Sports Data Analysis

Objective: Sports Data Analysis of cricket data as part of Lab Migration Project.

Methods:

(i) Import and load the dataset.
(ii) Perform cleaning of data.
(iii) Perform data manipulation and find inferences.
(iv) Perform statistical analysis of the data.
(v) Visualize the data graphically.
(vi) Apply correlation and regression analysis.
(vii) Conclusion

```
#Import the required libraries
library(MASS)
library("readxl")
library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.2 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v readr   2.1.2
```

```
## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
#Import and load the dataset
cricket <- read_excel("cricket_data.xlsx")
head(cricket)
```

```
## # A tibble: 6 x 10
##   score runs_sc~1 balls~2 strik~3 fours sixes oppos~4 ground date
##   <chr>     <dbl>   <dbl>   <dbl> <dbl> <dbl> <chr>   <chr>  <dttm>
## 1 0             0       1     0       0     0 v Bang~ Chatt~ 2004-12-23
00:00:00
## 2 12           12      11   109.      2     0 v Bang~ Dhaka  2004-12-26
00:00:00
## 3 7*            7       2   350       0     1 v Bang~ Dhaka  2004-12-27
00:00:00
## 4 3             3       7    42.8     0     0 v Paki~ Kochi  2005-04-02
00:00:00
## 5 148         148     123   120.     15     4 v Paki~ Visak~ 2005-04-05
00:00:00
## 6 28           28      24   117.      5     0 v Paki~ Jamsh~ 2005-04-09
00:00:00
## # ... with 1 more variable: odi_number <chr>, and abbreviated variable
names
## #   1: runs_scored, 2: balls_faced, 3: strike_rate, 4: opposition
## # i Use `colnames()` to see all variable names
```

Data Cleaning

```
#check for missing values
any(is.na(cricket))
```

```
## [1] FALSE
```

```
unique(cricket$opposition)
```

```
##  [1] "v Bangladesh"   "v Pakistan"     "v Sri Lanka"    "v West Indies"
##  [5] "v New Zealand"  "v Zimbabwe"     "v South Africa" "v England"
##  [9] "v Australia"    "v Bermuda"      "v Africa XI"    "v Scotland"
```

```
## [13] "v Hong Kong"    "v Ireland"      "v Netherlands"  "v U.A.E."
## [17] "v Afghanistan"
```

```
#cleaning unwanted substring "v" from the column
cricket <- cricket%>%
  mutate(opposition=gsub("v ","",opposition))
unique(cricket$opposition)
```

```
##  [1] "Bangladesh"   "Pakistan"     "Sri Lanka"    "West Indies"  "New
Zealand"
##  [6] "Zimbabwe"     "South Africa" "England"      "Australia"    "Bermuda"
## [11] "Africa XI"    "Scotland"     "Hong Kong"    "Ireland"
"Netherlands"
## [16] "U.A.E."       "Afghanistan"
```

```
#drop the odi_number feature because it adds no value to the analysis
cricket <- subset (cricket, select = -odi_number)
```

