT','NOV','DEC'),
labels = c(1, 2,3,4,5,6,7,8,9,10,11,12))

# Assign Day number to day name
dodgers$day_of_week <- factor(dodgers$day_of_week,
                            levels = c('Monday', 'Tuesday','Wednesday','Thursday','Frid
ay','Saturday','Sunday'),
                            labels = c(1, 2,3,4,5,6,7))

# Assign 0 and 1 to sky condition of clear and cloudy

# Note: The variable value 'CLean' in the dataset was actually typed 'Clean '(With a s
pace)
# If not specified as such , the factor function returns 'NS' for the value.
dodgers$skies <- factor(dodgers$skies,
                        levels = c('Clear ','Cloudy'),
                        labels = c(0,1))



# Assign 0 and 1 to sky condition of night and day
dodgers$day_night <- factor(dodgers$day_night,
                        levels = c('Night','Day'),
                        labels = c(0,1))


# Assign 0 and 1 to NO and YES values
dodgers$cap <- factor(dodgers$cap,
                        levels = c('NO','YES'),
                        labels = c(0,1))



dodgers$shirt <- factor(dodgers$shirt,
                        levels = c('NO','YES'),
                        labels = c(0,1))



dodgers$fireworks <- factor(dodgers$fireworks,
                        levels = c('NO','YES'),
                        labels = c(0,1))



dodgers$bobblehead <- factor(dodgers$bobblehead,
                        levels = c('NO','YES'),
                        labels = c(0,1))
```

```
# encode oponents
oponent <- dodgers$opponent
dodgers$opponent <- as.numeric(factor(oponent))


#add a new column for total number of items purchase}
dodgers$tot_pchd <- as.numeric(as.character(dodgers$cap))+ as.numeric(as.character(dod
gers$shirt)) + as.numeric(as.character(dodgers$fireworks)) + as.numeric(as.character(d
odgers$bobblehead))


head(dodgers)
```

```
##   month day attend day_of_week opponent temp skies day_night cap shirt
## 1     4  10  56000           2       13   67     0         1   0     0
## 2     4  11  29729           3       13   58     1         0   0     0
## 3     4  12  28328           4       13   57     1         0   0     0
## 4     4  13  31601           5       11   54     1         0   0     0
## 5     4  14  46549           6       11   57     1         0   0     0
## 6     4  15  38359           7       11   65     0         1   0     0
##   fireworks bobblehead tot_pchd
## 1         0          0        0
## 2         0          0        0
## 3         0          0        0
## 4         1          0        1
## 5         0          0        0
## 6         0          0        0
```

```
#write.csv(dodgers,file="Data/Clean_Dodgers.csv",row.names = FALSE)
```

```
# Read back clean dataset
#dodgers <- read.csv("Data/Clean_Dodgers.csv")
#head(dodgers)
```

EDA

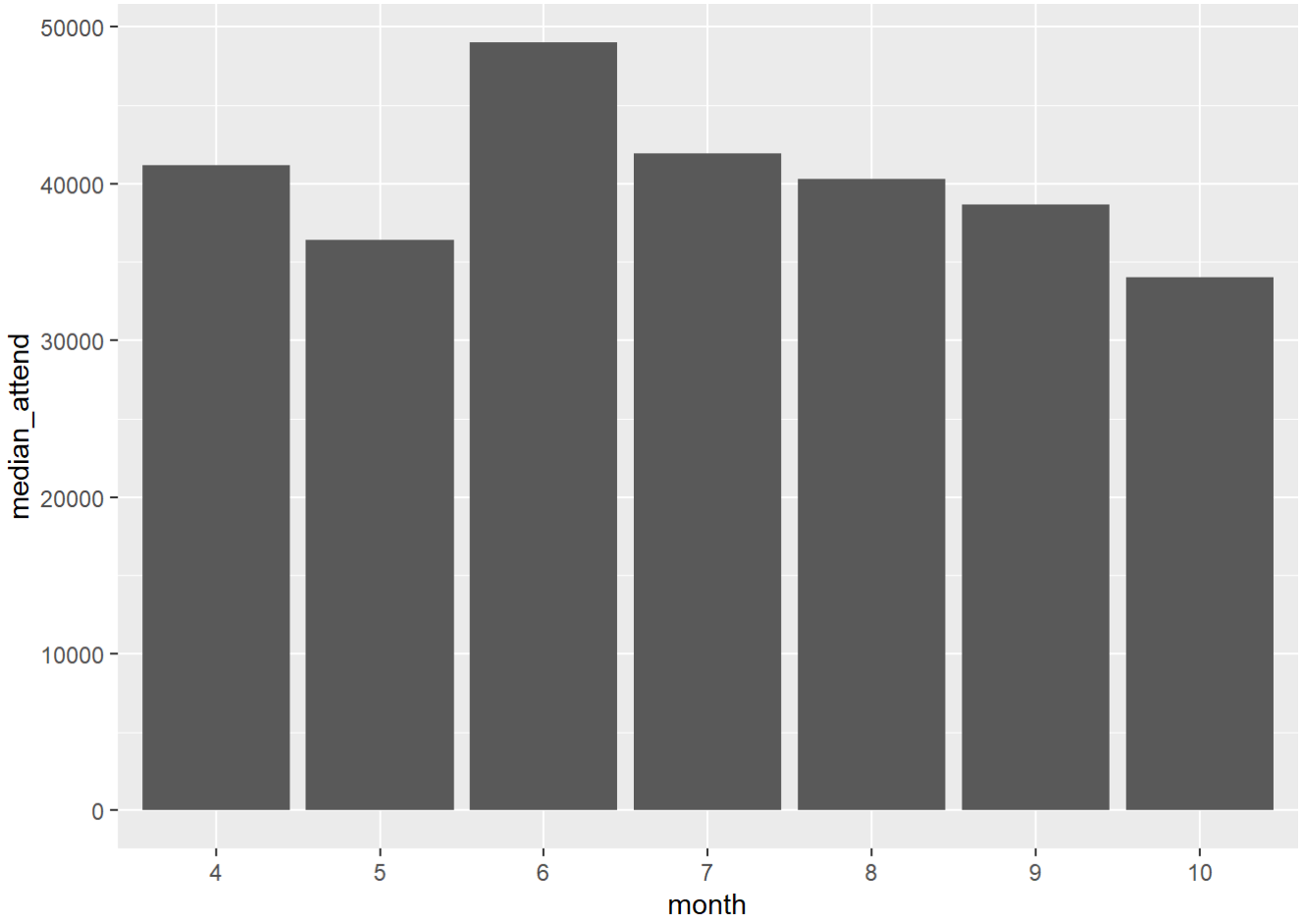
```
summary(dodgers)
```

