

IPL Auctions over the years:An overview

Sanjay chauhan

2/23/2022

Introduction

The following report shows an overview of auctions of Indian Premier League over the years. This report contains the [data](#) of 9 years i.e. from 2013 to 2022. This reflects some of the key factors of IPL Auctions such as “Most expensive players of each year and franchises of respective players”, “Popularity of Indian and Overseas players”, “Franchises with most amount spent” to name some. This analysis based upon yearly analysis i.e. each year analysed separately. In the end we mentioned the list of top players who made most appearances in the IPL Auctions of the given time period.

Now, we look at the [data](#)

```
df <- read_csv("IPL_Auction.csv")
## Rows: 970 Columns: 7
## -- Column specification -----
## Delimiter: ","
## chr (4): Player, Role, Team, Player Origin
## dbl (3): sr.no, Amount, Year
##
## 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.
view(df)
```

Data cleaning process-

Now, let's have a look at the datatypes of each column rectify it if any;

```
glimpse(df)
## Rows: 970
## Columns: 7
## $ sr.no          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16,~
## $ Player         <chr> "Aaron Finch", "Aaron Finch", "Aaron Finch",
"Aaron Fi~
## $ Role           <chr> "Batsman", "Batsman", "Batsman", "Batsman",
"Batsman",~
```

```
## $ Amount          <dbl> 40000000, 32000000, 10000000, 62000000, 44000000,
2000~
## $ Team            <chr> "Sunrisers Hyderabad", "Mumbai Indians", "Gujarat
Lion~
## $ Year            <dbl> 2014, 2015, 2016, 2018, 2020, 2020, 2022, 2015,
2016, ~
## $ `Player Origin` <chr> "Overseas", "Overseas", "Overseas", "Overseas",
"Overs~
```

Making column names consistent-

```
df <- rename(df, Player_Origin="Player Origin",
             sr_no=sr.no)
df <- rename_all(df, tolower)
```

If we look at the “player.origin” and “Role” both are characters as of now but it would be suitable converting them into factors since both are categorical

```
df$player_origin <- as.factor(df$player_origin)
class(df$player_origin)
## [1] "factor"
levels(df$player_origin)
## [1] "Indian" "Overseas"
df$role <- as.factor(df$role)
class(df$role)
## [1] "factor"
levels(df$role)
## [1] "All-Rounder" "Batsman" "Bowler" "Wicket Keeper"
```

Now, let’s have a look into dataset if there are some NA values-

```
df %>% map(~sum(is.na(.)))
## $sr_no
## [1] 0
##
## $player
## [1] 0
##
## $role
## [1] 0
##
## $amount
## [1] 0
##
## $team
## [1] 0
##
## $year
## [1] 1
##
## $player_origin
## [1] 0
```

So in the given data there’s one NA in “Year”. We’ll try to find why it’s not available-

```
df %>% filter(!complete.cases(.))
## # A tibble: 1 x 7
##   sr_no player      role    amount team      year player_origin
##   <dbl> <chr>      <fct>    <dbl> <chr>    <dbl> <fct>
## 1    314 Ishan Porel Bowler 2000000 Kings XI Punjab    NA Indian
```

Let's look around sr_no. 314 and try to figure out if there is consistency in years

```
df %>%
  filter(sr_no >= 290 & sr_no <=350) %>% view()
```

Examining conditional data, any sort of consistency cannot be determined. Due to unavailability of other alternatives we have to drop that attribute so that it does not halt analysis process in later phases-

```
df <- df %>% drop_na(year)
df %>% map(~sum(is.na(.)))
## $sr_no
## [1] 0
##
## $player
## [1] 0
##
## $role
## [1] 0
##
## $amount
## [1] 0
##
## $team
## [1] 0
##
## $year
## [1] 0
##
## $player_origin
## [1] 0
```

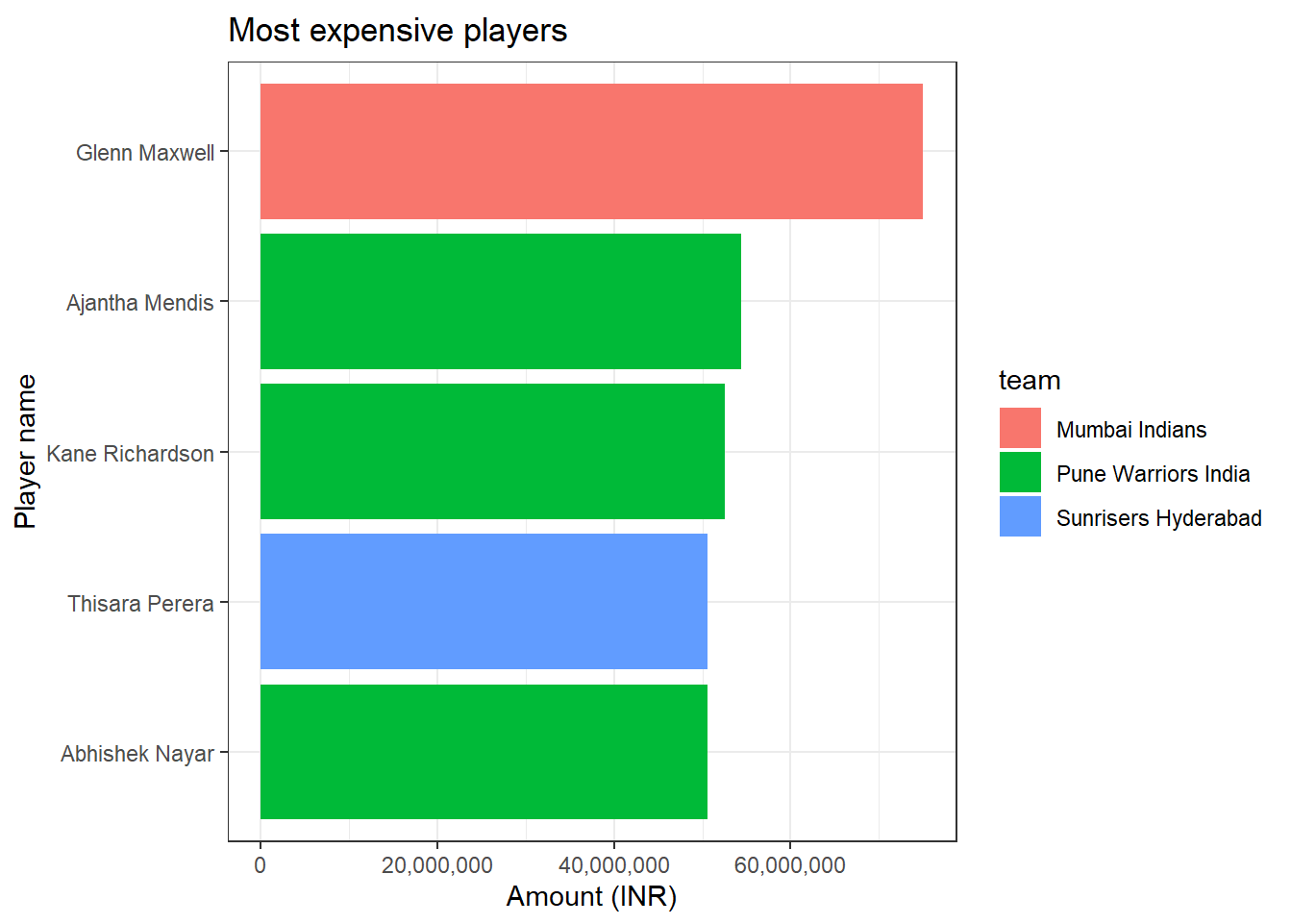
After completing data wrangling process and examining it, it seems accurate, clean and consistent and we can proceed to next phase i.e. analysis phase-