Data Manipulation

```
#display top five records with runs scored and whose opposition is Bangladesh
cricket %>%
  filter(opposition=='Bangladesh') %>%
  arrange(desc(runs_scored)) %>%
  head(5)
```

```
## # A tibble: 5 x 9
##   score runs_sc~1 balls~2 strik~3 fours sixes oppos~4 ground date
##   <chr>     <dbl>   <dbl>   <dbl> <dbl> <dbl> <chr>   <chr>  <dttm>
## 1 101*        101     107    94.4     9     0 Bangla~ Dhaka  2010-01-07
00:00:00
## 2 91*          91     106    85.8     7     0 Bangla~ Dhaka  2007-05-10
00:00:00
## 3 69           69      77    89.6     6     1 Bangla~ Dhaka  2015-06-24
00:00:00
## 4 47           47      75    62.7     3     0 Bangla~ Dhaka  2015-06-21
00:00:00
## 5 38*          38      45    84.4     3     1 Bangla~ Dambu~ 2010-06-16
00:00:00
## # ... with abbreviated variable names 1: runs_scored, 2: balls_faced,
## #   3: strike_rate, 4: opposition
```

```
#display ten records with strike rate>100
cricket %>%
  filter(strike_rate>100) %>%
  head(10)
```

```
## # A tibble: 10 x 9
##    score runs_s~1 balls~2 strik~3 fours sixes oppos~4 ground date
##    <chr>    <dbl>   <dbl>   <dbl> <dbl> <dbl> <chr>   <chr>  <dttm>
##  1 12          12      11    109.     2     0 Bangla~ Dhaka  2004-12-26
00:00:00
```

```
## 2 7*            7       2    350       0      1 Bangla~ Dhaka  2004-12-27
00:00:00
## 3 148         148     123    120.     15      4 Pakist~ Visak~ 2005-04-05
00:00:00
## 4 28           28      24    117.      5      0 Pakist~ Jamsh~ 2005-04-09
00:00:00
## 5 15*          15      11    136.      1      1 West I~ Dambu~ 2005-07-31
00:00:00
## 6 28*          28      13    215.      1      2 West I~ Colom~ 2005-08-07
00:00:00
## 7 56           56      46    122.      2      4 Zimbab~ Harare 2005-08-29
00:00:00
## 8 37*          37      27    137.      3      2 New Ze~ Harare 2005-09-02
00:00:00
## 9 67*          67      63    106.      1      3 Zimbab~ Harare 2005-09-04
00:00:00
## 10 38          38      28    136.      3      2 Sri La~ Nagpur 2005-10-25
00:00:00
## # ... with abbreviated variable names 1: runs_scored, 2: balls_faced,
## #   3: strike_rate, 4: opposition
```

```
#display random 5 rows of score, runs scored, balls faced columns
cricket %>%
  dplyr::select(score,runs_scored,balls_faced) %>%
  slice_sample(n=5)
```

```
## # A tibble: 5 x 3
##   score runs_scored balls_faced
##   <chr>       <dbl>       <dbl>
## 1 88*            88          95
## 2 49             49          74
## 3 18             18          13
## 4 29             29          42
## 5 DNB             0           0
```

```
#find the first match date
minc <- cricket %>%
  arrange(date) %>%
  head(1)
minc['date']
```

```
## # A tibble: 1 x 1
##   date
##   <dttm>
## 1 2004-12-23 00:00:00
```

```
#find the last match date
maxc <- cricket %>%
  arrange(desc(date)) %>%
  head(1)
maxc['date']
```

```
## # A tibble: 1 x 1
##   date
##   <dttm>
## 1 2019-07-09 00:00:00
```

```
#find the number of balls faced in career
sum(cricket$balls_faced)
```

```
## [1] 12303
```

```
#find highest runs scored
max(cricket$runs_scored)
```

```
## [1] 183
```

Statistical Analysis

```
#find the number of matches in each ground
ground_freq <- table(cricket$ground)
ground_freq
```

```
##
##          Abu Dhabi            Adelaide           Ahmedabad
Auckland
##                  2                   6                   5
3
##         Basseterre             Belfast            Bengaluru
Birmingham
##                  1                   2                   8
8
##           Brisbane             Bristol            Bulawayo
Canberra
##                  6                   1                   1
2
##          Cape Town             Cardiff           Centurion
Chandigarh
##                  3                   4                   7
1
##         Chattogram             Chennai  Chester0le0Street
Christchurch
##                  1                   6                   1
1
##      Colombo (RPS)       Colombo (SSC)             Cuttack
Dambulla
##                 16                   1                   3
16
##              Delhi               Dhaka          Dharamsala        Dubai
(DSC)
##                  9                  19                   4
6
##             Durban           Faridabad             Glasgow         Gros
Islet
```

```
##                    4                  1                  1
2
##             Guwahati            Gwalior         Hambantota
Hamilton
##                    3                  2                  2
4
##               Harare             Hobart Hyderabad (Deccan)
Indore
##                    7                  2                  5
3
##               Jaipur         Jamshedpur       Johannesburg
Kanpur
##                    5                  2                  4
6
##              Karachi           Kingston              Kochi
Kolkata
##                    7                  6                  6
6
##         Kuala Lumpur             Lahore              Leeds
Lord's
##                    4                  1                  4
3
##           Manchester             Margao          Melbourne
Mohali
##                    4                  2                  7
10
##      Mount Maunganui             Multan             Mumbai        Mumbai
(BS)
##                    1                  1                  6
1
##               Nagpur             Napier        North Sound
Nottingham
##                    9                  3                  2
2
##            Pallekele              Perth           Peshawar     Port
Elizabeth
##                    3                  5                  1
3
##        Port of Spain               Pune             Rajkot
Ranchi
##                    8                  5                  6
4
##           Rawalpindi        Southampton             Sydney           The
Oval
##                    1                  4                  7
7
## Thiruvananthapuram           Vadodara      Visakhapatnam
Wellington
##                    1                  4                  7
3
```

