

IPL1.R

2023-06-06

```
library(dplyr)

##
## Attaching package: 'dplyr'

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

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

library(tidyr)
library(readxl)

## Warning: package 'readxl' was built under R version 4.2.3

library(readr)

## Warning: package 'readr' was built under R version 4.2.3

df_ipl <-
read.csv("C:\\Users\\OneDrive\\Desktop\\BA2\\EDA\\IPL_Ball_by_Ball_2008_202
2.csv")
ipl_salary <- read.csv("C:\\Users\\OneDrive\\Desktop\\BA2\\EDA\\ipl
salary.csv")
ipl_match <- read.csv("C:\\Users\\OneDrive\\Desktop\\BA2\\EDA\\IPL Matches
2008-2020.csv")

df_ipl$match = as.numeric(substr(df_ipl$ID, 1,3))
unique(df_ipl$match)

## [1] 131 130 125 123 121 118 117 113 108 981 980 829 734 733 729 598 597
548 501
## [20] 419 392 336 335

table(df_ipl$match)

##
##   108   113   117   118   121   123   125   130   131   335   336   392
419
## 13862 14286 13337   975 13580   979 14425 16935   977 4261  9228 13606
14498
##   501   548   597   598   729   733   734   829   980   981
## 17013 17767   486 17691  4763  3534  6003 13652 11690  2406
```

```

library(dplyr)

df_ip1$new = ifelse(df_ip1$match==118,117,
                    ifelse(df_ip1$match==123,121,
                            ifelse(df_ip1$match==131,130,
                                    ifelse(df_ip1$match==981,980,
                                            ifelse(df_ip1$match==336,335,
                                                    ifelse(df_ip1$match==597,598,

ifelse(df_ip1$match==733,729,

ifelse(df_ip1$match==734,729,

df_ip1$match)))))))))
table(df_ip1$new)

##
##  108   113   117   121   125   130   335   392   419   501   548   598
729
## 13862 14286 14312 14559 14425 17912 13489 13606 14498 17013 17767 18177
14300
##   829   980
## 13652 14096

df_ip1$season <- ifelse(df_ip1$new == 392, "2009",
                        ifelse(df_ip1$new == 419, "2010",
                                ifelse(df_ip1$new == 501, "2011",
                                        ifelse(df_ip1$new == 548, "2012",
                                                ifelse(df_ip1$new == 598, "2013",
                                                        ifelse(df_ip1$new ==
729, "2014",
                                                                    ifelse(df_ip1$new
== 829, "2015",

ifelse(df_ip1$new == 980, "2016",
ifelse(df_ip1$new == 108, "2017",
ifelse(df_ip1$new == 113, "2018",
ifelse(df_ip1$new == 117, "2019",
ifelse(df_ip1$new == 121, "2020",
ifelse(df_ip1$new == 125, "2021",
ifelse(df_ip1$new == 130, "2022", "unkown")))))))))))

df_ip1$season

```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```

fix( df_ip1)
####b) Arrange the data IPL round wise and batsman and ball and runs
####and wickets per player per match.
####Indicate the top three run getters and low three wicket takers in each
player IPL round
player = df_ip1 %>%
  select (ID, batter, total_run, bowler,isWicketDelivery,season)%>%
  filter (season== 2020|season== 2021|
          season== 2022)%>%
  group_by (season, batter)%>%
  summarise (total= sum (total_run))%>%
  arrange (season, -total)

## `summarise()` has grouped output by 'season'. You can override using the
## `.groups` argument.

fix(player)

player1 = df_ip1 %>%
  select(ID, batter, total_run, bowler, isWicketDelivery, season) %>%
  filter(season== 2020 |season== 2021 | season== 2022)%>%
  group_by(season, bowler)%>%
  summarise(total= sum(isWicketDelivery))%>%
  arrange(season, -total)

## `summarise()` has grouped output by 'season'. You can override using the
## `.groups` argument.

fix(player1)

rahul = player[player$batter=='KL Rahul',]

library (MASS)

## Warning: package 'MASS' was built under R version 4.2.3
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##      select
fitdistr (rahul$total, "lognormal")

##      meanlog      sdlog
## 6.49975769 0.03245790
## (0.01873958) (0.01325088)

fitdistr (rahul$total, "exponential")

```

```

##          rate
## 0.0015030060
## (0.0008677609)

fitdistr (rahul$total, "gamma")

##          shape          rate
## 619.6945106      0.9314055
## (504.4548691) ( 0.7585058)

fitdistr (rahul$total, "normal")

##          mean          sd
## 665.333333      21.822516
## ( 12.599236) ( 8.909005)

names(ipl_salary)

## [1] "Id"          "Name"          "Year"          "Final.Price"  "Role"
## [6] "Nationality"  "Team"          "Ent"           "Age"          "Matches"
## [11] "LMatches"     "Runs"          "LRuns"         "HS"           "LHS"
## [16] "Ave"          "LAve"          "StrRate"       "LStrRate"     "Fifties"
## [21] "LFifties"     "Hundreds"      "LHundreds"     "Fours"        "LFours"
## [26] "Sixes"        "LSixes"        "Catches"       "LCatches"     "Stumps"
## [31] "LStumps"      "Wkts"          "LWkts"         "Econ"         "LEcon"
## [36] "FourWkts"     "LFourWkts"     "FiveWkts"      "LFiveWkts"    "Indian"
## [41] "Specialist"   "Status"

x=ipl_salary$Runs
y=ipl_salary$Final.Price
cor(x,y)

## [1] 0.4712774

library(caTools)

## Warning: package 'caTools' was built under R version 4.2.3

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

df<-data.frame(x,y)

set.seed(123)
split <- sample.split(df$y, SplitRatio = 0.8)
train <- subset(df, split == TRUE)
test  <- subset(df, split == FALSE)

model <- lm(y ~ x,data = train)

```

```
prediction <- predict(model, newdata = test)

R_squared <- cor(prediction, test$y)^2

print(paste("R-squared:", R_squared))

## [1] "R-squared: 0.333813797229825"

MAPE <- function(actual, predicted) {
  mean(abs((actual - predicted) / actual))
}

mape_value <- MAPE(test$y, prediction)
print(mape_value)

## [1] 4.837156
```