Exploratory Analysis-

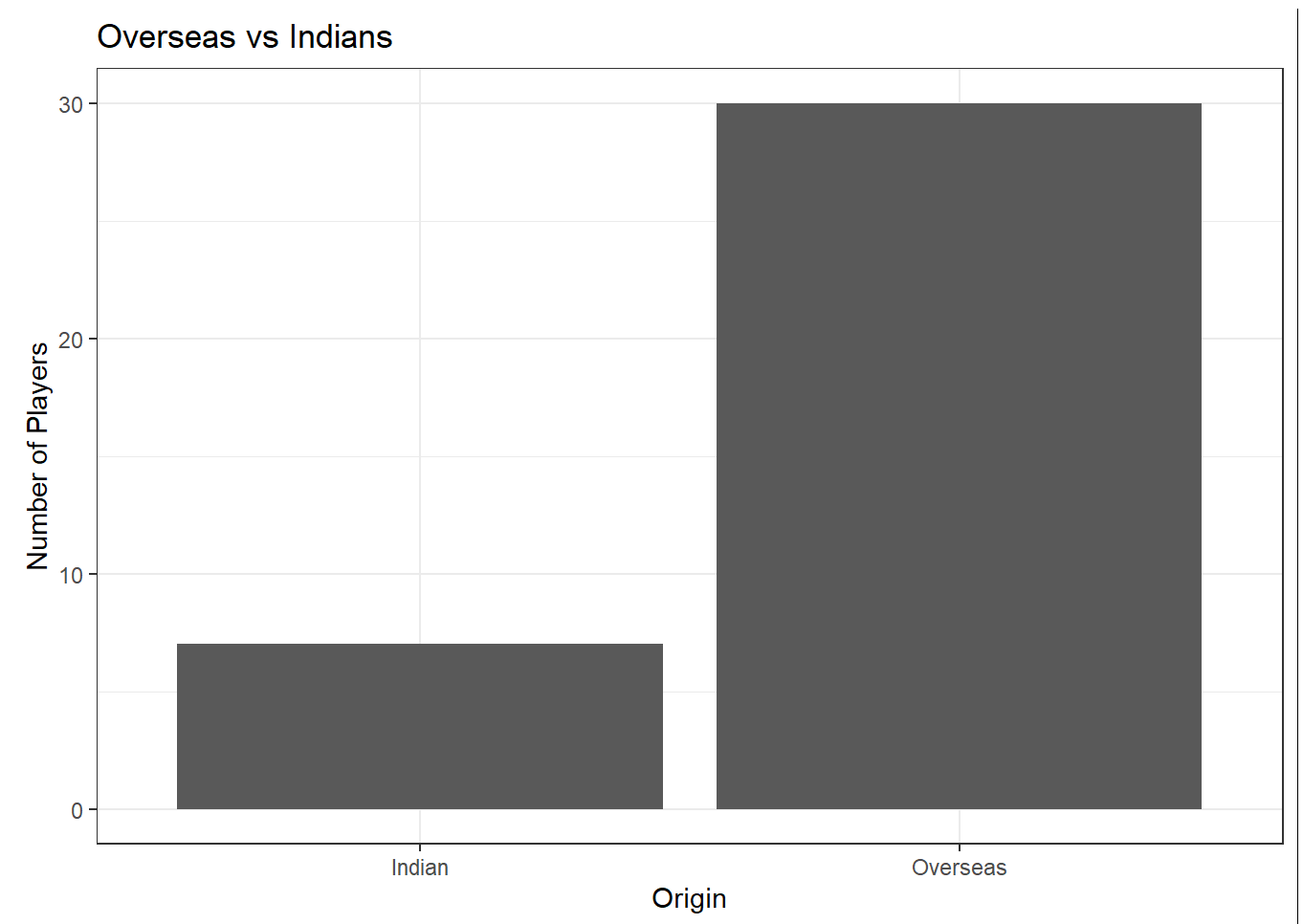
Year wise data processing

2013

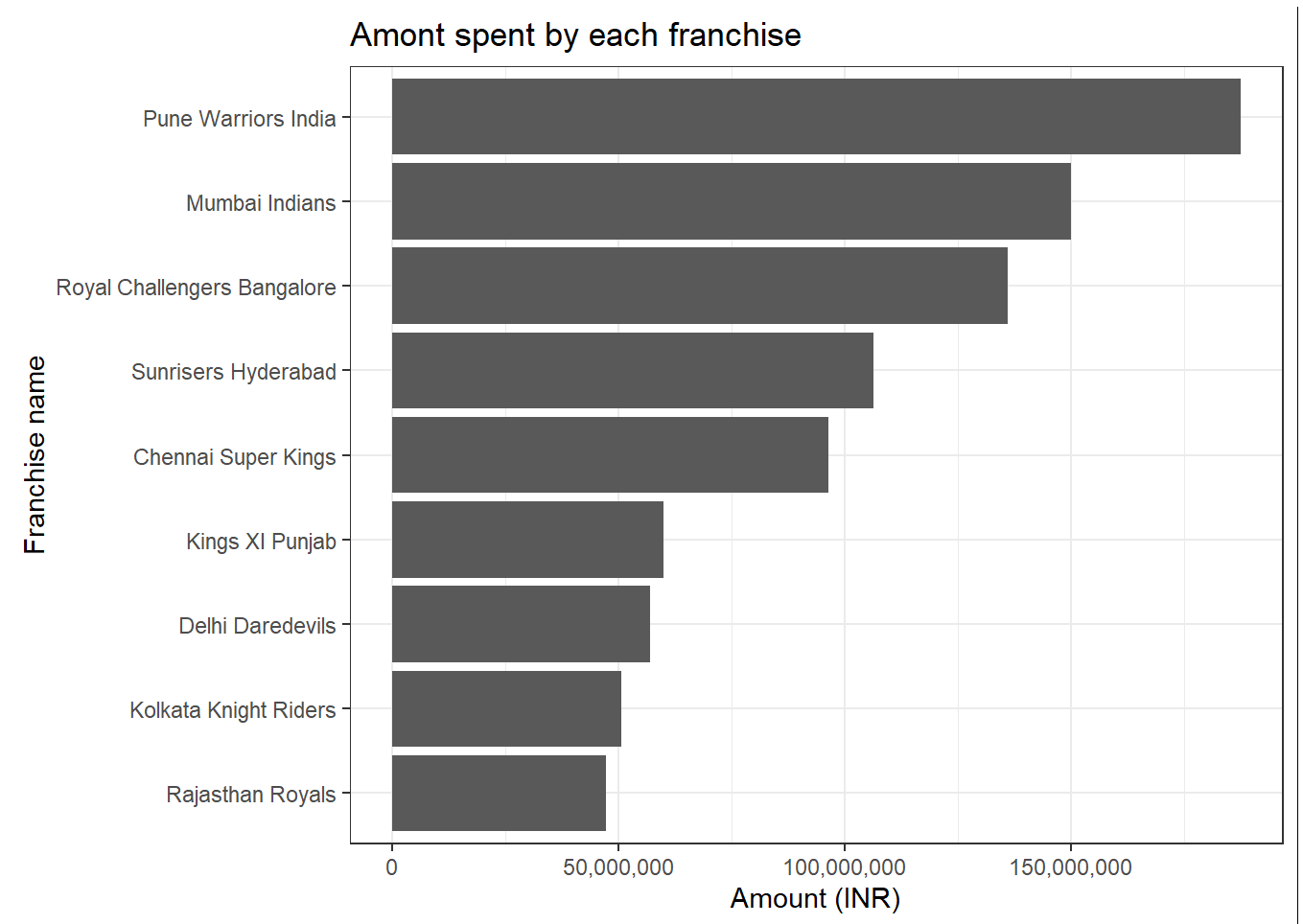
Most expensive players-



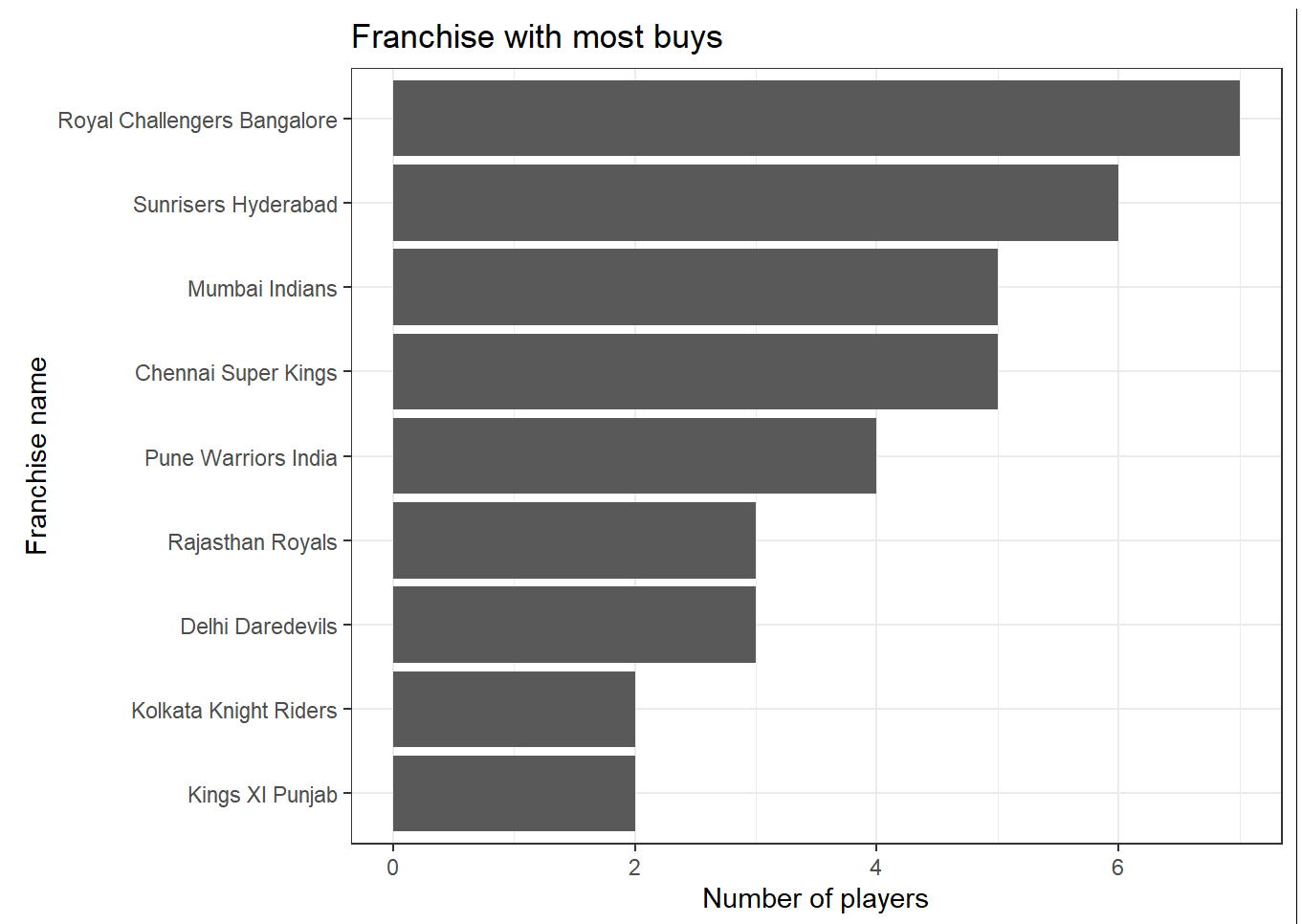
Overseas vs Indian players-



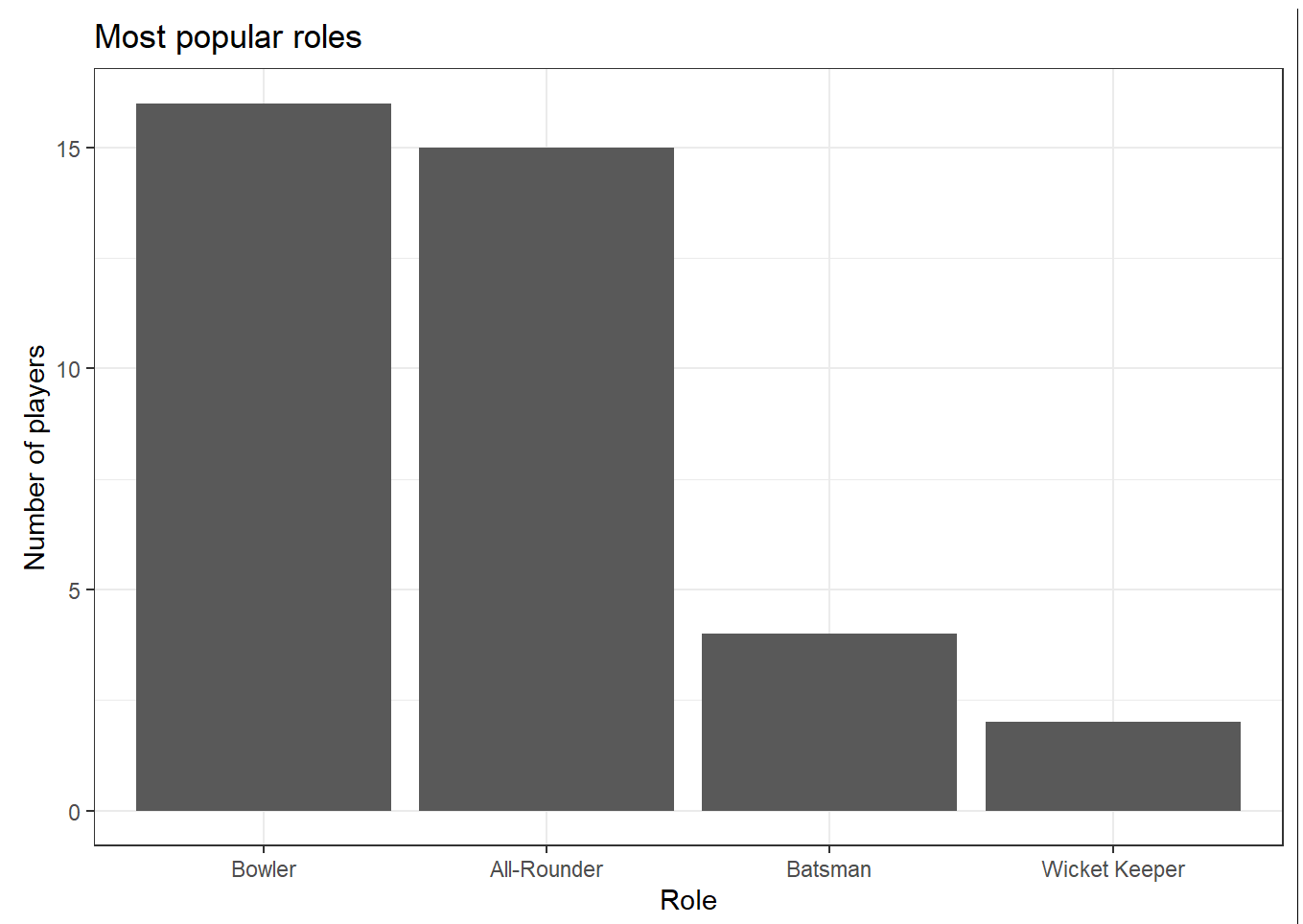
Team with most amount spent



Team with most buys-



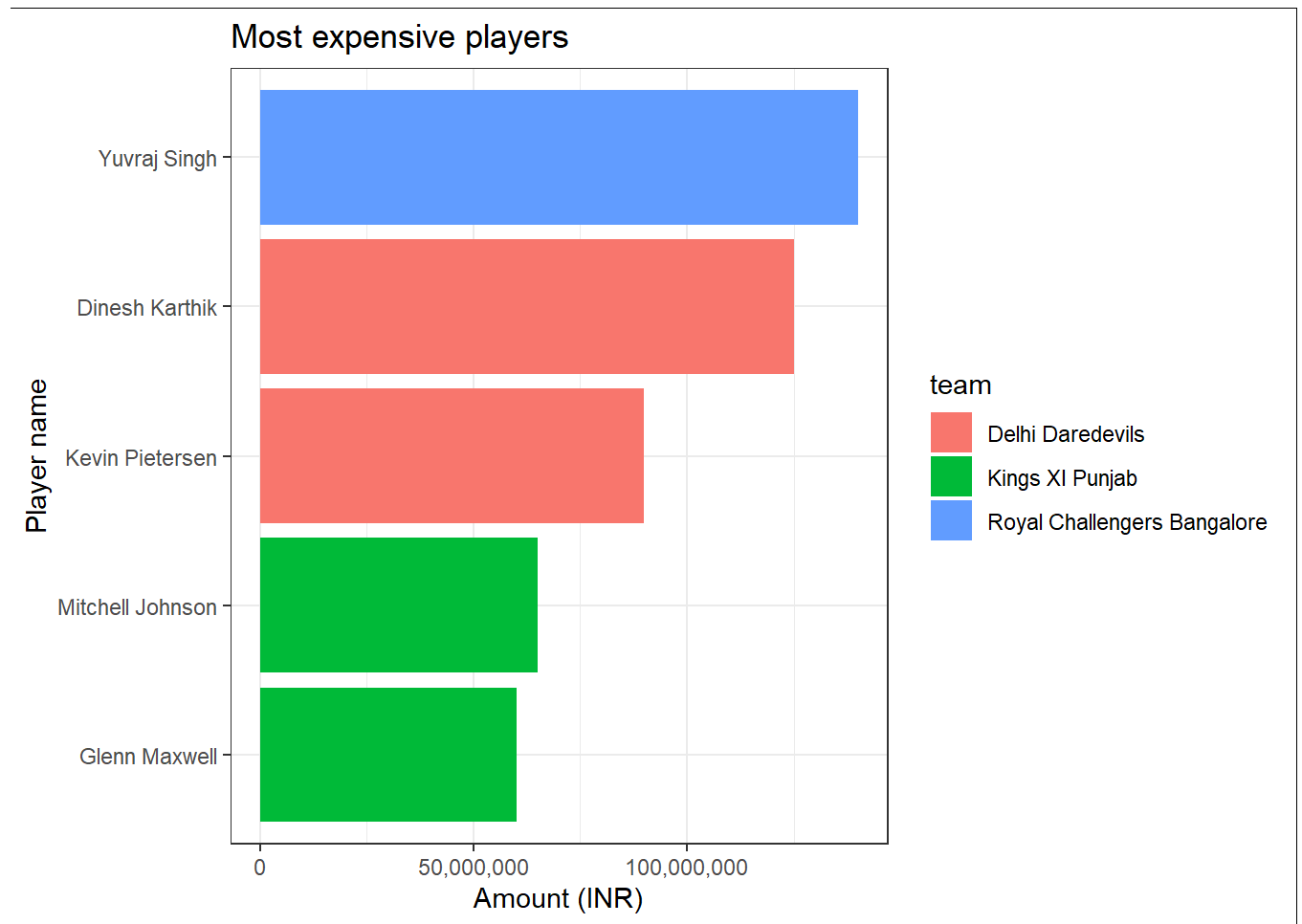
Most popular roles-



2014

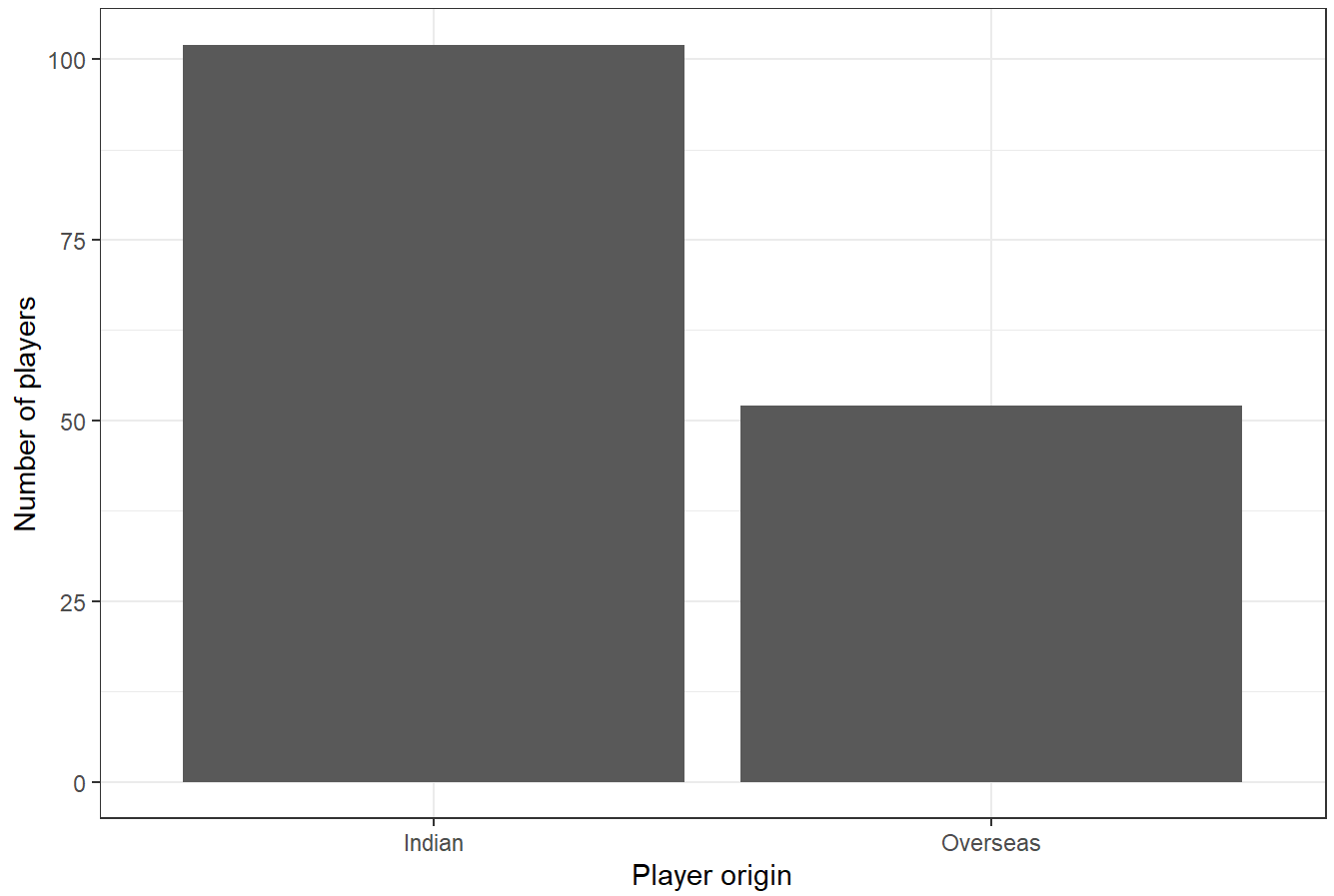
Most expensive players-

``summarise()`` has grouped output by 'player', 'team'. You can override using the ``.groups`` argument.