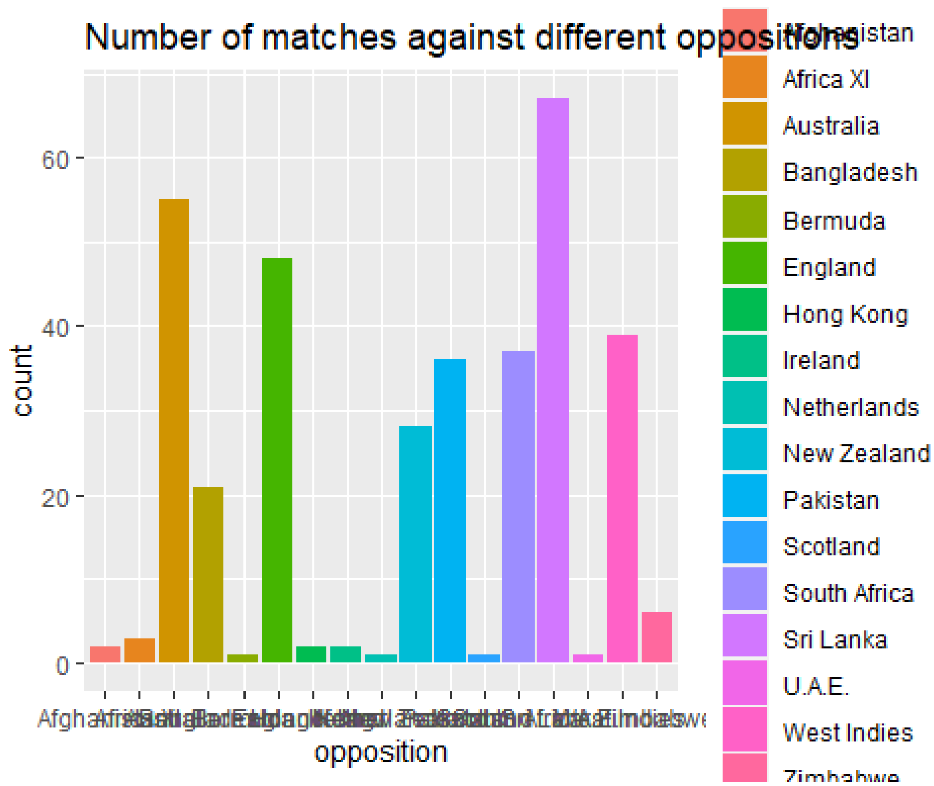
```
#find the relative frequency distribution of the ground and display them with
the precision of two decimal places
n=nrow(cricket)
ground_relfreq=ground_freq/n
format(round(ground_relfreq,2),nsmall=2)

##
##          Abu Dhabi            Adelaide           Ahmedabad
Auckland
##             "0.01"              "0.02"              "0.01"
"0.01"
##         Basseterre             Belfast           Bengaluru
Birmingham
##             "0.00"              "0.01"              "0.02"
"0.02"
##           Brisbane             Bristol            Bulawayo
Canberra
##             "0.02"              "0.00"              "0.00"
"0.01"
##          Cape Town             Cardiff           Centurion
Chandigarh
##             "0.01"              "0.01"              "0.02"
"0.00"
##         Chattogram             Chennai  Chester0le0Street
Christchurch
##             "0.00"              "0.02"              "0.00"
"0.00"
##      Colombo (RPS)       Colombo (SSC)             Cuttack
Dambulla
##             "0.05"              "0.00"              "0.01"
"0.05"
##              Delhi               Dhaka          Dharamsala        Dubai
(DSC)
##             "0.03"              "0.05"              "0.01"
"0.02"
##             Durban           Faridabad             Glasgow         Gros
Islet
##             "0.01"              "0.00"              "0.00"
"0.01"
##           Guwahati             Gwalior          Hambantota
Hamilton
##             "0.01"              "0.01"              "0.01"
"0.01"
##             Harare              Hobart Hyderabad (Deccan)
Indore
##             "0.02"              "0.01"              "0.01"
"0.01"
##             Jaipur          Jamshedpur        Johannesburg
Kanpur
##             "0.01"              "0.01"              "0.01"
```

```
"0.02"
##            Karachi            Kingston               Kochi
Kolkata
##             "0.02"              "0.02"              "0.02"
"0.02"
##       Kuala Lumpur              Lahore               Leeds
Lord's
##             "0.01"              "0.00"              "0.01"
"0.01"
##         Manchester              Margao           Melbourne
Mohali
##             "0.01"              "0.01"              "0.02"
"0.03"
##    Mount Maunganui              Multan              Mumbai         Mumbai
(BS)
##             "0.00"              "0.00"              "0.02"
"0.00"
##             Nagpur              Napier         North Sound
Nottingham
##             "0.03"              "0.01"              "0.01"
"0.01"
##          Pallekele               Perth            Peshawar      Port
Elizabeth
##             "0.01"              "0.01"              "0.00"
"0.01"
##      Port of Spain                Pune              Rajkot
Ranchi
##             "0.02"              "0.01"              "0.02"
"0.01"
##         Rawalpindi         Southampton              Sydney            The
Oval
##             "0.00"              "0.01"              "0.02"
"0.02"
## Thiruvananthapuram            Vadodara       Visakhapatnam
Wellington
##             "0.00"              "0.01"              "0.02"
"0.01"

#find the range of strike rate of the player
s=cricket$strike_rate
range(s)

## [1]   0 400

#compute the mean, variance and standard deviation of strike rate in career
mean(s)

## [1] 74.62334

var(s)
```