```
##      month         day            attend      day_of_week    opponent
##  5      :18   Min.   : 1.00   Min.   :24312   1:12        Min.   : 1.000
##  8      :15   1st Qu.: 8.00   1st Qu.:34493   2:13        1st Qu.: 6.000
##  4      :12   Median :15.00   Median :40284   3:12        Median :10.000
##  7      :12   Mean   :16.14   Mean   :41040   4: 5        Mean   : 9.704
##  9      :12   3rd Qu.:25.00   3rd Qu.:46588   5:13        3rd Qu.:15.000
##  6      : 9   Max.   :31.00   Max.   :56000   6:13        Max.   :17.000
##  (Other): 3                                   7:13
##       temp       skies  day_night cap    shirt  fireworks bobblehead
##  Min.   :54.00   0:62   0:66      0:79   0:78   0:67      0:70
##  1st Qu.:67.00   1:19   1:15      1: 2   1: 3   1:14      1:11
##  Median :73.00
##  Mean   :73.15
##  3rd Qu.:79.00
##  Max.   :95.00
##
##     tot_pchd
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3704
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
```

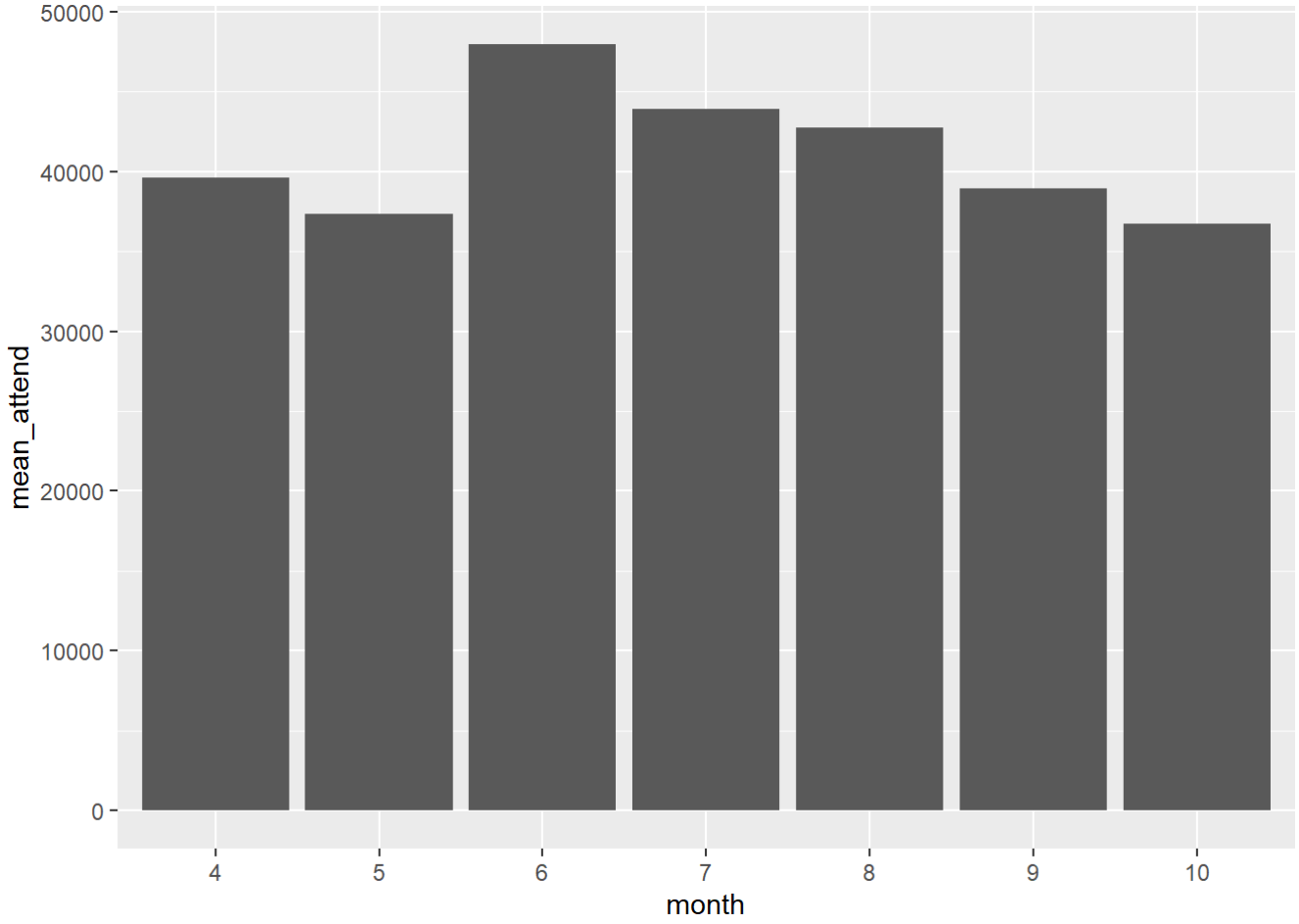
Summary statistics show the maximum attendance is 5600 and minimum is 24312. The max number of products purchased is 1 which doesn’t show too much interest in purchasing any product on any given day or who the oponent is.

Graphs

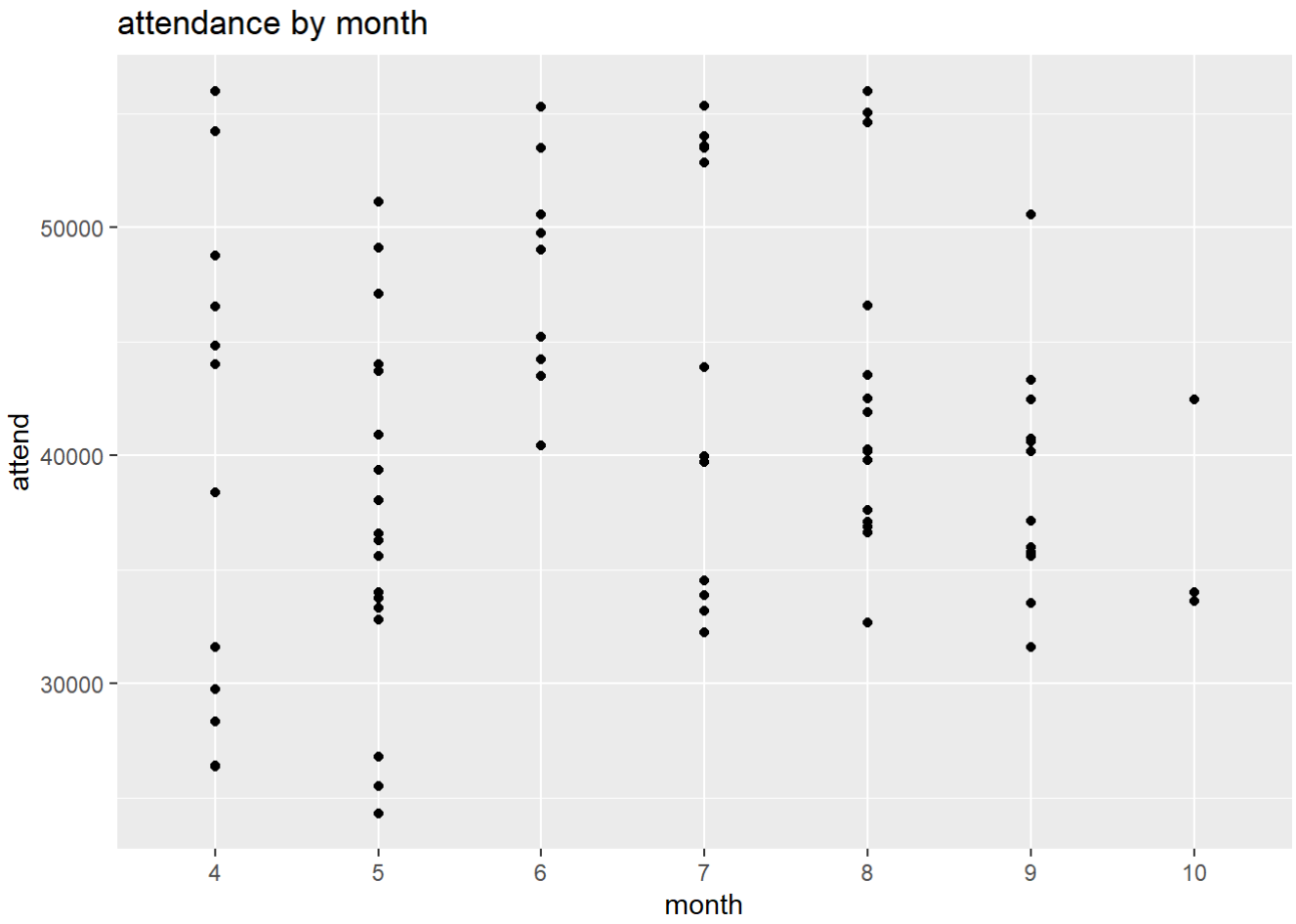
```
plotdata <- dodgers %>%
  group_by(month) %>%
  summarize(median_attend = median(attend))