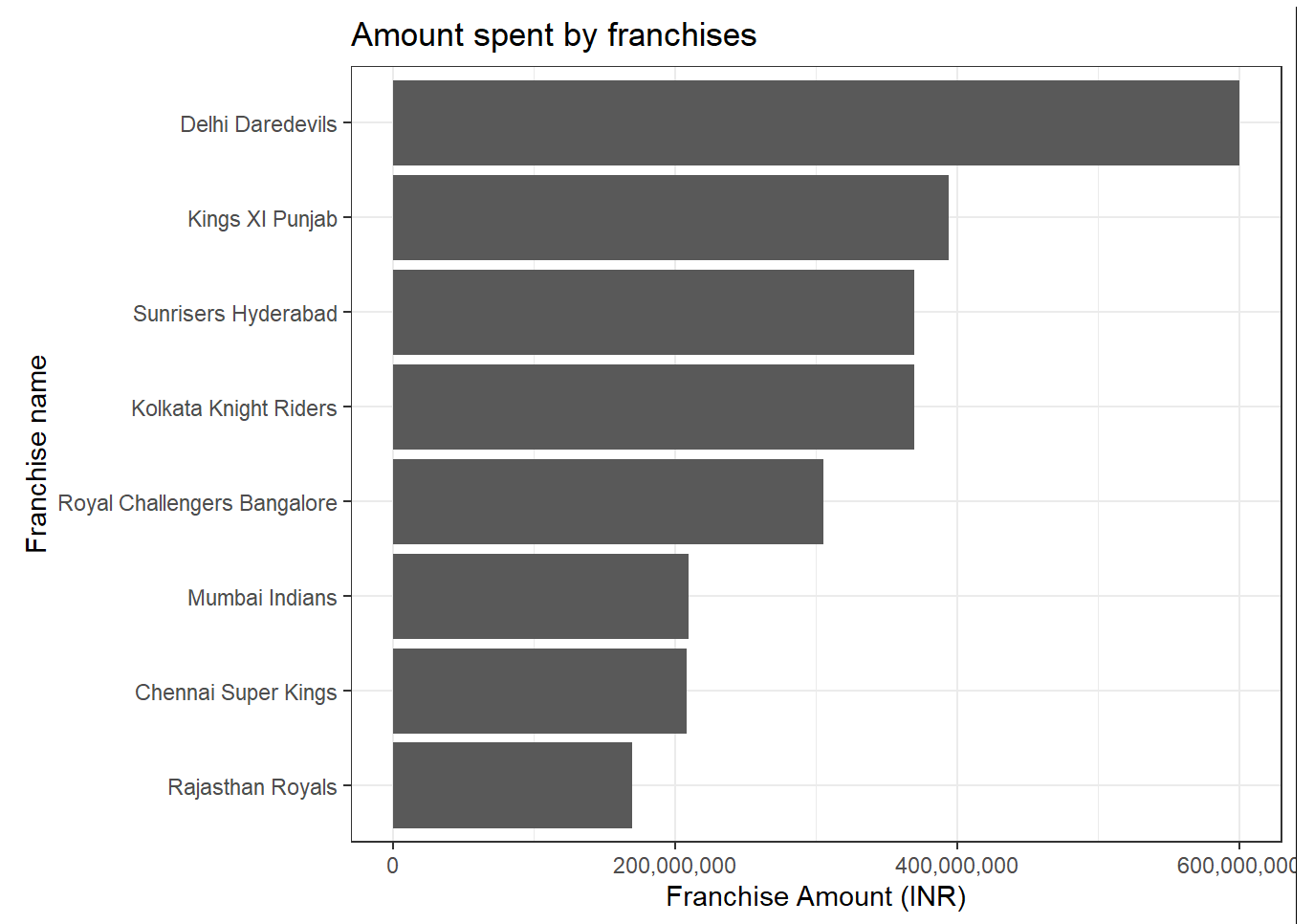


Overseas vs Indian players-

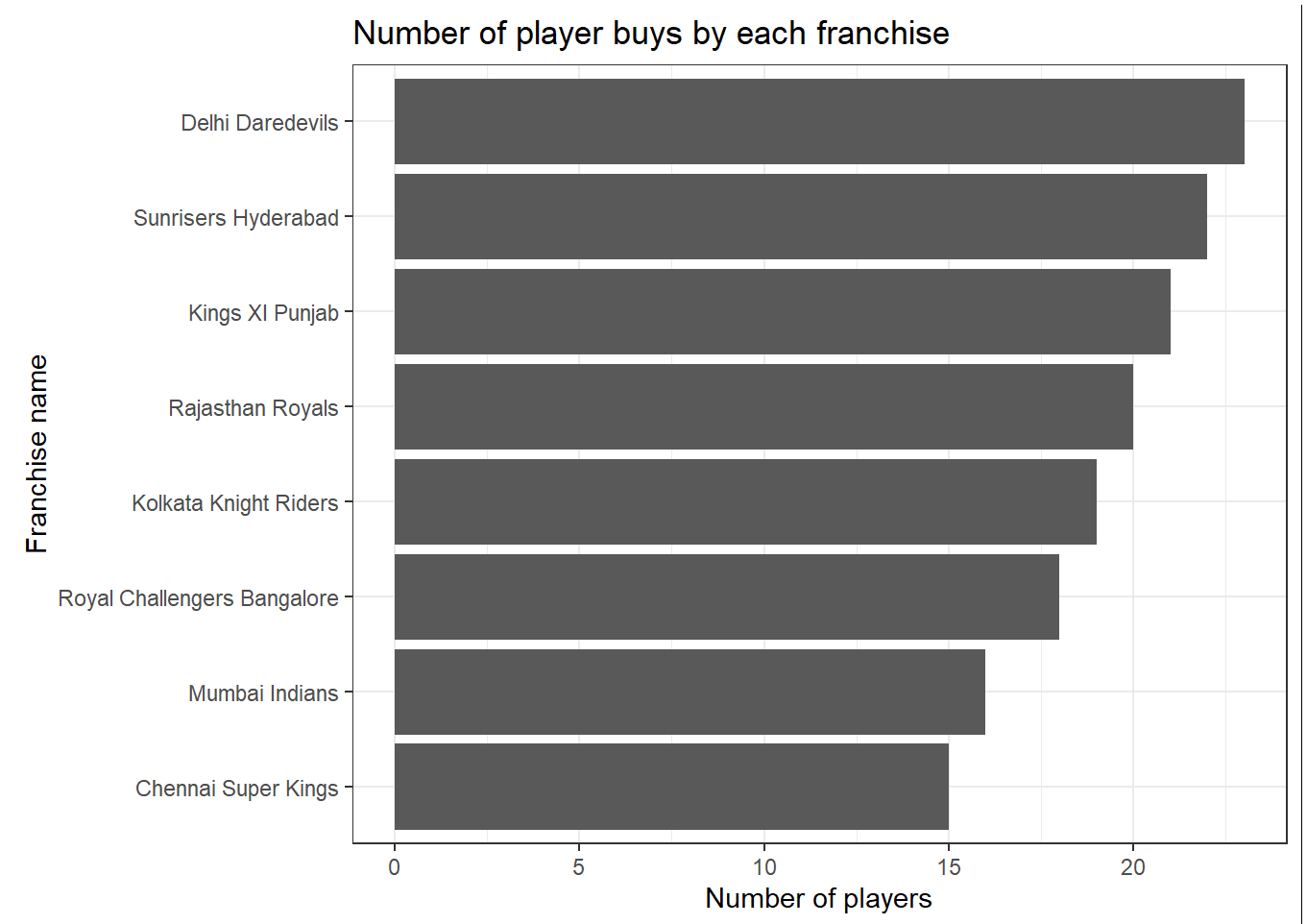
Overseas vs Indians players



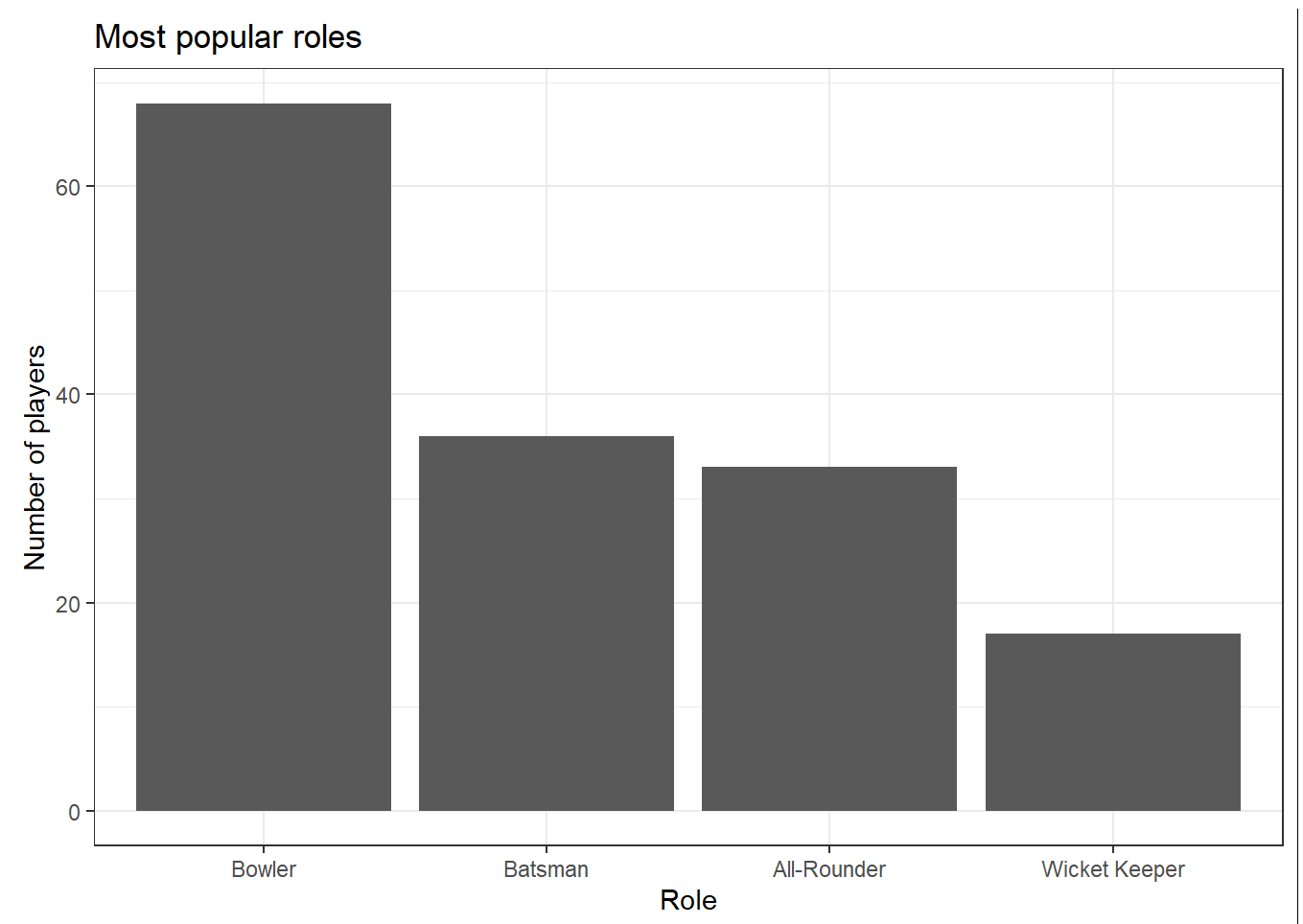
Franchises with amount spent-



Franchises with most buys-



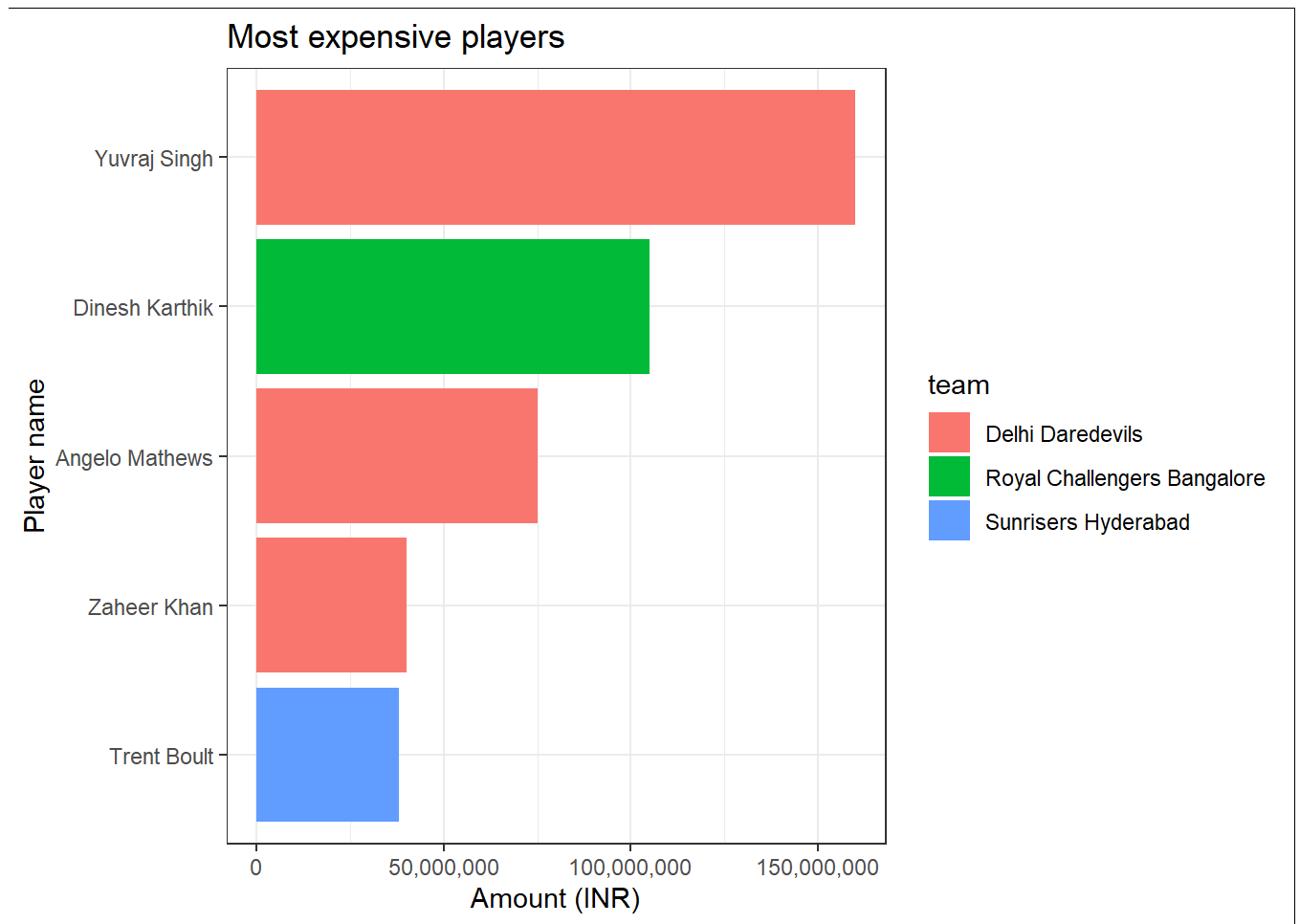
Most popular roles-



2015

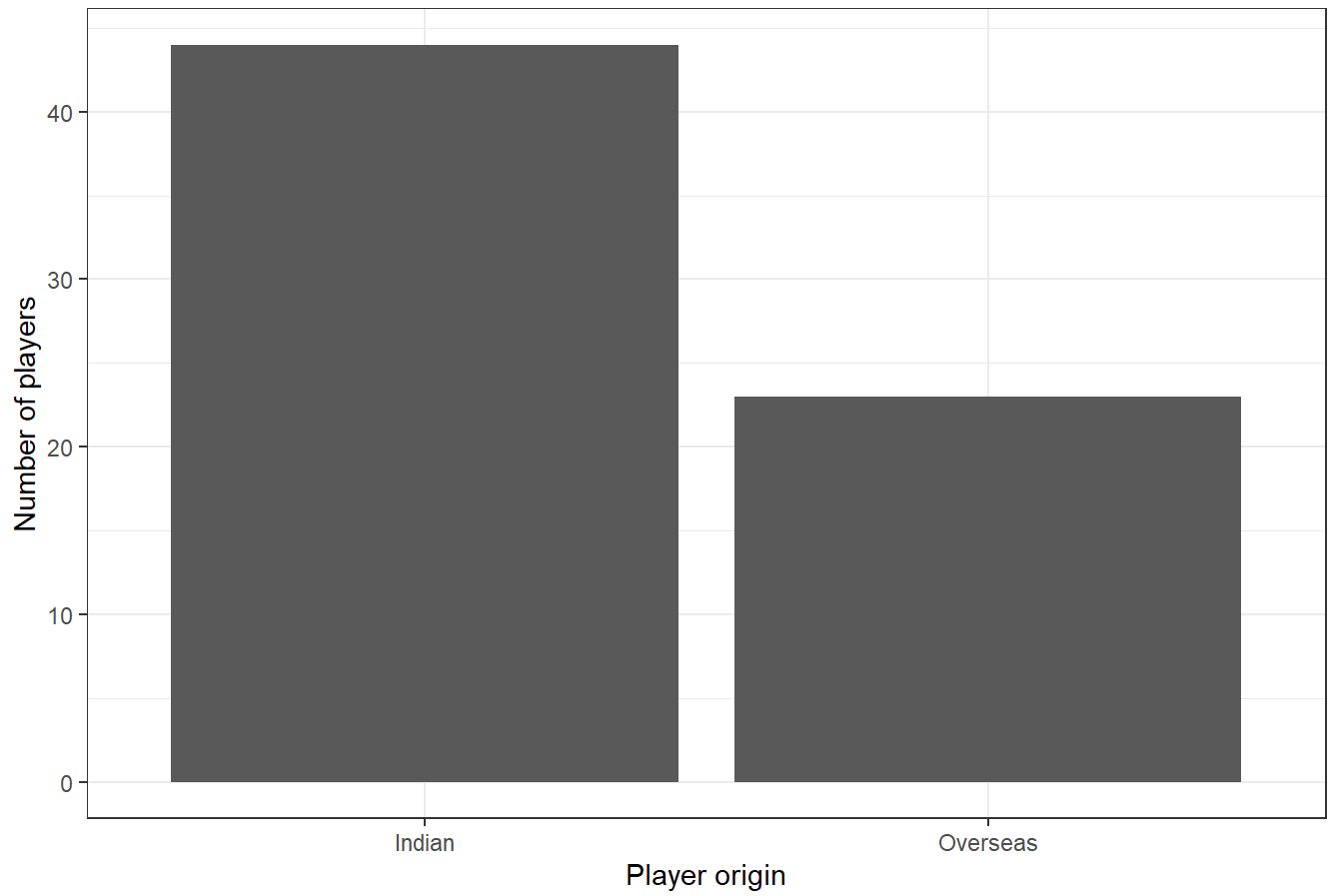
Most expensive players-

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

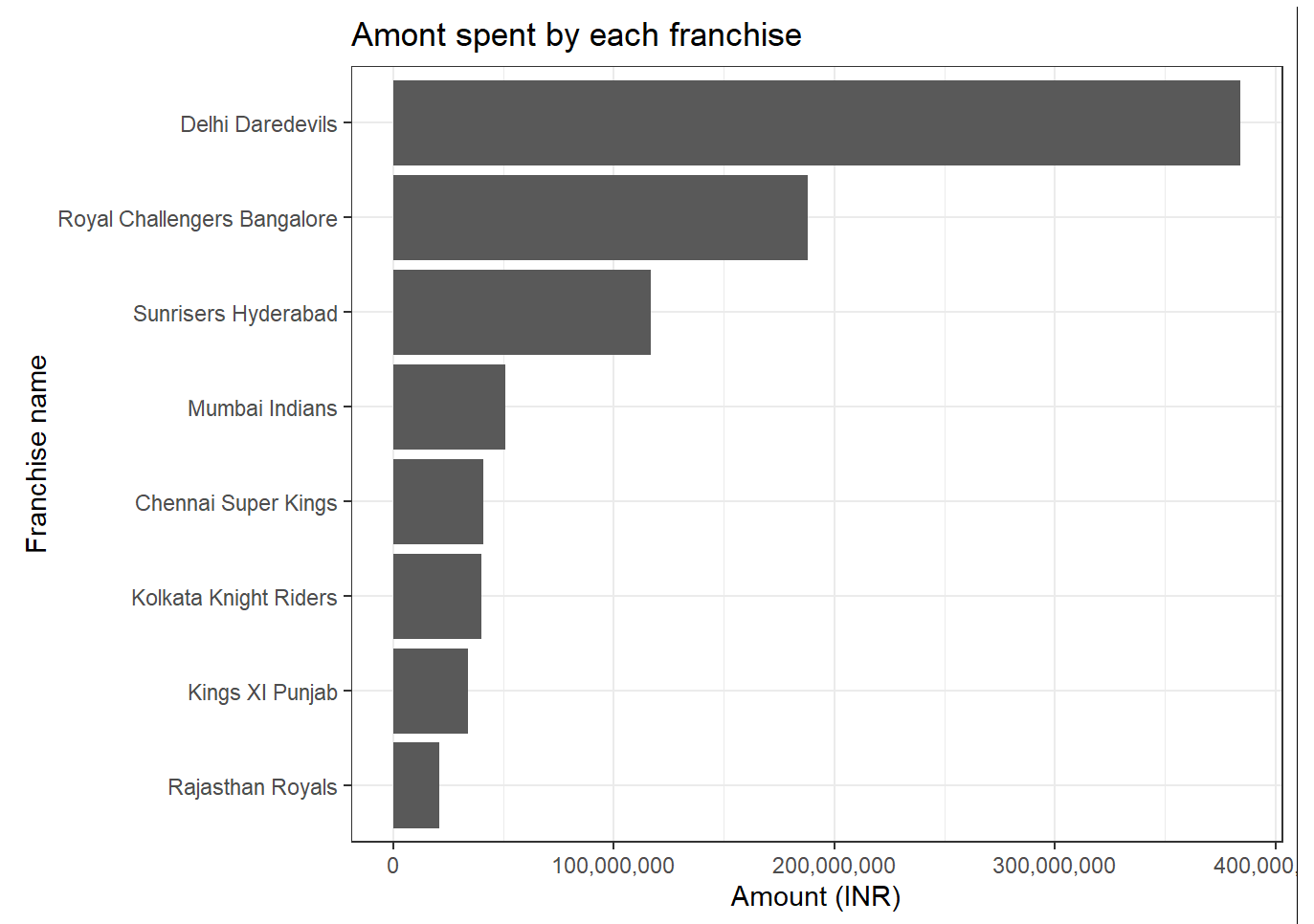