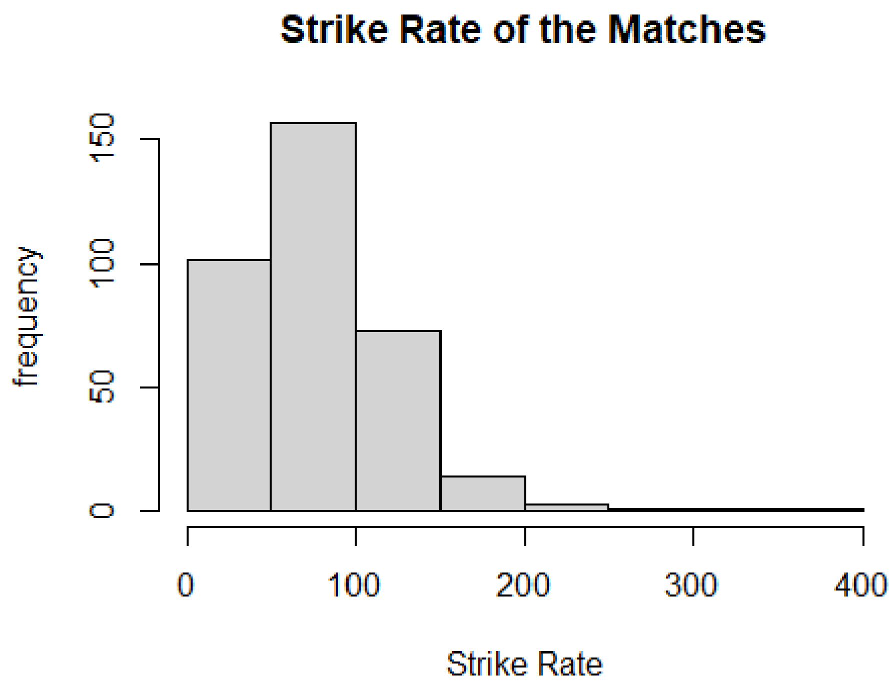
```
## [1] 2908.585

sd(s)

## [1] 53.9313
```

Basic Visualization

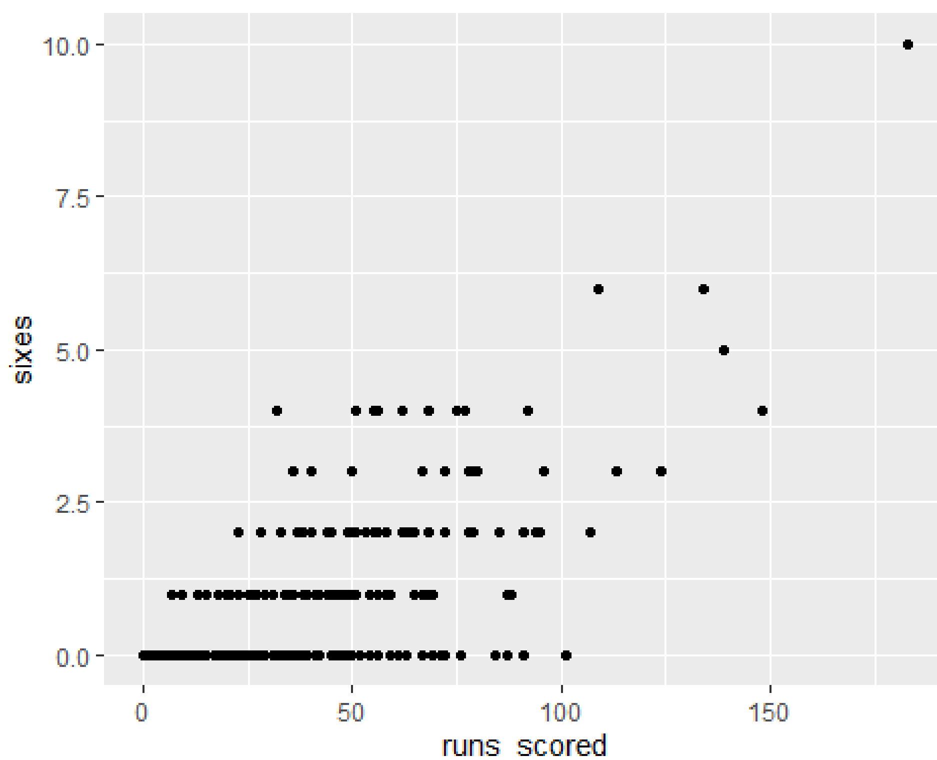
```
#Plot a graph for the number of matches against different oppositions
ggplot(cricket,aes(x=opposition,fill=opposition))+geom_bar()+ggtitle("Number
of matches against different oppositions")
```




```
#Draw the histogram of strike rate of matches
hist(cricket$strike_rate,main="Strike Rate of the Matches",xlab="Strike
Rate",ylab="frequency")
```




```
#Plot relationship of runs scored and number of sixes
ggplot(data=cricket,mapping=aes(x=runs_scored,y=sixes))+geom_point()
```



Inference: The graph shows that majority of the matches were played against Sri Lanka. Strike rate is frequent at the rate of about 50-100. When number of sixes are more, runs scored also increases.