# plot mean salaries
ggplot(plotdata,
       aes(x = month,
           y = median_attend)) +
  geom_bar(stat = "identity")
```



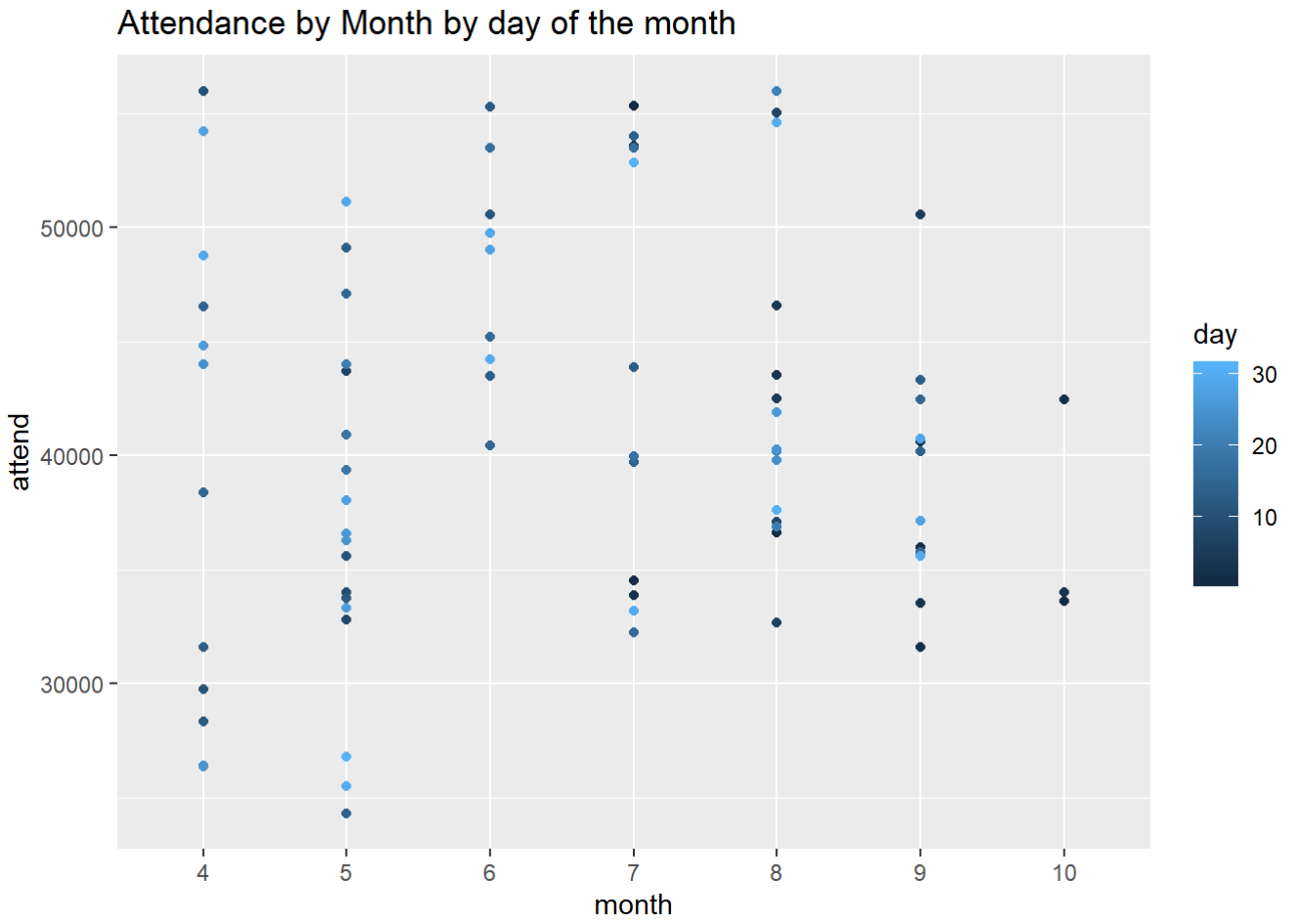