Overseas vs Indian players

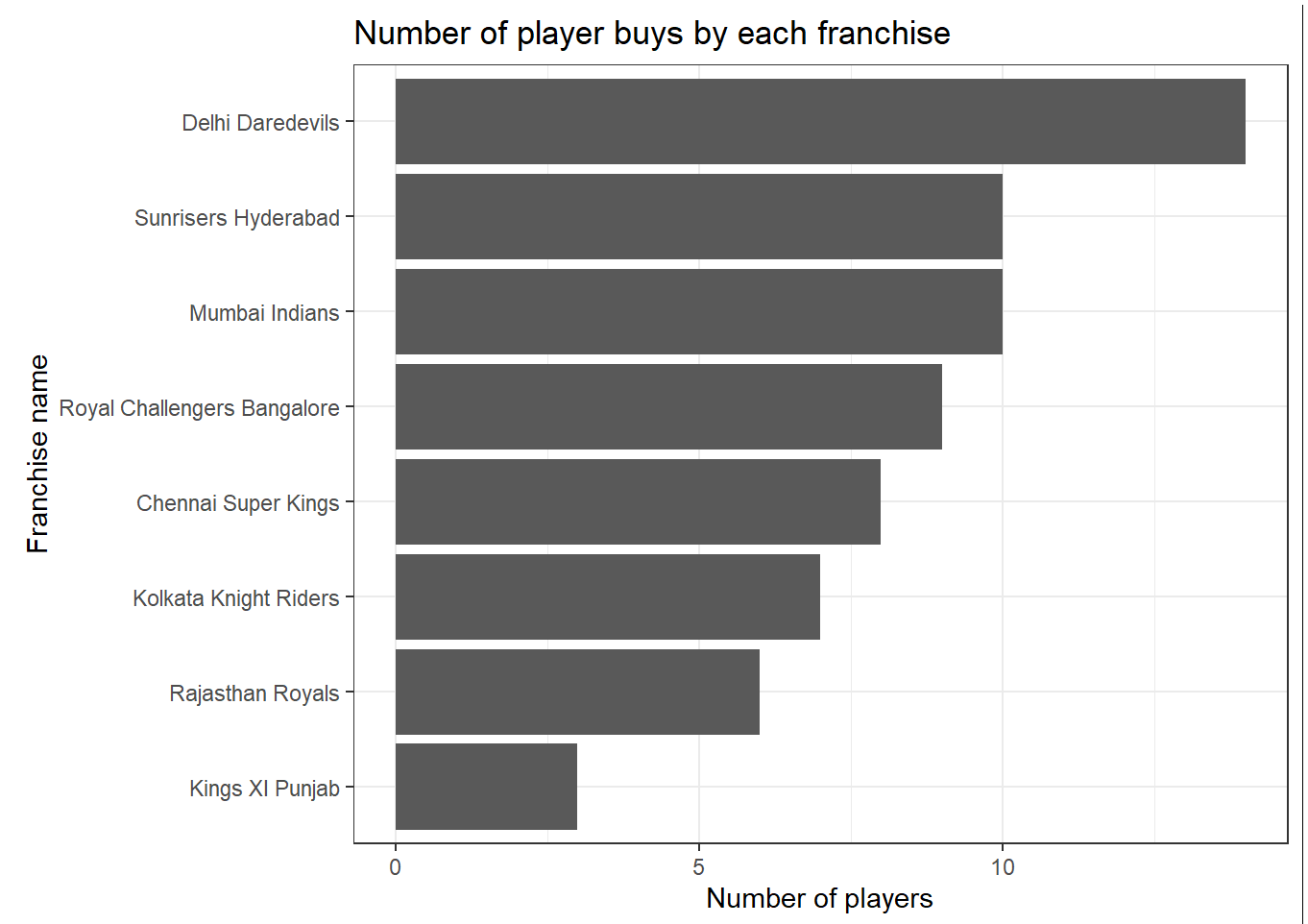
Overseas vs Indians players



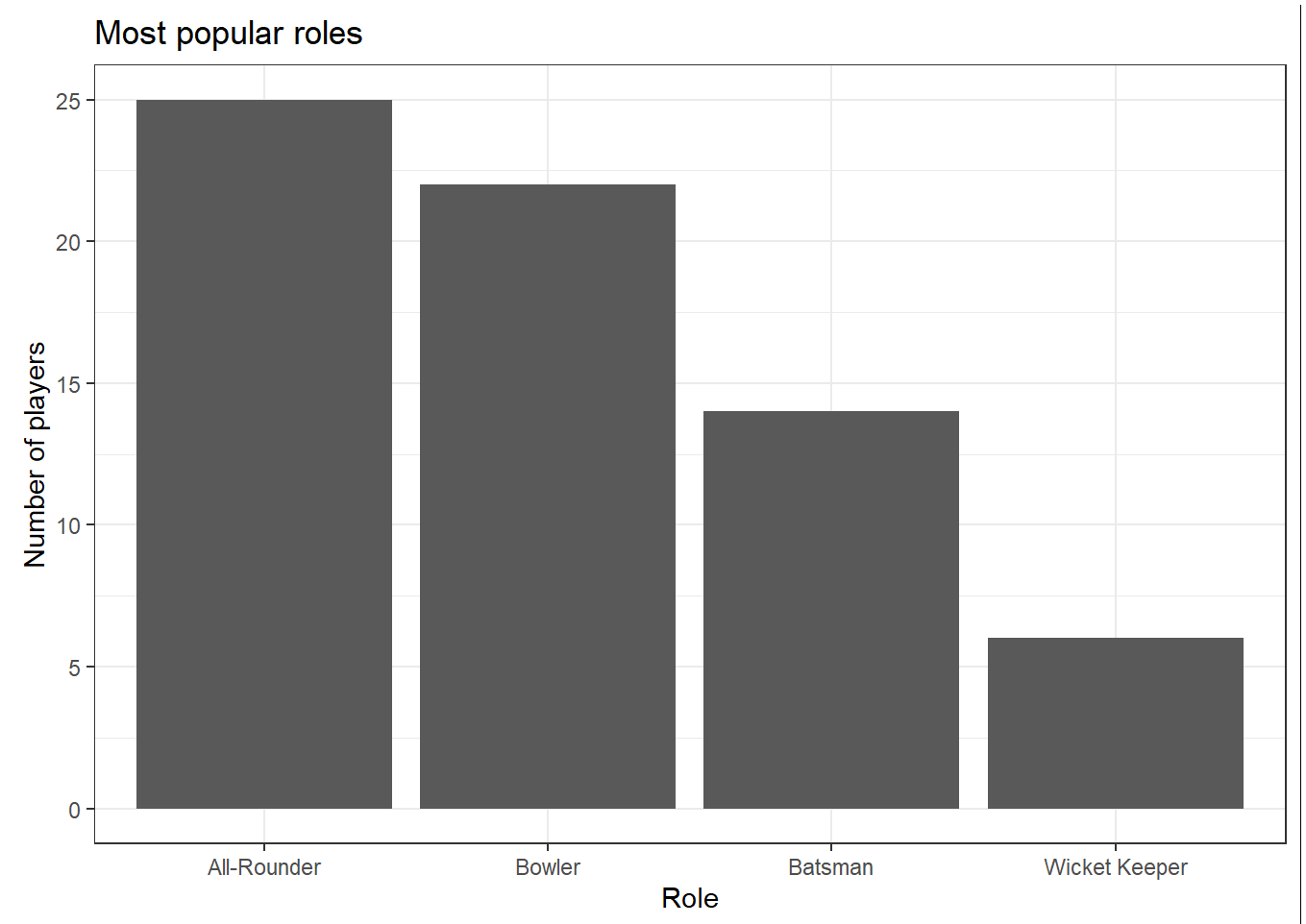
Franchises with amount spent-



Franchises with most buys-



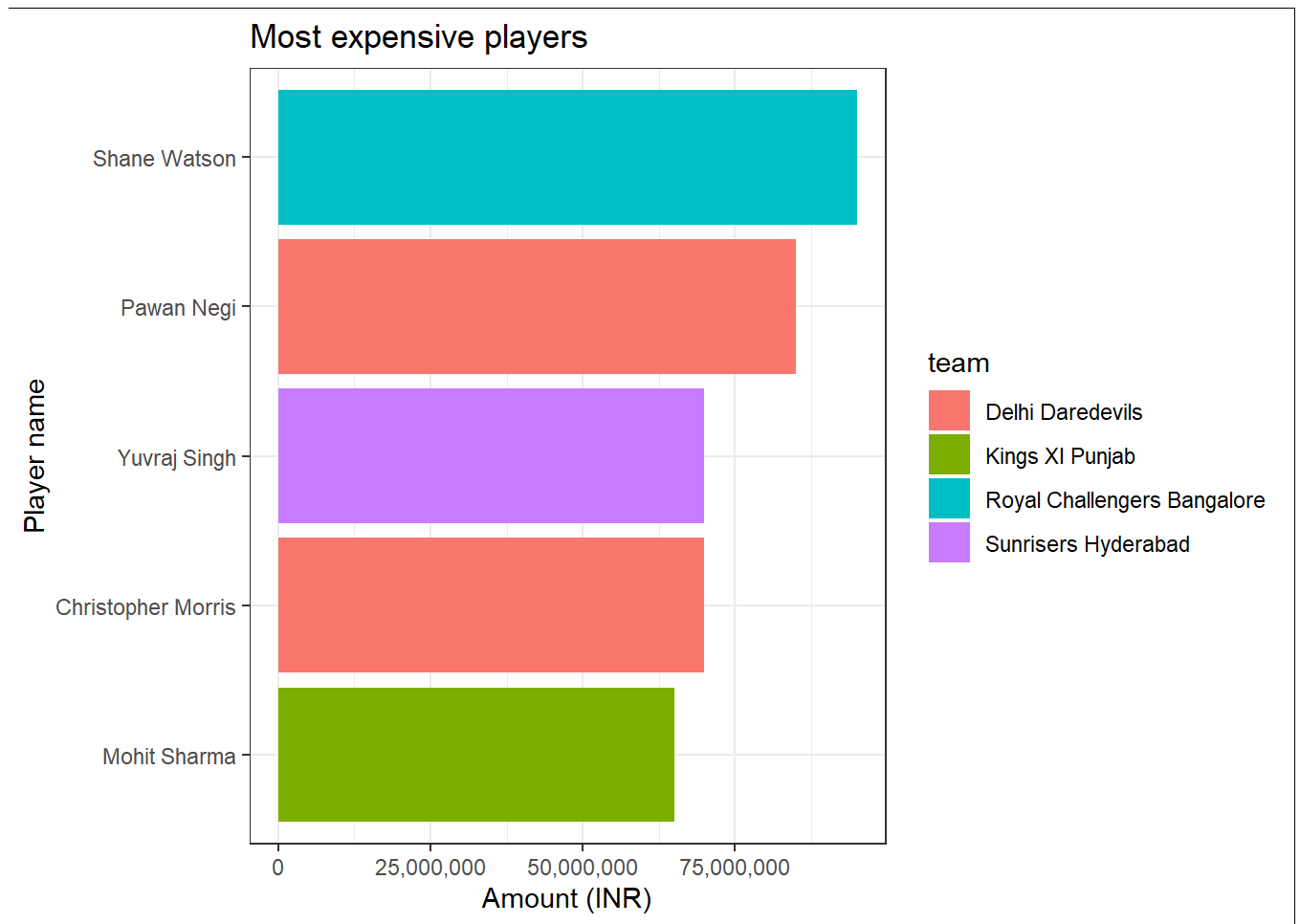
Most popular roles-



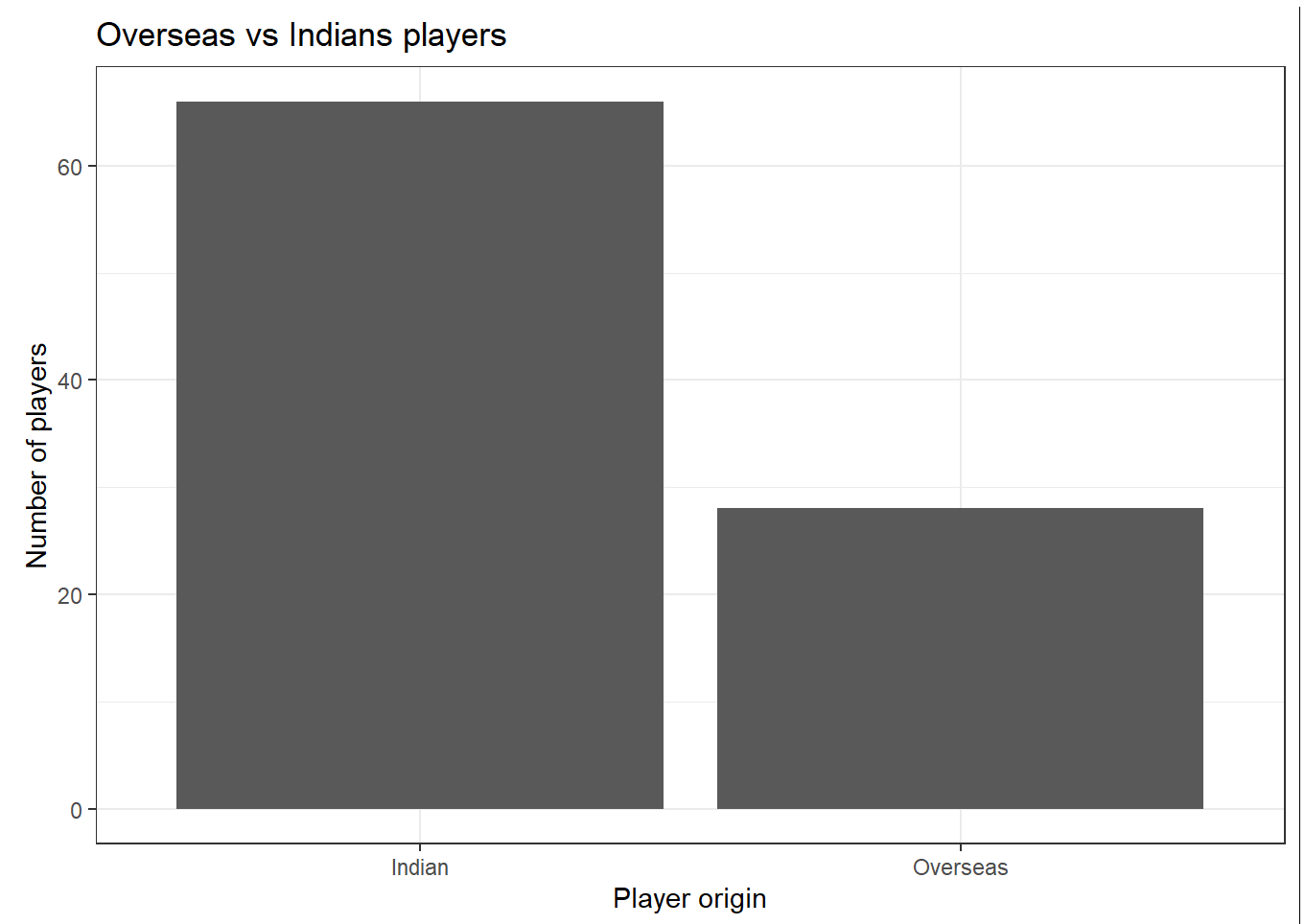
2016

Most expensive players-

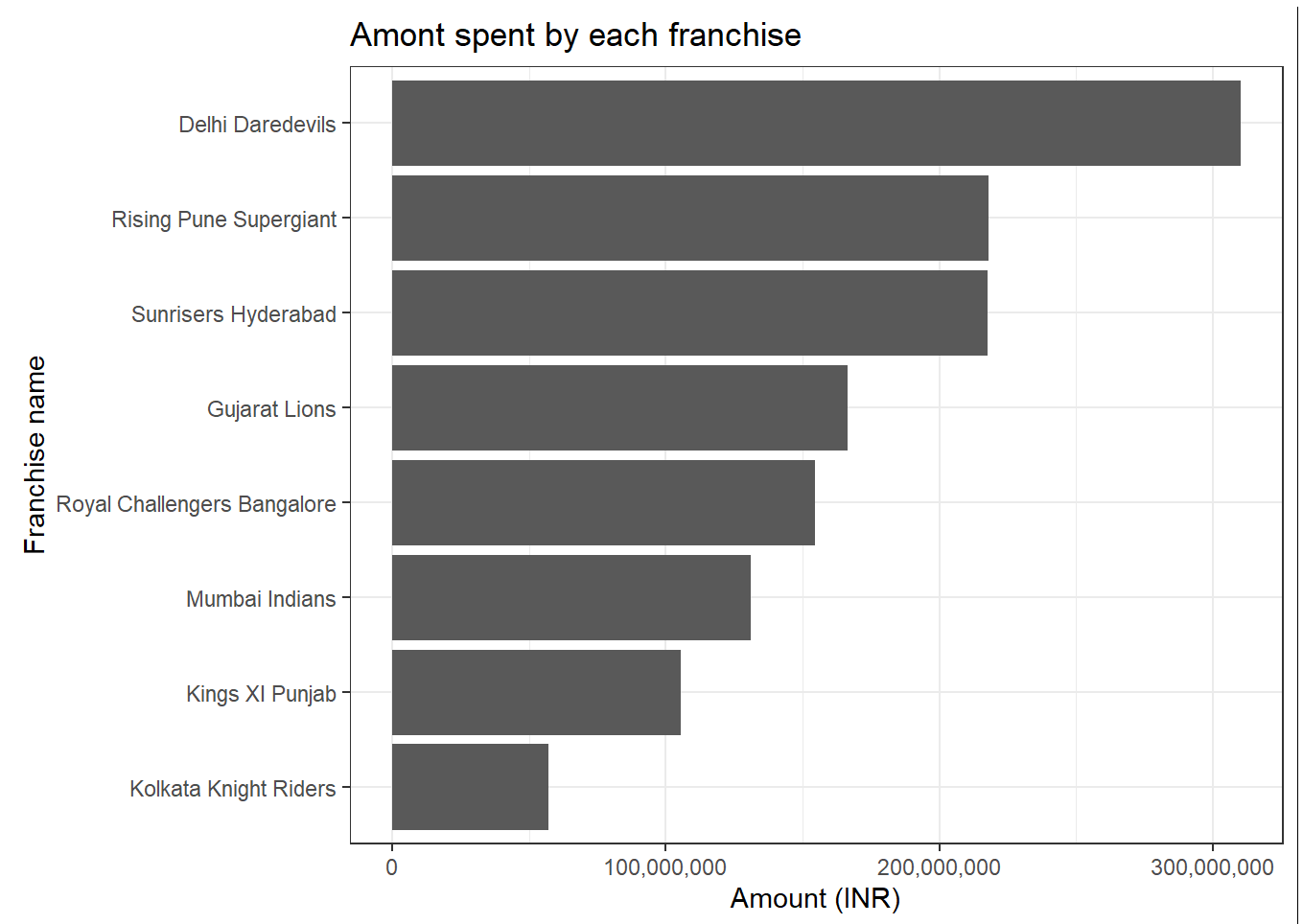
`summarise()` has grouped output by 'player', 'amount'. You can override using the `.groups` argument.