Correlation and Regression Analysis

```
#find correlation between strike rate and runs scored
cor(cricket$strike_rate,cricket$runs_scored)

## [1] 0.4422119

#find correlation between strike rate and balls faced
cor(cricket$strike_rate,cricket$balls_faced)

## [1] 0.2664671

#find correlation between strike rate and fours
cor(cricket$strike_rate,cricket$fours)

## [1] 0.4486592

#find correlation between strike rate and sixes
cor(cricket$strike_rate,cricket$sixes)

## [1] 0.4117161

#to find the most significant variables wrt strike rate
#Split the data into training and testing data set
set.seed(123)
train_samples<-cricket$strike_rate %>%
  createDataPartition(p=0.8,list=FALSE) #80% training and 20% testing
head(train_samples)

##      Resample1
## [1,]         3
## [2,]         4
## [3,]         5
## [4,]         6
## [5,]         7
## [6,]         9

train<-cricket[train_samples,]
test<-cricket[-train_samples,]

#Building a regression model
model<-lm(strike_rate~.,data=train)
summary(model)

##
## Call:
## lm(formula = strike_rate ~ ., data = train)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -93.635  -3.231   0.000   3.997  93.635
##
## Coefficients: (7 not defined because of singularities)
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -2.530e+00  8.053e+01  -0.031 0.975023
## score1                    1.423e+01  2.532e+01   0.562 0.575704
## score1*                   8.985e+01  3.209e+01   2.800 0.006475 **
## score10                   7.617e+01  3.189e+01   2.388 0.019401 *
## score107                  3.038e+02  6.528e+01   4.654 1.35e-05 ***
## score109*                 2.715e+02  7.923e+01   3.427 0.000989 ***
## score11                   9.291e+01  2.713e+01   3.425 0.000996 ***
## score113*                 2.843e+02  7.352e+01   3.867 0.000231 ***
## score12                   1.053e+02  3.554e+01   2.963 0.004071 **
## score12*                  1.021e+02  3.376e+01   3.024 0.003405 **
## score124                  2.952e+02  6.892e+01   4.284 5.31e-05 ***
## score13                   9.836e+01  2.643e+01   3.721 0.000378 ***
## score13*                  1.511e+02  2.579e+01   5.857 1.13e-07 ***
## score139*                 2.805e+02  8.298e+01   3.380 0.001148 **
## score14                   1.058e+02  2.349e+01   4.505 2.37e-05 ***
## score14*                  1.381e+02  3.548e+01   3.893 0.000211 ***
## score148                  2.926e+02  8.840e+01   3.310 0.001429 **
## score15                   8.819e+01  3.234e+01   2.727 0.007937 **
## score15*                  1.248e+02  2.920e+01   4.275 5.48e-05 ***
## score17                   1.337e+02  4.415e+01   3.028 0.003359 **
## score17*                  1.079e+02  3.875e+01   2.784 0.006777 **
## score18                   1.465e+02  2.563e+01   5.717 2.02e-07 ***
## score18*                  1.118e+02  4.137e+01   2.702 0.008496 **
## score183*                 3.274e+02  1.204e+02   2.720 0.008093 **
## score19                   1.180e+02  2.452e+01   4.811 7.43e-06 ***
## score19*                  1.305e+02  5.617e+01   2.324 0.022812 *
## score2                    3.157e+01  2.095e+01   1.507 0.135888
## score2*                   2.362e+01  5.792e+01   0.408 0.684611
## score20                   1.109e+02  2.686e+01   4.128 9.28e-05 ***
## score21                   8.652e+01  3.864e+01   2.239 0.028082 *
## score21*                  1.791e+02  3.150e+01   5.687 2.29e-07 ***
## score23                   1.575e+02  2.617e+01   6.017 5.82e-08 ***
## score23*                  1.246e+02  2.809e+01   4.437 3.04e-05 ***
## score24                   1.141e+02  2.785e+01   4.096 0.000104 ***
## score25                   1.431e+02  2.517e+01   5.684 2.32e-07 ***
## score25*                  1.431e+02  3.261e+01   4.388 3.64e-05 ***
## score26                   1.316e+02  2.728e+01   4.824 7.09e-06 ***
## score26*                  1.232e+02  4.184e+01   2.945 0.004284 **
## score27                   1.490e+02  2.672e+01   5.578 3.56e-07 ***
## score28                   1.416e+02  3.447e+01   4.107 0.000100 ***
## score28*                  2.117e+02  4.004e+01   5.286 1.16e-06 ***
## score29                   1.160e+02  3.630e+01   3.195 0.002038 **
## score29*                  1.370e+02  4.407e+01   3.109 0.002639 **
## score3                    1.536e+01  2.966e+01   0.518 0.606046
## score3*                   2.992e+01  4.850e+01   0.617 0.539137
```