```
plotdata <- dodgers %>%
  group_by(month) %>%
  summarize(mean_attend = mean(attend))

# plot mean salaries
ggplot(plotdata,
       aes(x = month,
           y = mean_attend)) +
  geom_bar(stat = "identity")
```



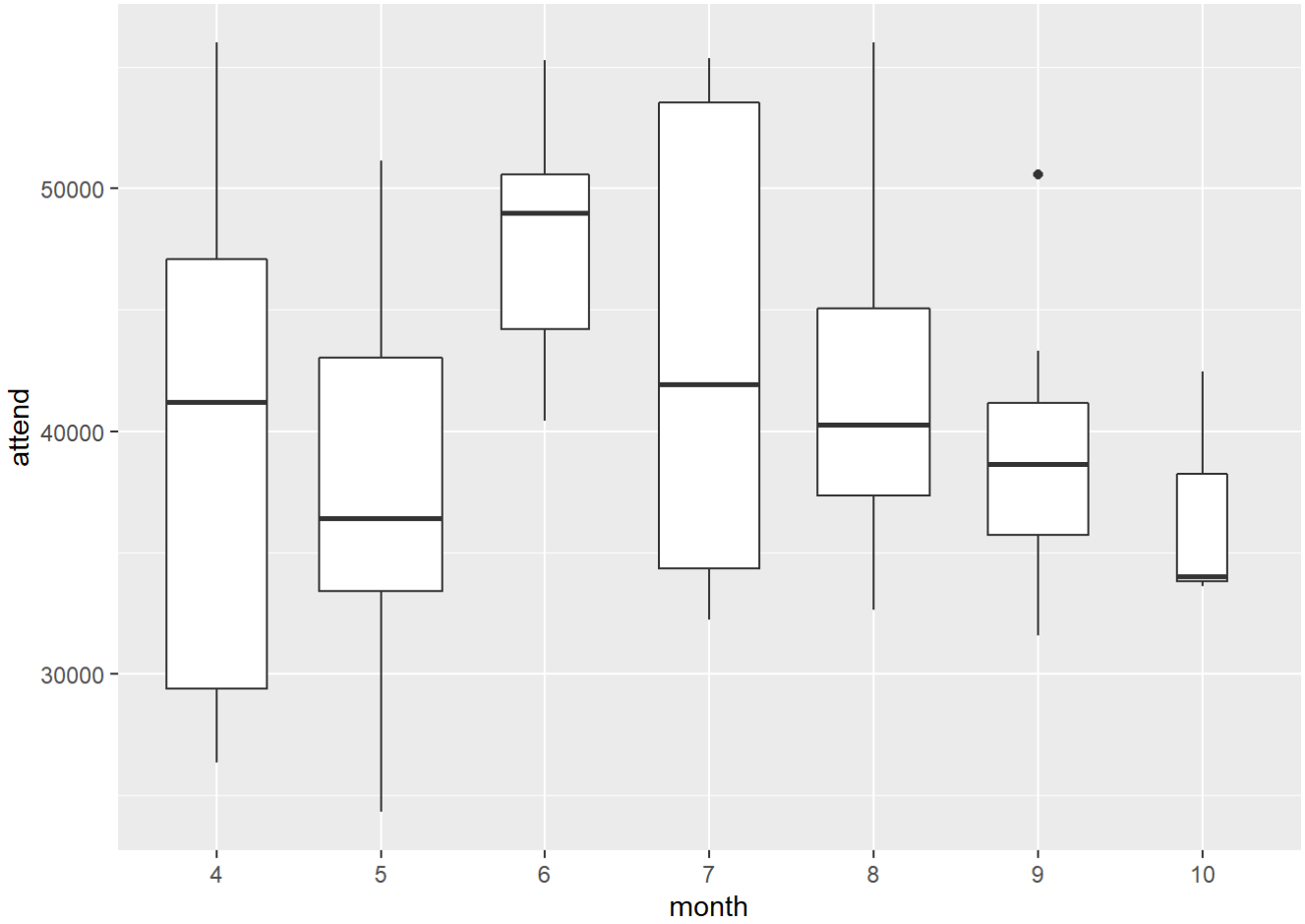