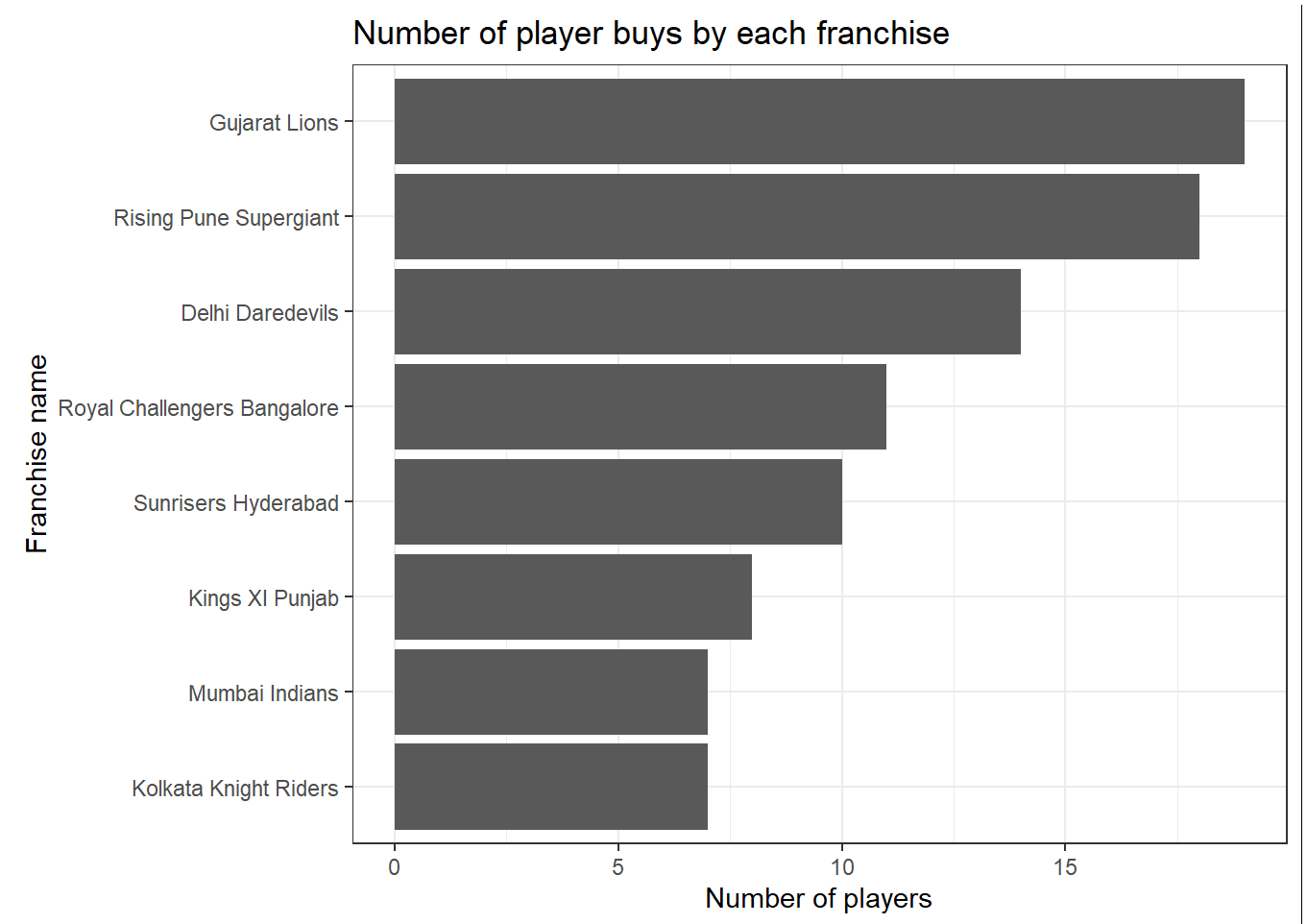
Overseas vs Indian players-



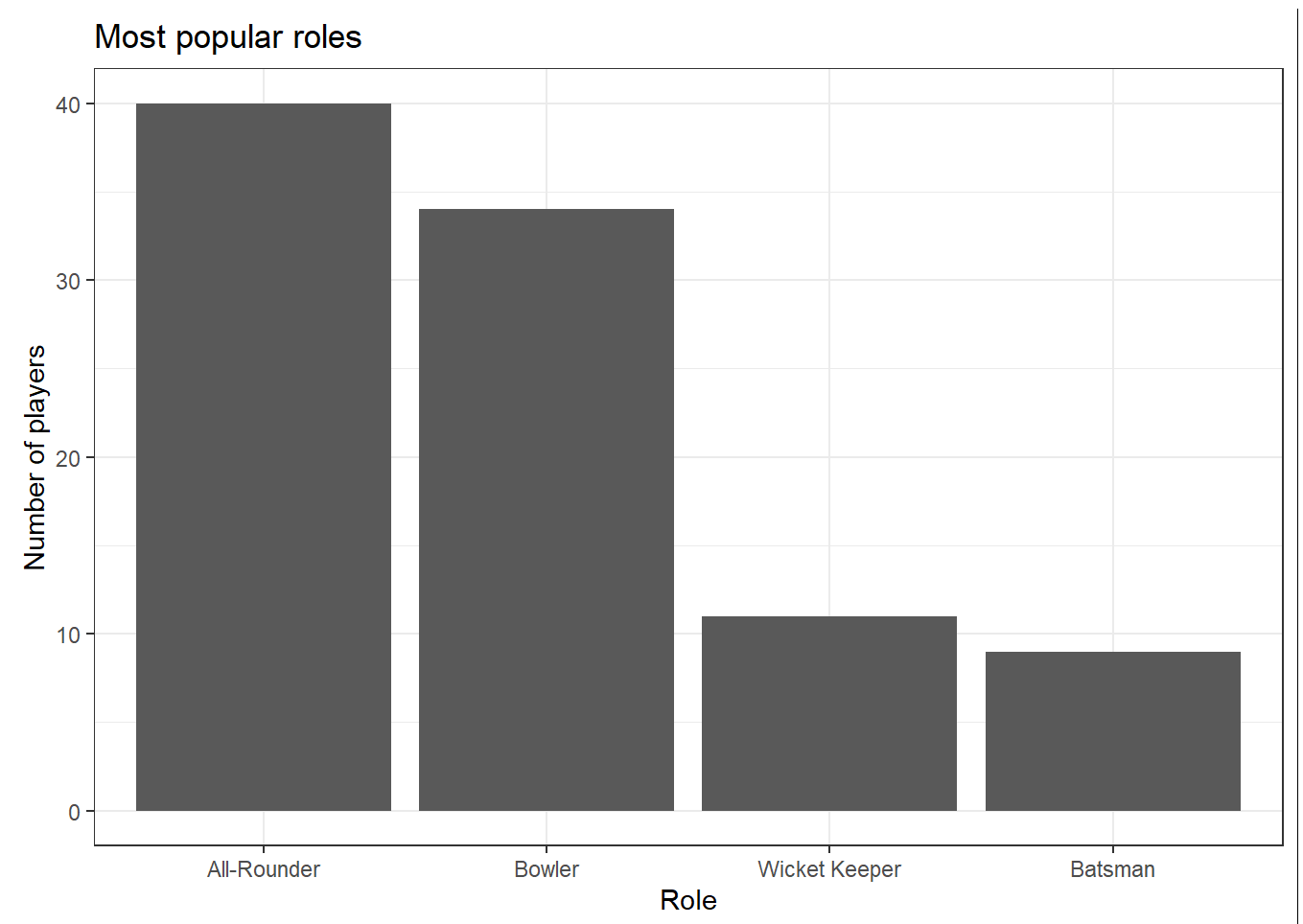
Franchises with amount spent-



Franchises with most buys-



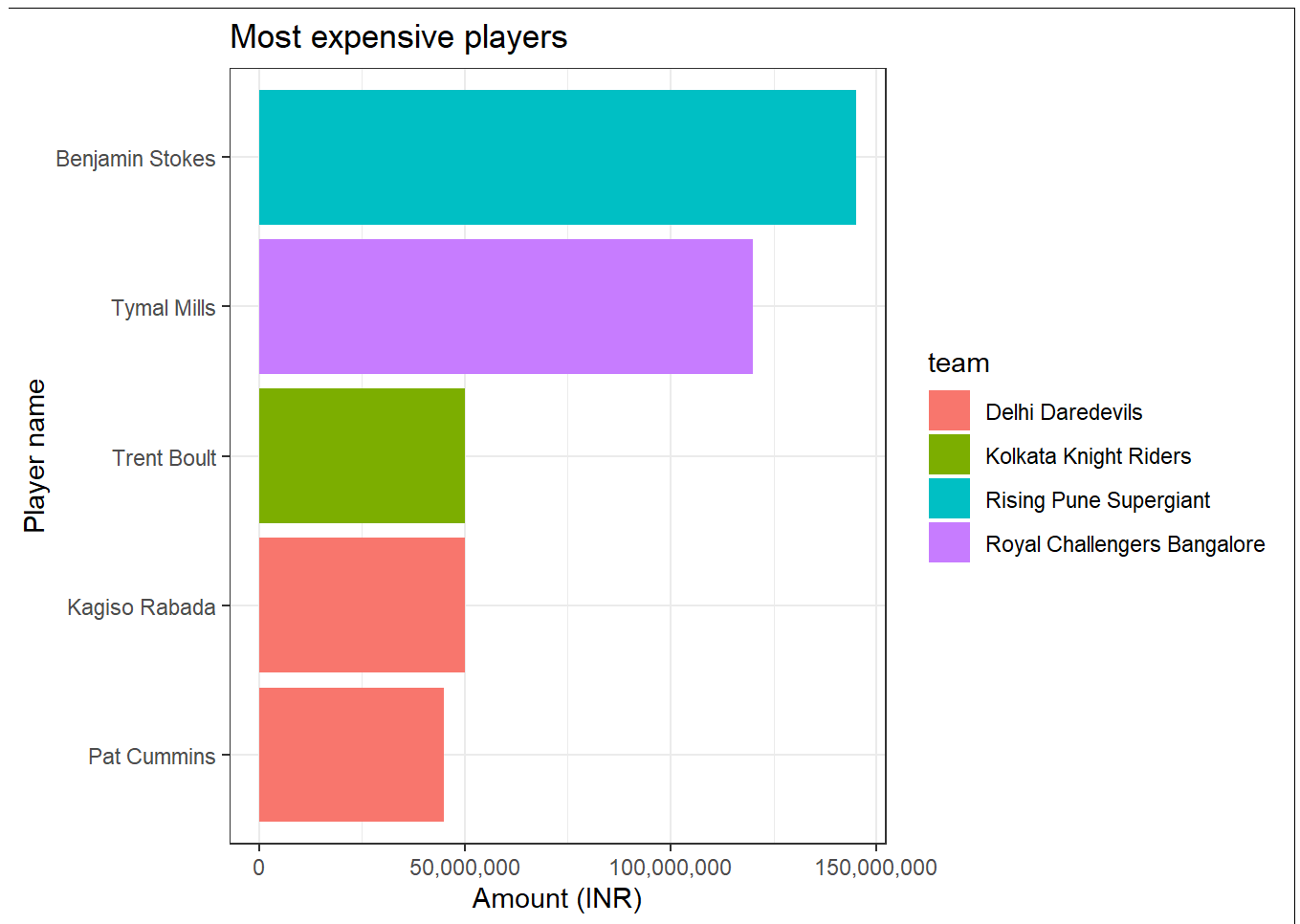
Most popular roles-



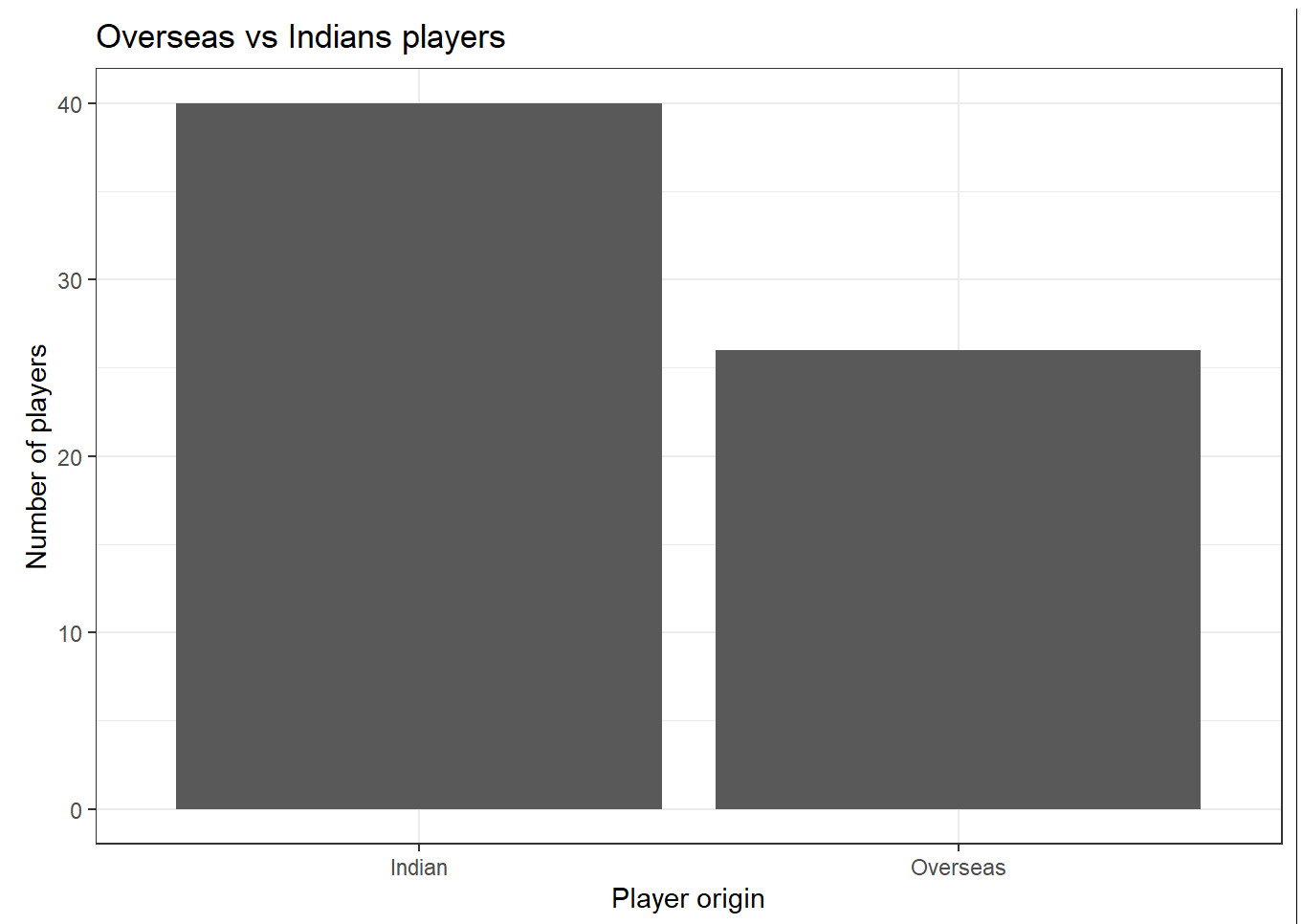
2017

Most expensive Players-

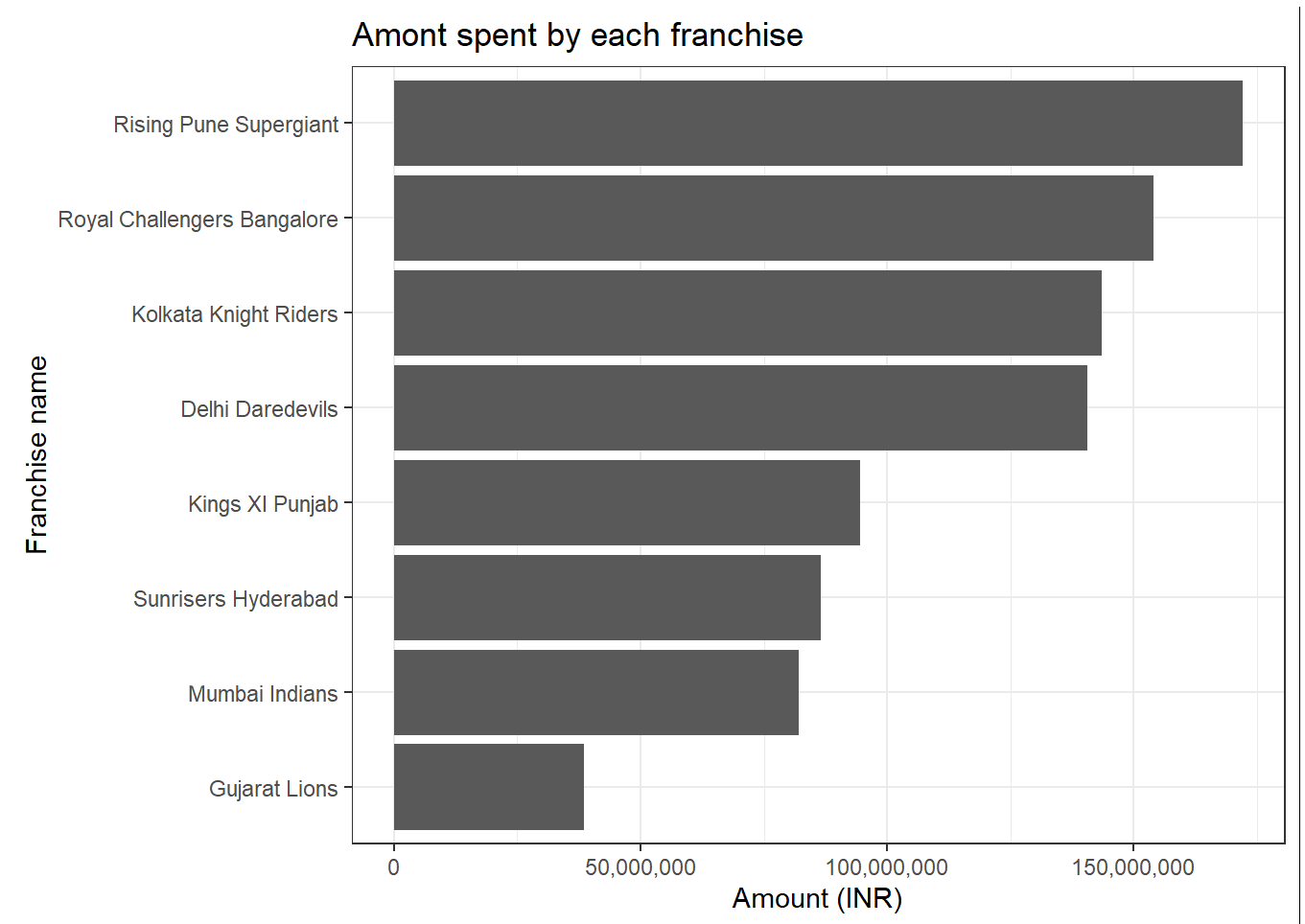
``summarise()`` has grouped output by 'player'. You can override using the ``.groups`` argument.