```
## score31                       1.458e+02  2.833e+01   5.146 2.02e-06 ***
## score32                       1.296e+02  3.479e+01   3.725 0.000373 ***
## score33                       1.408e+02  3.434e+01   4.101 0.000102 ***
## score34                       1.447e+02  3.091e+01   4.680 1.22e-05 ***
## score35                       1.431e+02  3.021e+01   4.735 9.93e-06 ***
## score35*                      1.617e+02  2.987e+01   5.412 6.99e-07 ***
## score36                       1.528e+02  3.355e+01   4.554 1.97e-05 ***
## score37                       1.603e+02  3.602e+01   4.451 2.88e-05 ***
## score37*                      1.657e+02  4.860e+01   3.411 0.001041 **
## score38                       1.738e+02  4.046e+01   4.295 5.10e-05 ***
## score38*                      1.532e+02  4.016e+01   3.814 0.000276 ***
## score39                       1.650e+02  3.408e+01   4.842 6.60e-06 ***
## score4                        5.151e+01  2.369e+01   2.175 0.032773 *
## score4*                       2.594e+02  2.706e+01   9.584 1.02e-14 ***
## score40                       1.453e+02  4.606e+01   3.155 0.002302 **
## score40*                      2.098e+02  4.461e+01   4.702 1.13e-05 ***
## score41                       1.621e+02  3.481e+01   4.656 1.34e-05 ***
## score42                       1.599e+02  4.657e+01   3.434 0.000966 ***
## score42*                      1.645e+02  3.309e+01   4.970 4.03e-06 ***
## score44                       1.574e+02  4.162e+01   3.783 0.000307 ***
## score44*                      1.730e+02  5.989e+01   2.890 0.005023 **
## score45*                      1.716e+02  3.420e+01   5.018 3.34e-06 ***
## score46*                      1.800e+02  3.937e+01   4.572 1.84e-05 ***
## score47                       1.869e+02  3.696e+01   5.057 2.87e-06 ***
## score48                       2.004e+02  4.289e+01   4.672 1.26e-05 ***
## score48*                      1.793e+02  6.443e+01   2.783 0.006801 **
## score49                       2.117e+02  5.025e+01   4.212 6.89e-05 ***
## score5                        5.566e+01  2.207e+01   2.522 0.013770 *
## score50                       2.270e+02  4.200e+01   5.405 7.20e-07 ***
## score50*                      2.092e+02  4.385e+01   4.771 8.68e-06 ***
## score51                       2.037e+02  4.584e+01   4.445 2.95e-05 ***
## score51*                      1.675e+02  4.308e+01   3.888 0.000215 ***
## score52                       1.866e+02  4.495e+01   4.152 8.54e-05 ***
## score53                       1.875e+02  4.258e+01   4.404 3.43e-05 ***
## score55*                      1.836e+02  6.247e+01   2.939 0.004363 **
## score56                       1.909e+02  3.776e+01   5.055 2.89e-06 ***
## score56*                      1.842e+02  5.450e+01   3.379 0.001149 **
## score58                       1.980e+02  5.307e+01   3.732 0.000365 ***
## score59*                      2.232e+02  5.471e+01   4.080 0.000110 ***
## score6                        7.655e+01  2.203e+01   3.475 0.000848 ***
## score61*                      2.077e+02  4.188e+01   4.960 4.19e-06 ***
## score62                       1.778e+02  4.618e+01   3.850 0.000244 ***
## score62*                      1.989e+02  5.056e+01   3.934 0.000183 ***
## score63                       1.776e+02  4.635e+01   3.832 0.000260 ***
## score64                       1.683e+02  4.500e+01   3.740 0.000355 ***
## score65                       1.964e+02  4.725e+01   4.156 8.41e-05 ***
## score67                       2.338e+02  4.712e+01   4.961 4.17e-06 ***
## score67*                      1.970e+02  4.247e+01   4.638 1.44e-05 ***
## score68                       1.970e+02  6.793e+01   2.901 0.004870 **
## score68*                      2.281e+02  6.440e+01   3.542 0.000683 ***
```

```