```
# Scatter plot of b
ggplot(dodgers,aes(x=month,y=attend)) +
  geom_point() +
  labs(title = "attendance by month")
```



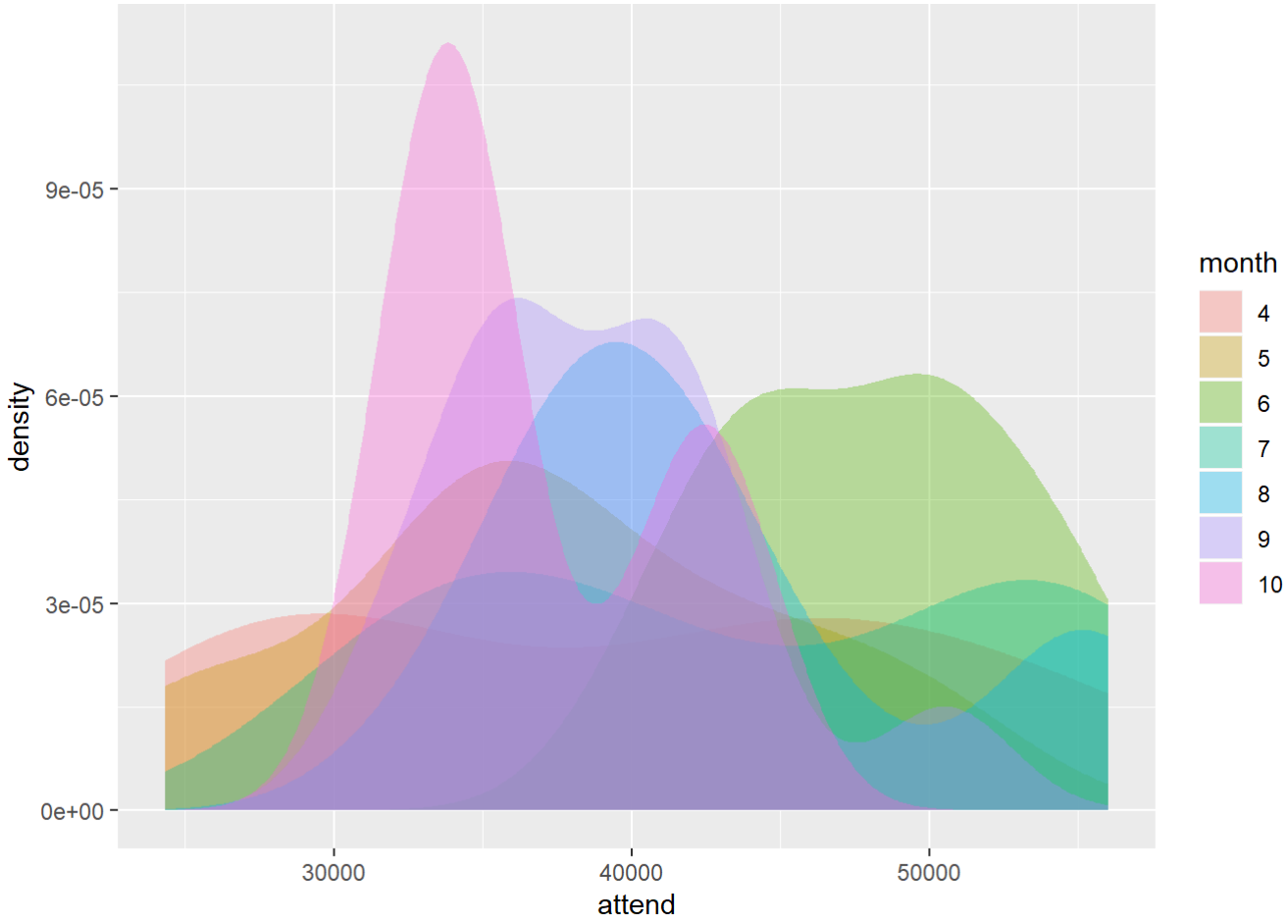

```
ggplot(dodgers,aes(x=month,y=attend,color=day)) +
  geom_point() +
  labs(title = "Attendance by Month by day of the month")
```