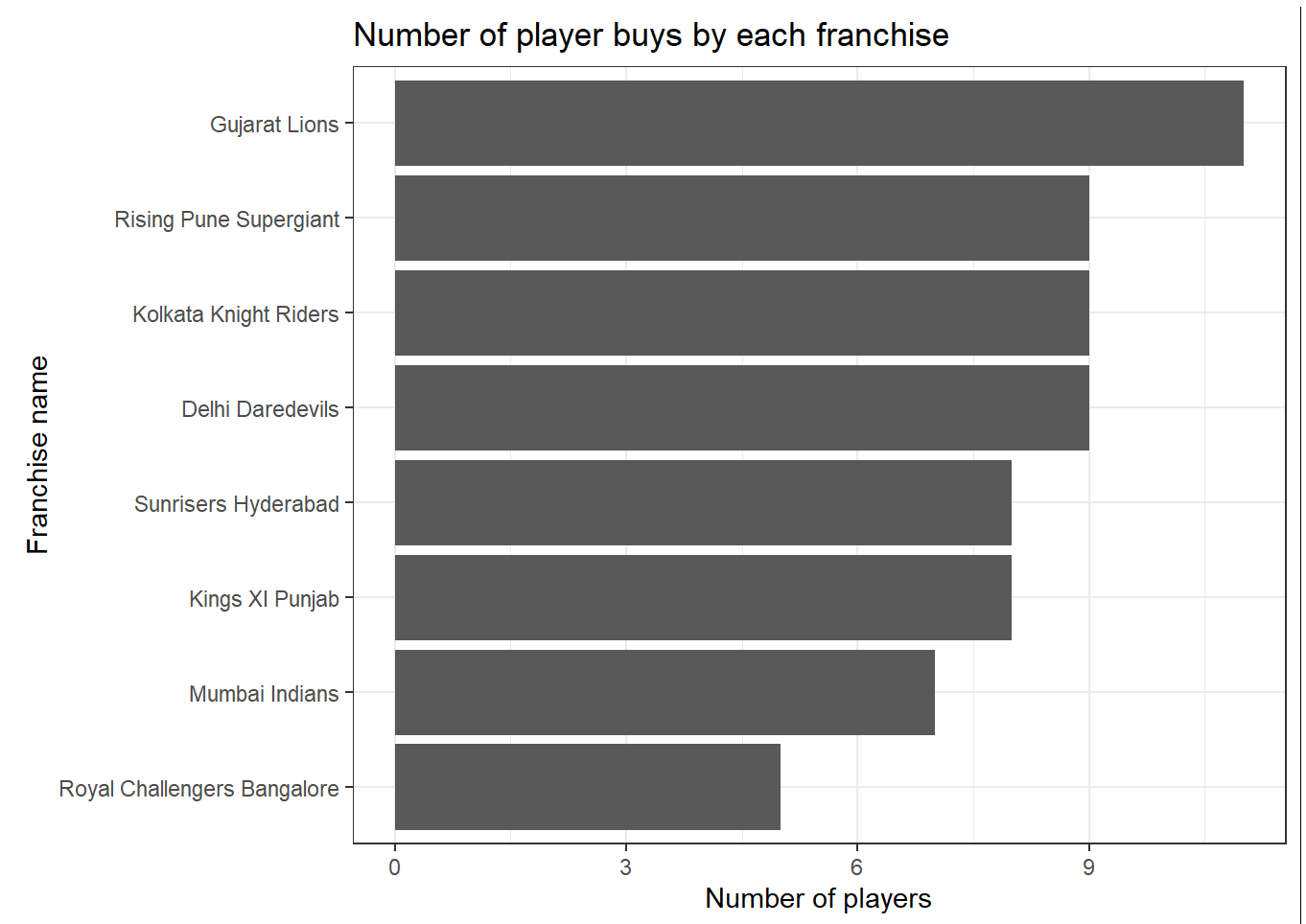
Overseas vs Indian players-



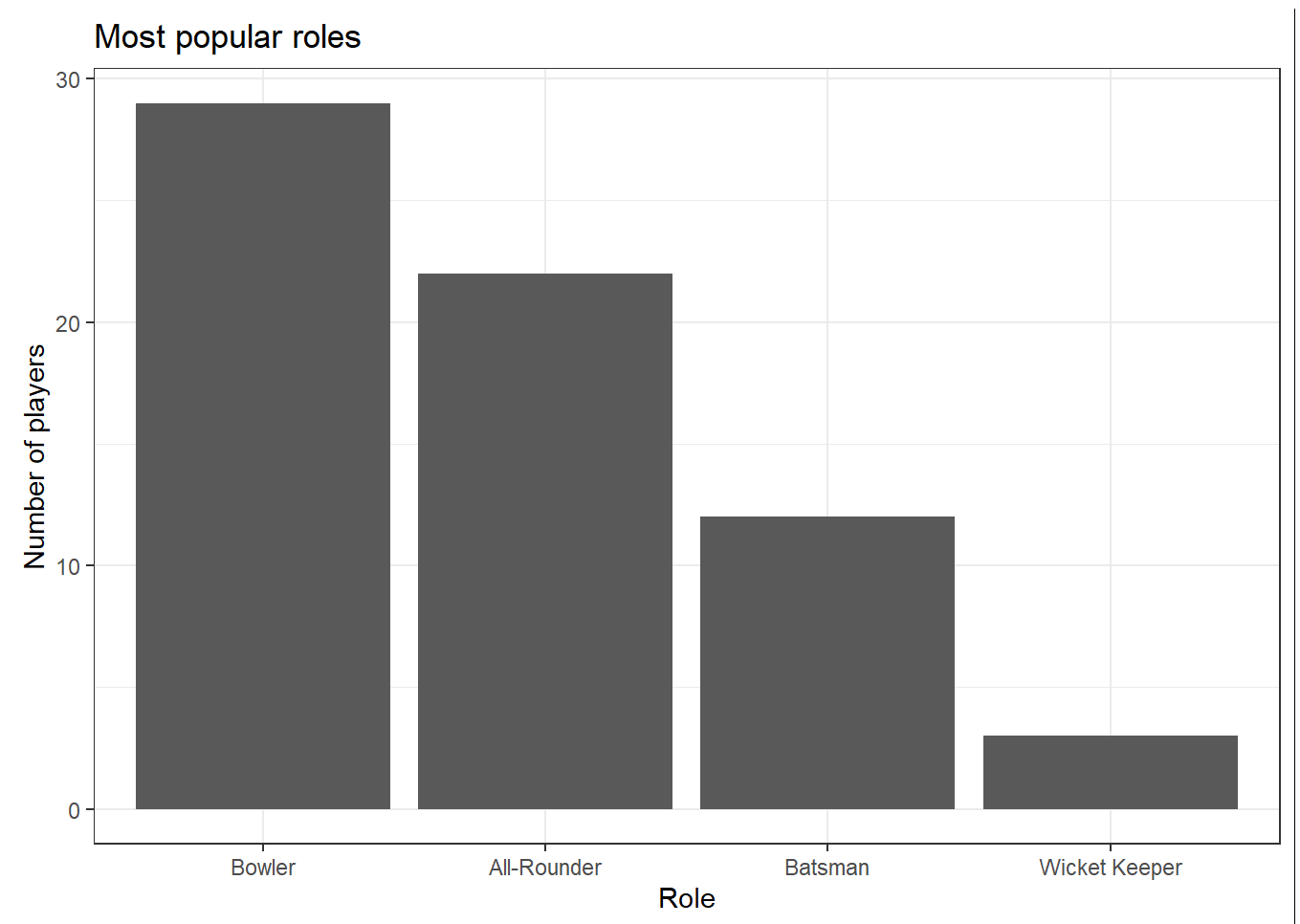
Franchises with amount spent-



Franchises with most buys-



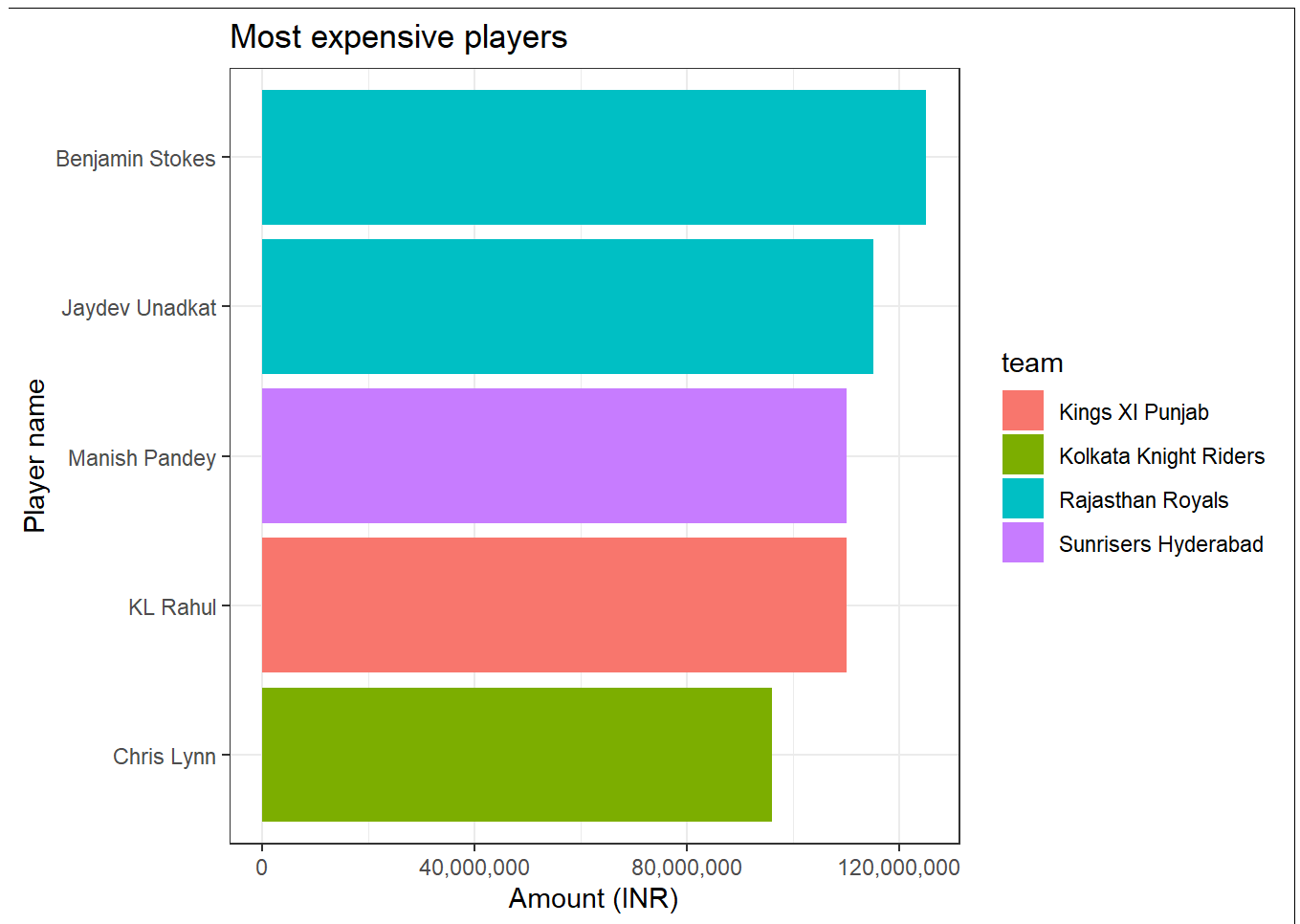
Most popular roles-



2018

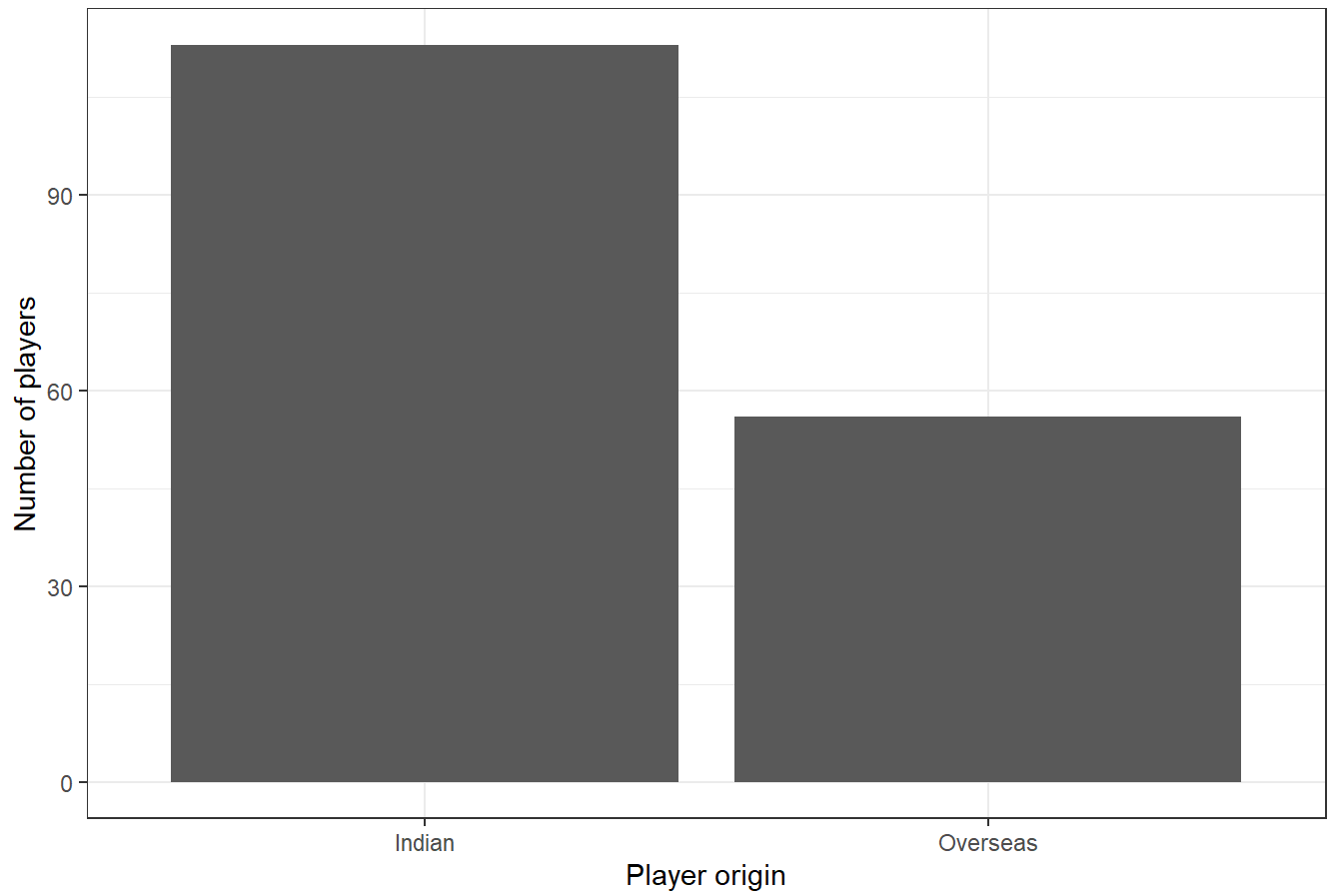
Most expensive players-

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