## score69                    2.256e+02  4.705e+01   4.794 7.94e-06 ***
## score7                     9.277e+01  2.043e+01   4.541 2.07e-05 ***
## score7*                    3.147e+02  3.142e+01  10.014 1.56e-15 ***
## score71                    2.413e+02  4.555e+01   5.296 1.11e-06 ***
## score71*                   2.394e+02  5.344e+01   4.480 2.59e-05 ***
## score72                    1.927e+02  5.310e+01   3.630 0.000512 ***
## score72*                   2.007e+02  7.910e+01   2.537 0.013226 *
## score75*                   2.180e+02  5.316e+01   4.101 0.000102 ***
## score76                    2.386e+02  5.202e+01   4.586 1.74e-05 ***
## score77*                   1.947e+02  6.048e+01   3.219 0.001893 **
## score78*                   2.462e+02  6.172e+01   3.989 0.000152 ***
## score79                    2.221e+02  5.894e+01   3.769 0.000322 ***
## score8                     6.153e+01  2.692e+01   2.286 0.025038 *
## score80                    2.230e+02  5.605e+01   3.979 0.000157 ***
## score85*                   2.856e+02  6.823e+01   4.185 7.57e-05 ***
## score87*                   2.717e+02  5.793e+01   4.690 1.18e-05 ***
## score88*                   2.540e+02  5.630e+01   4.512 2.30e-05 ***
## score9                     1.430e+02  2.390e+01   5.984 6.68e-08 ***
## score9*                    1.190e+02  3.173e+01   3.751 0.000341 ***
## score91*                   2.645e+02  5.251e+01   5.037 3.11e-06 ***
## score92*                   2.265e+02  7.142e+01   3.172 0.002185 **
## score94                    2.704e+02  5.536e+01   4.885 5.61e-06 ***
## score95                    3.123e+02  6.753e+01   4.625 1.51e-05 ***
## scoreDNB                  -2.211e+01  1.534e+01  -1.442 0.153480
## scoreTDNB                 -1.497e+01  2.279e+01  -0.657 0.513376
## runs_scored                       NA         NA      NA       NA
## balls_faced               -2.185e+00  3.622e-01  -6.034 5.42e-08 ***
## fours                      2.234e+00  2.969e+00   0.753 0.454045
## sixes                      7.509e+00  5.486e+00   1.369 0.175110
## oppositionAfrica XI        5.340e+00  5.213e+01   0.102 0.918683
## oppositionAustralia        2.946e+01  4.531e+01   0.650 0.517549
## oppositionBangladesh       1.841e+01  4.894e+01   0.376 0.707869
## oppositionBermuda          6.413e+01  6.358e+01   1.009 0.316329
## oppositionEngland          1.932e+01  4.780e+01   0.404 0.687202
## oppositionHong Kong        5.917e+00  5.825e+01   0.102 0.919363
## oppositionIreland          1.951e+01  5.939e+01   0.328 0.743448
## oppositionNetherlands             NA         NA      NA       NA
## oppositionNew Zealand      2.783e+01  4.832e+01   0.576 0.566382
## oppositionPakistan         3.402e+01  4.956e+01   0.686 0.494526
## oppositionScotland         2.900e+01  7.331e+01   0.396 0.693525
## oppositionSouth Africa     1.874e+01  4.689e+01   0.400 0.690499
## oppositionSri Lanka        1.987e+01  4.780e+01   0.416 0.678790
## oppositionWest Indies      2.102e+01  4.839e+01   0.434 0.665285
## oppositionZimbabwe         3.572e+01  5.763e+01   0.620 0.537204
## groundAhmedabad           -1.472e+01  5.011e+01  -0.294 0.769779
## groundAuckland            -6.848e+01  5.550e+01  -1.234 0.220989
## groundBasseterre           9.808e+00  6.403e+01   0.153 0.878664
## groundBelfast             -3.133e+00  5.708e+01  -0.055 0.956377
## groundBengaluru            1.590e+01  5.002e+01   0.318 0.751486
## groundBirmingham           5.463e+00  5.039e+01   0.108 0.913941
```

```
## groundBrisbane            1.459e+01  4.997e+01   0.292 0.771131
## groundBulawayo           -1.818e+00  5.532e+01  -0.033 0.973868
## groundCanberra            1.063e+01  5.593e+01   0.190 0.849823
## groundCape Town           6.715e+00  5.955e+01   0.113 0.910509
## groundCardiff             1.770e+00  4.215e+01   0.042 0.966621
## groundCenturion           8.005e+00  5.062e+01   0.158 0.874770
## groundChandigarh         -3.109e+01  4.142e+01  -0.751 0.455142
## groundChennai             1.459e+01  5.818e+01   0.251 0.802740
## groundChester0le0Street   1.309e+01  5.970e+01   0.219 0.827090
## groundChristchurch               NA         NA      NA       NA
## groundColombo (RPS)       5.161e-01  4.536e+01   0.011 0.990953
## groundColombo (SSC)       9.011e+00  5.177e+01   0.174 0.862281
## groundCuttack            -2.642e+01  5.452e+01  -0.485 0.629402
## groundDambulla            4.192e+00  4.636e+01   0.090 0.928193
## groundDelhi               7.205e+00  4.810e+01   0.150 0.881316
## groundDhaka               2.035e+01  4.818e+01   0.422 0.673924