```
#


ggplot(dodgers,aes(x=month,y=attend)) +
  geom_boxplot(varwidth = TRUE)
```




```
ggplot(dodgers,aes(x=attend,fill=month)) +
  geom_density(col = NA,alpha = 0.35)
```



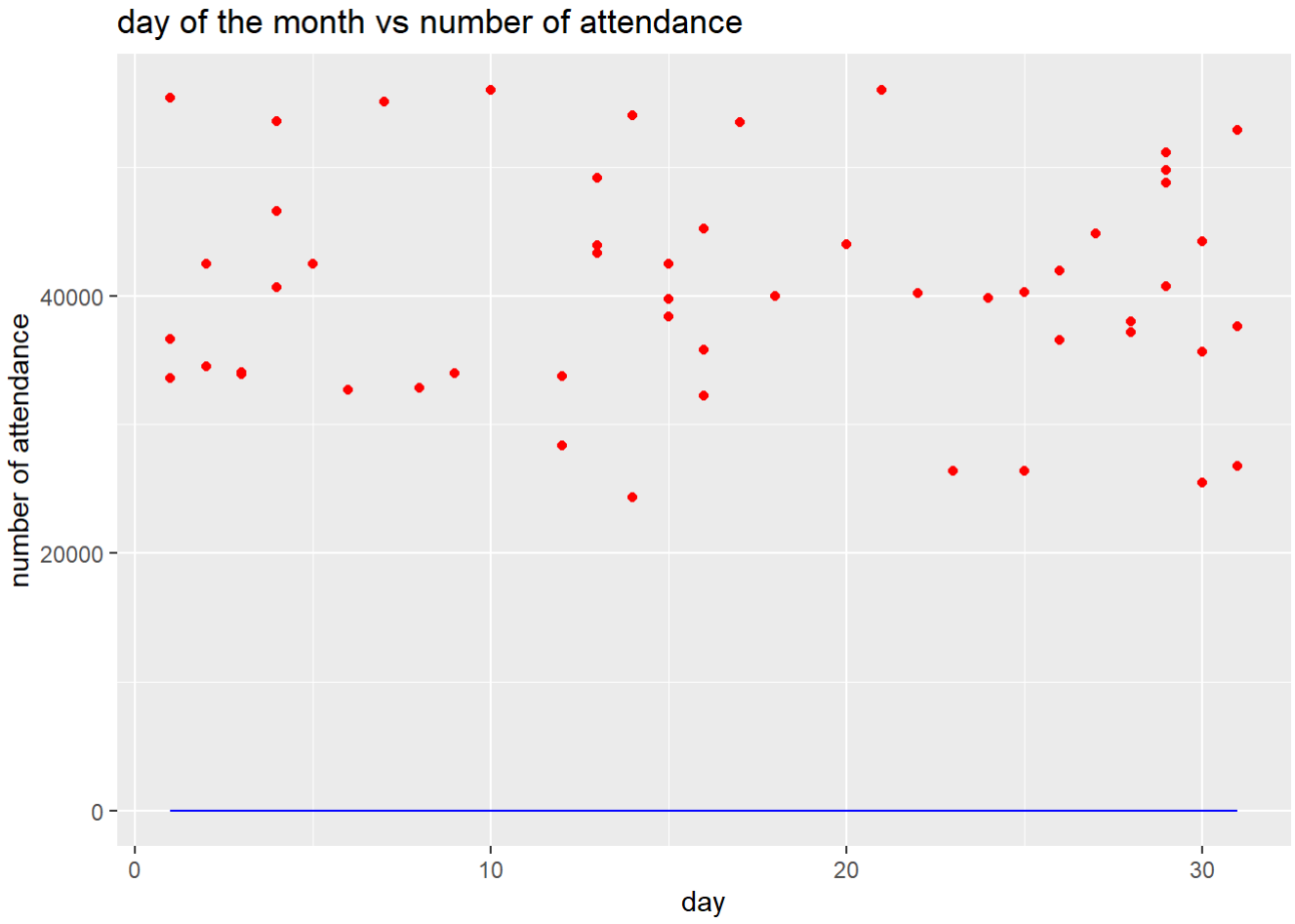
Modeling