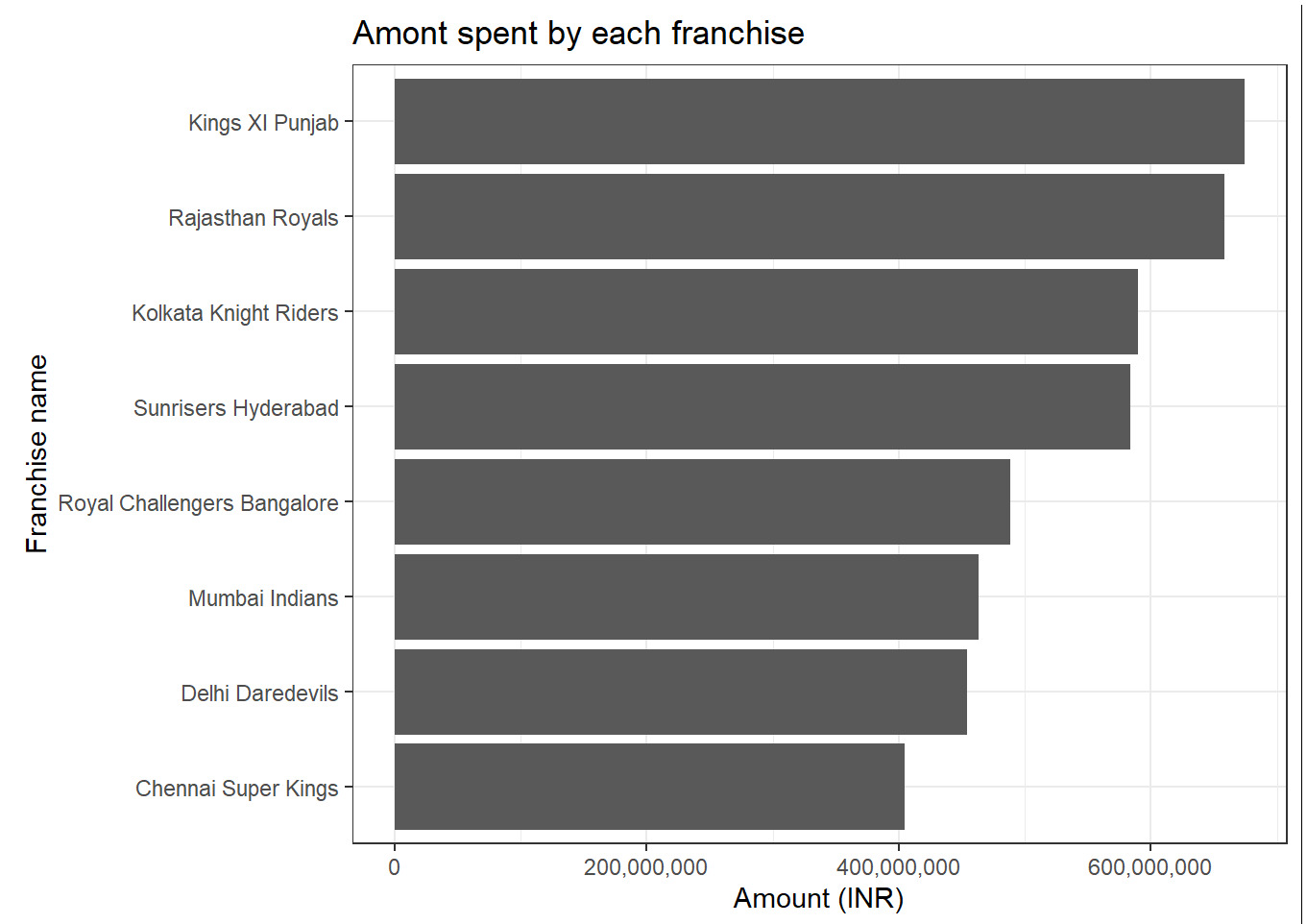


Overseas vs Indian players-

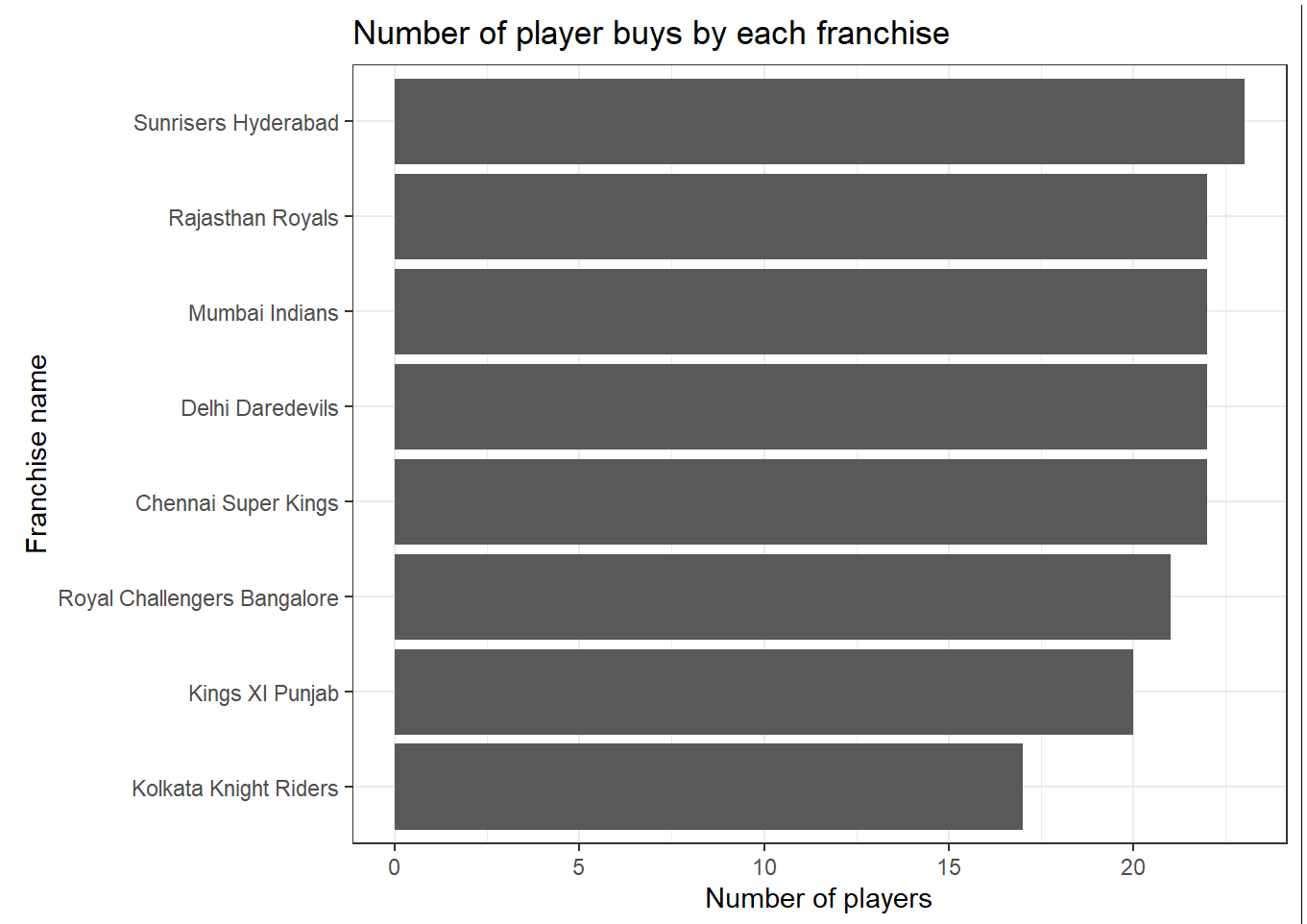
Overseas vs Indians players



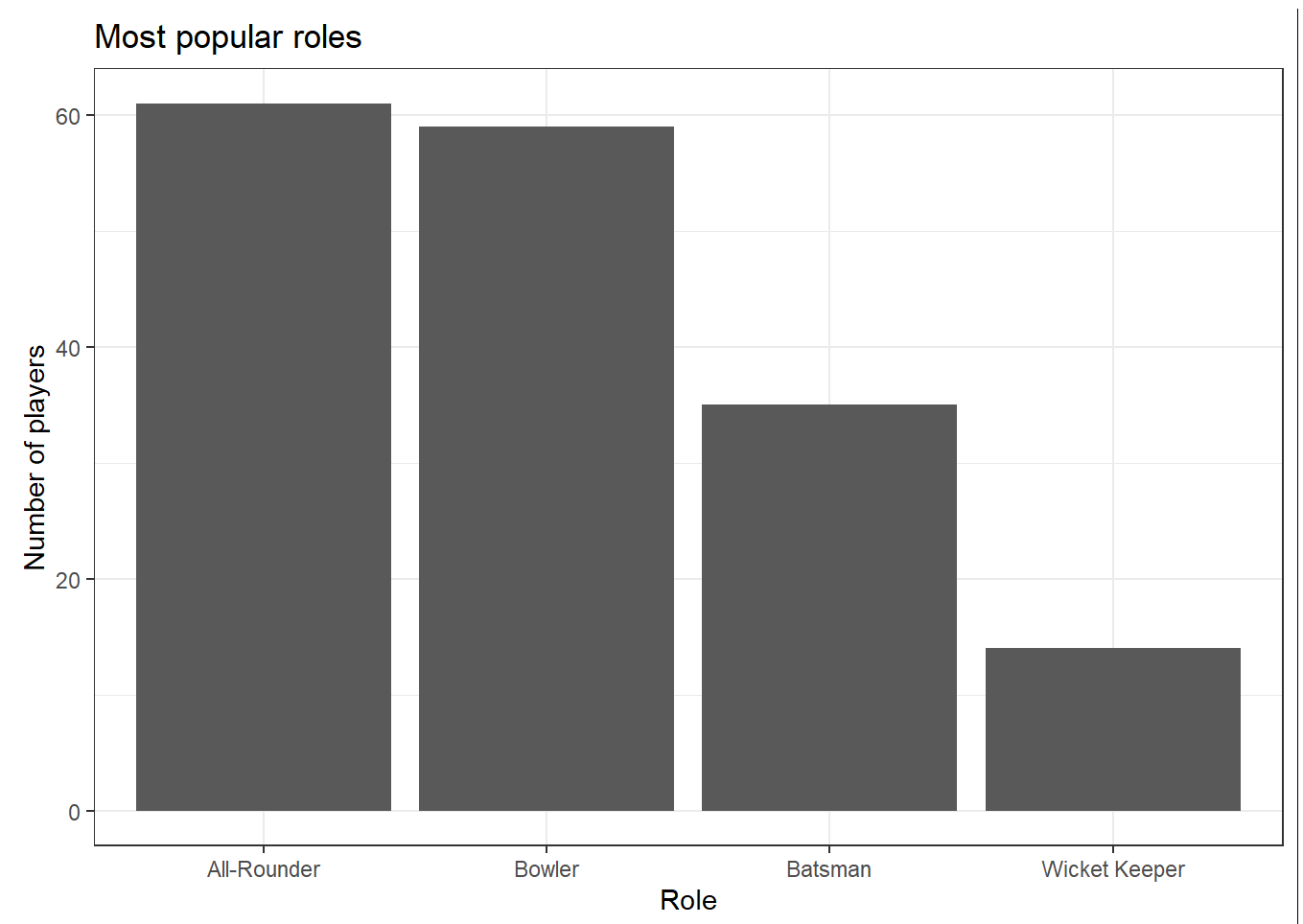
Amount spent by each franchise-



Franchise with most buys-



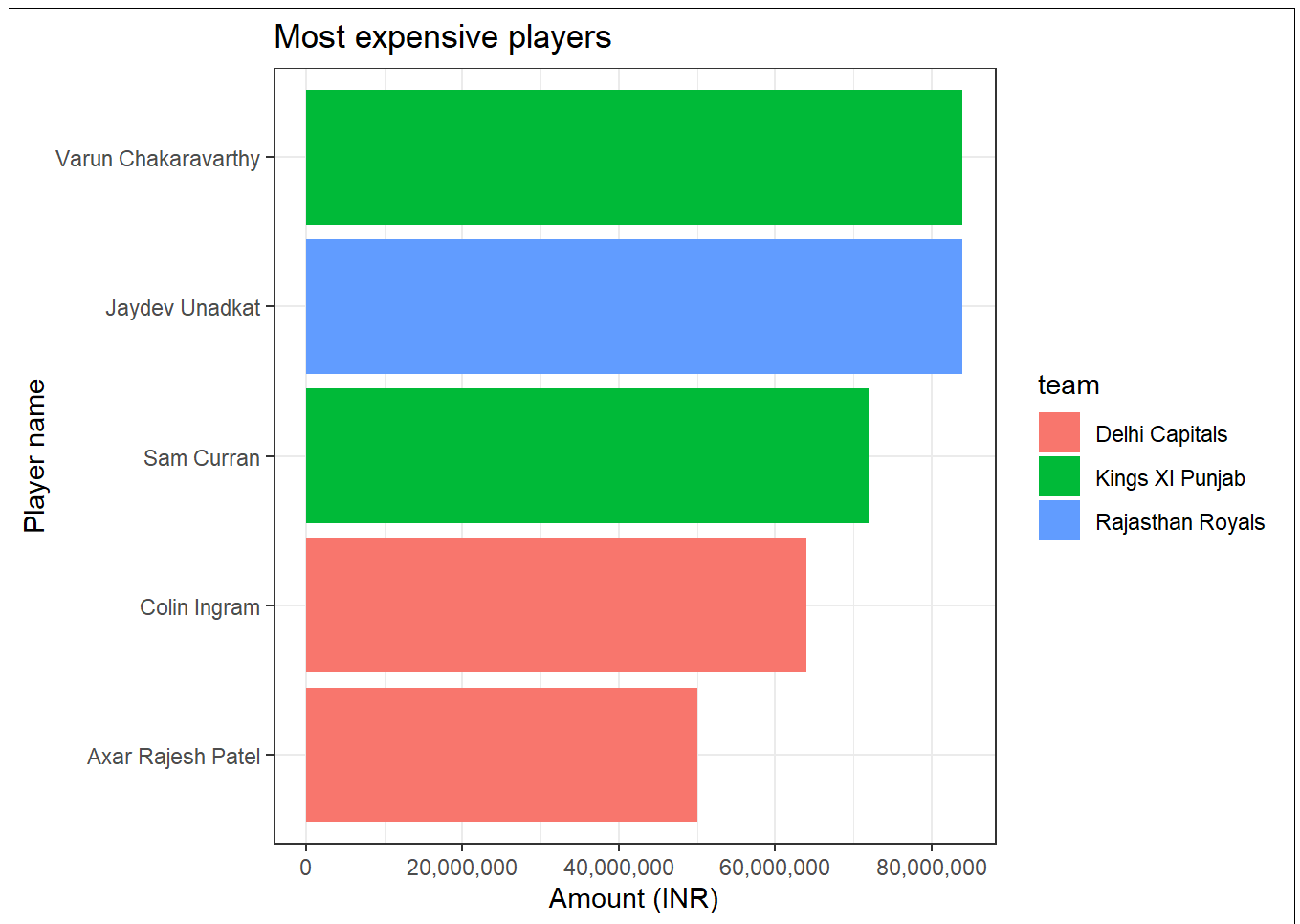
Most popular roles-



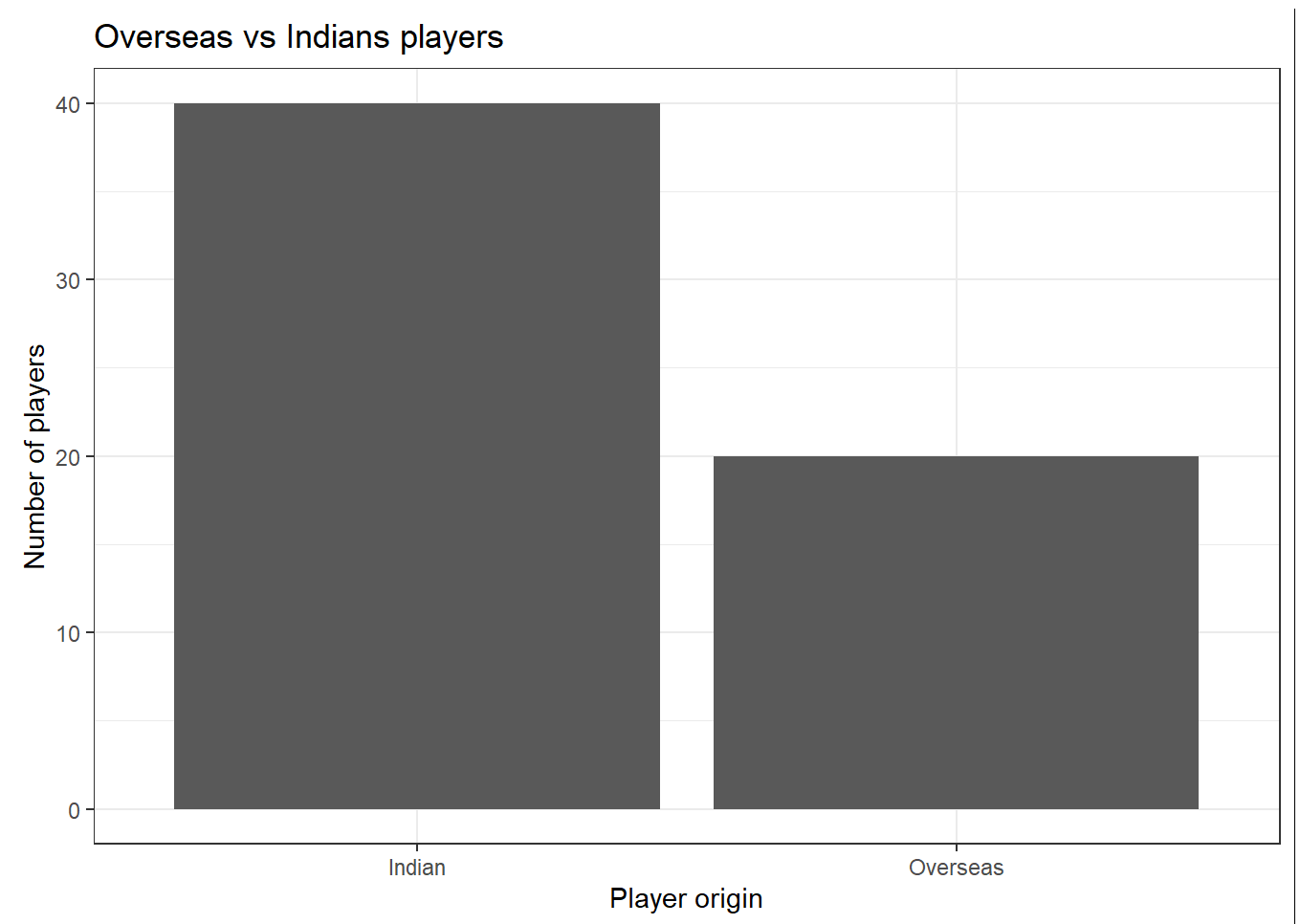
2019

Most expensive players-