## groundDharamsala          2.380e+01  5.402e+01   0.440 0.660831
## groundDubai (DSC)         8.808e+00  5.149e+01   0.171 0.864642
## groundDurban             -3.872e+01  5.000e+01  -0.774 0.441109
## groundFaridabad          -1.553e+00  6.423e+01  -0.024 0.980773
## groundGlasgow                    NA         NA      NA       NA
## groundGuwahati            1.839e+01  5.298e+01   0.347 0.729521
## groundGwalior            -1.459e+01  5.917e+01  -0.247 0.805931
## groundHambantota          1.435e+01  5.648e+01   0.254 0.800151
## groundHamilton           -4.730e+00  5.456e+01  -0.087 0.931142
## groundHarare             -1.260e+01  5.825e+01  -0.216 0.829341
## groundHobart              9.651e+00  5.183e+01   0.186 0.852771
## groundHyderabad (Deccan) -2.609e+01  5.697e+01  -0.458 0.648303
## groundIndore              1.179e+01  6.145e+01   0.192 0.848344
## groundJaipur             -6.166e+00  4.938e+01  -0.125 0.900961
## groundJamshedpur         -1.102e+01  6.425e+01  -0.172 0.864238
## groundJohannesburg        8.125e+00  5.174e+01   0.157 0.875628
## groundKanpur              1.005e+01  5.128e+01   0.196 0.845173
## groundKarachi            -5.553e+00  5.243e+01  -0.106 0.915935
## groundKingston            2.534e+00  4.908e+01   0.052 0.958968
## groundKochi               1.812e+01  5.205e+01   0.348 0.728656
## groundKolkata            -7.192e+00  4.925e+01  -0.146 0.884279
## groundKuala Lumpur       -1.674e+01  4.970e+01  -0.337 0.737123
## groundLahore                     NA         NA      NA       NA
## groundLeeds               2.781e+01  5.085e+01   0.547 0.586034
## groundLord's             -2.907e+01  6.052e+01  -0.480 0.632348
## groundManchester          6.452e+00  5.302e+01   0.122 0.903457
## groundMargao              4.483e+01  5.237e+01   0.856 0.394611
## groundMelbourne           1.910e+01  5.106e+01   0.374 0.709375
## groundMohali              3.309e+00  4.785e+01   0.069 0.945042
## groundMount Maunganui            NA         NA      NA       NA
## groundMultan                     NA         NA      NA       NA
## groundMumbai             -1.662e+01  4.835e+01  -0.344 0.731977
## groundMumbai (BS)         1.134e+01  5.851e+01   0.194 0.846818
## groundNagpur             -1.048e+01  5.141e+01  -0.204 0.839075
```

```
## groundNapier              2.494e+00  5.557e+01   0.045 0.964321
## groundNottingham          1.072e+01  5.275e+01   0.203 0.839571
## groundPallekele           3.484e+01  5.065e+01   0.688 0.493583
## groundPerth               5.800e+00  5.111e+01   0.113 0.909960
## groundPort Elizabeth      9.164e-01  5.249e+01   0.017 0.986118
## groundPort of Spain      -1.460e+01  4.993e+01  -0.292 0.770738
## groundPune               -6.653e+00  5.181e+01  -0.128 0.898171
## groundRajkot              3.510e+00  5.065e+01   0.069 0.944932
## groundRanchi             -6.808e+00  5.053e+01  -0.135 0.893185
## groundRawalpindi         -5.197e+00  5.509e+01  -0.094 0.925084
## groundSouthampton         2.748e+01  5.373e+01   0.511 0.610504
## groundSydney              3.480e+00  5.051e+01   0.069 0.945256
## groundThe Oval            2.379e+01  4.944e+01   0.481 0.631841
## groundThiruvananthapuram  9.281e+00  5.534e+01   0.168 0.867257
## groundVadodara           -5.021e+00  5.404e+01  -0.093 0.926210
## groundVisakhapatnam       5.566e+00  4.870e+01   0.114 0.909313
## groundWellington          5.805e+00  5.191e+01   0.112 0.911248
## date                     -3.668e-09  2.475e-08  -0.148 0.882578
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.6 on 76 degrees of freedom
## Multiple R-squared:  0.9482, Adjusted R-squared:  0.8084
## F-statistic: 6.785 on 205 and 76 DF,  p-value: < 2.2e-16
```

Inference: Here balls_faced and score columns show to have high significance while predicting the model.

Conclusion:

The cricket data is loaded, and then data cleaning and pre-processing are applied to it. The data is subjected to statistical analysis and exploratory analysis. Then visualization of the number of matched and strike rates is done to gain more inferences. Then finally, correlation and regression analysis are performed, which revealed that the number of balls encountered and the score had a significant impact on the model prediction.