```
# Splitting the dataset into the Training set and Test set
# install.packages('caTools')
library(caTools)
```

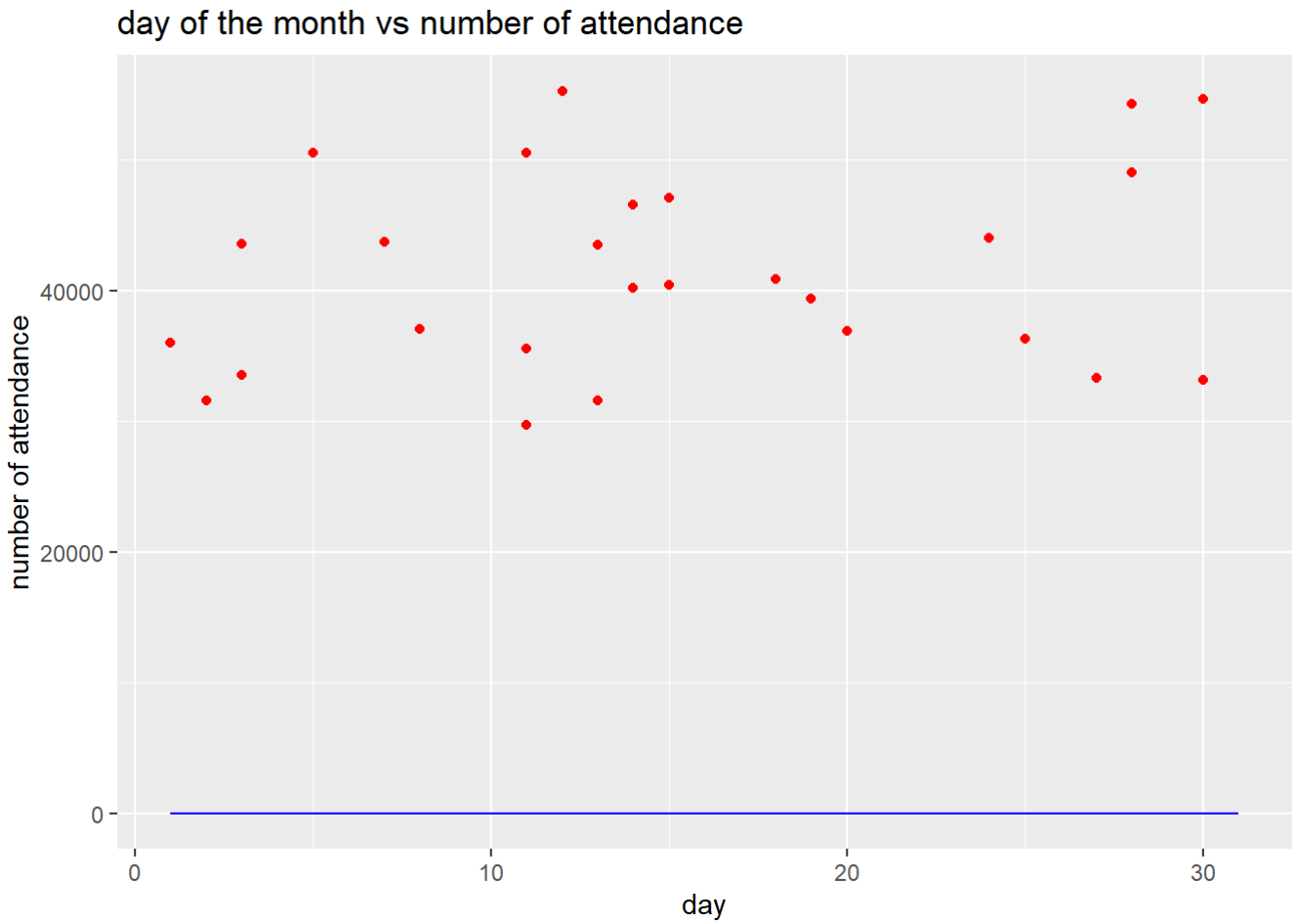
```
## Warning: package 'caTools' was built under R version 3.6.3
```

```
set.seed(123)
split = sample.split(dodgers$attend, SplitRatio = 2/3)
training_set = subset(dodgers, split == TRUE)
test_set = subset(dodgers, split == FALSE)

# Feature Scaling
# training_set = scale(training_set)
# test_set = scale(test_set)

# Fitting Simple Linear Regression to the Training set
regressor = lm(formula = day ~ attend,
               data = training_set)

# Predicting the Test set results
y_pred = predict(regressor, newdata = test_set)
```

```
# Visualising the Training set results
library(ggplot2)
ggplot() +
  geom_point(aes(x = training_set$day, y = training_set$attend),
             colour = 'red') +
  geom_line(aes(x = training_set$day, y = predict(regressor, newdata = training_set)),
            colour = 'blue') +
  ggtitle('day of the month vs number of attendance') +
  xlab('day') +
  ylab('number of attendance')
```




```
# Visualising the Test set results
library(ggplot2)
ggplot() +
  geom_point(aes(x = test_set$day, y = test_set$attend),
             colour = 'red') +
  geom_line(aes(x = training_set$day, y = predict(regressor, newdata = training_set)),
            colour = 'blue') +
 ggtitle('day of the month vs number of attendance') +
  xlab('day') +
  ylab('number of attendance')
```


day of the month vs number of attendance
number of attendance
40000
20000
0
0
10
20
30
day