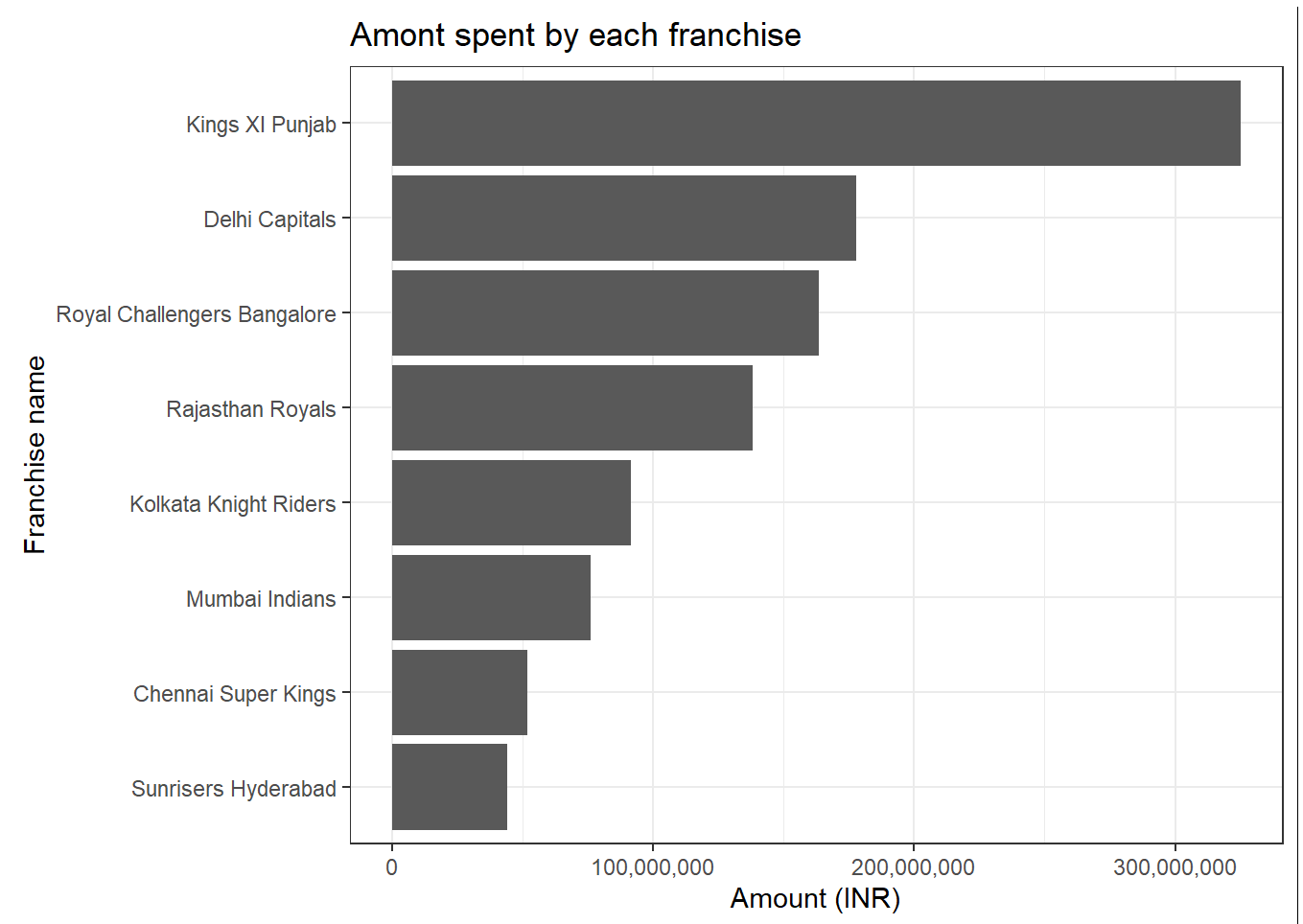
``summarise()`` has grouped output by 'player'. You can override using the ``.groups`` argument.



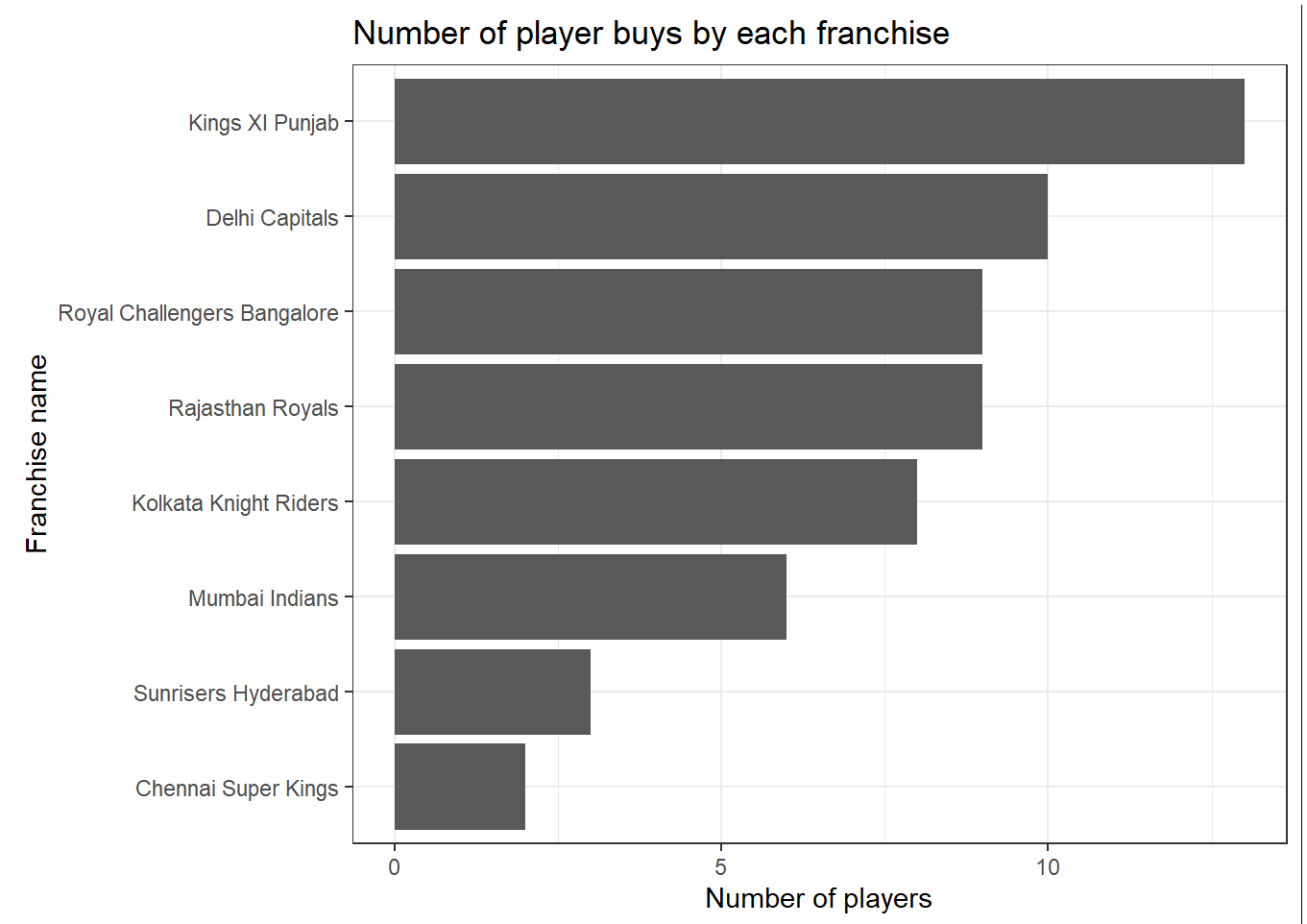
Overseas vs Indian players-



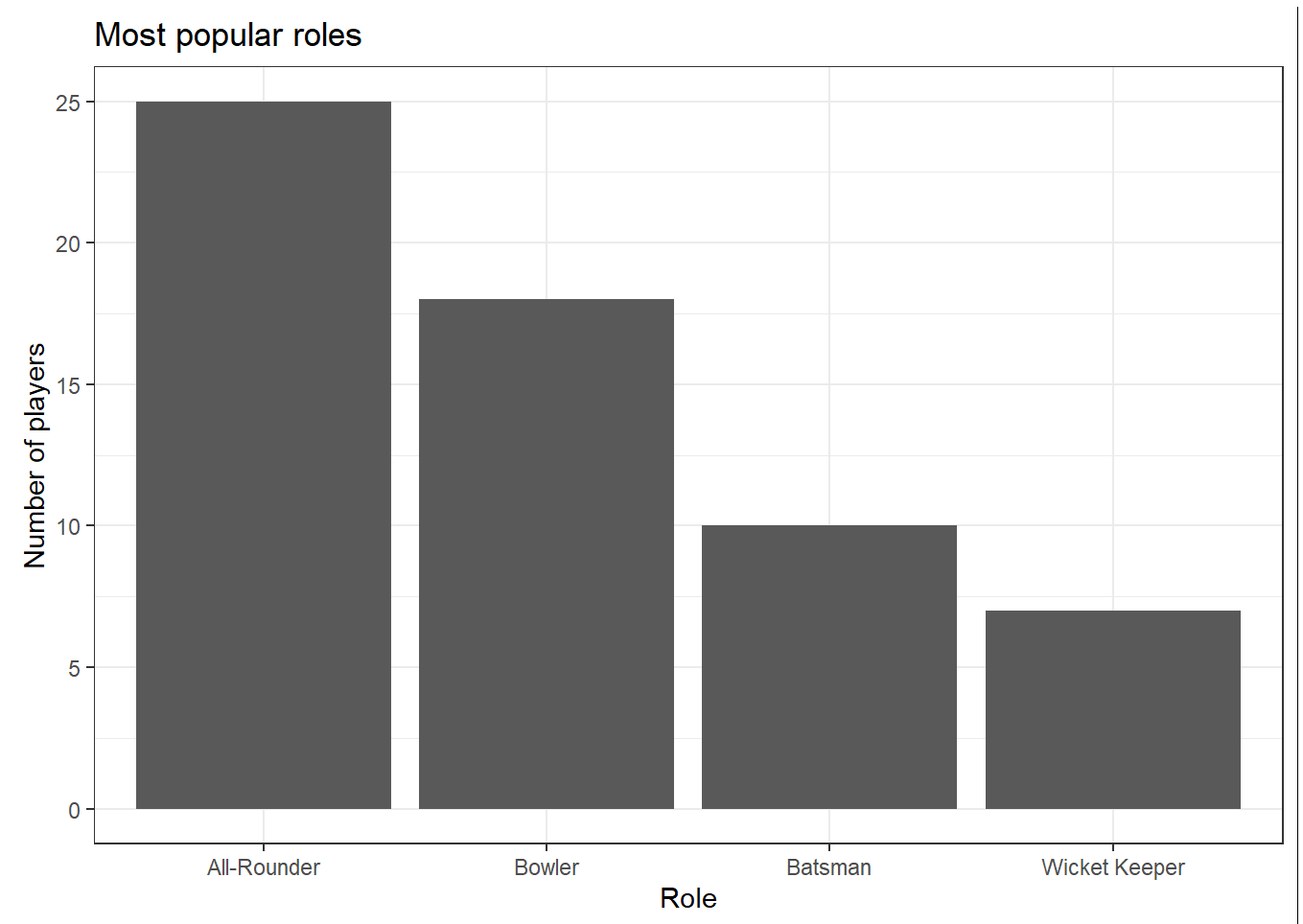
Amount spent by each franchise-



Franchises with most buys-



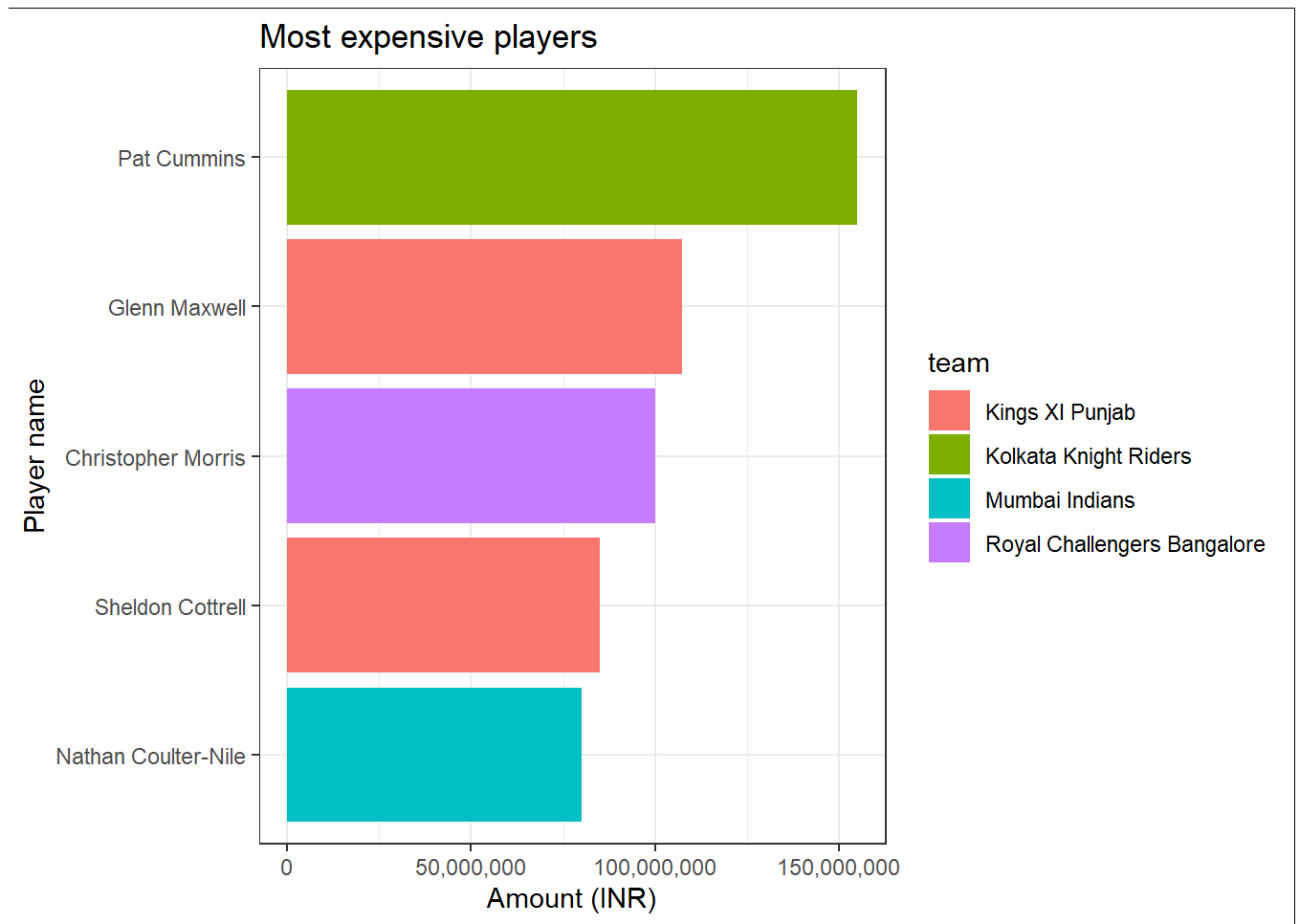
Most popular roles-



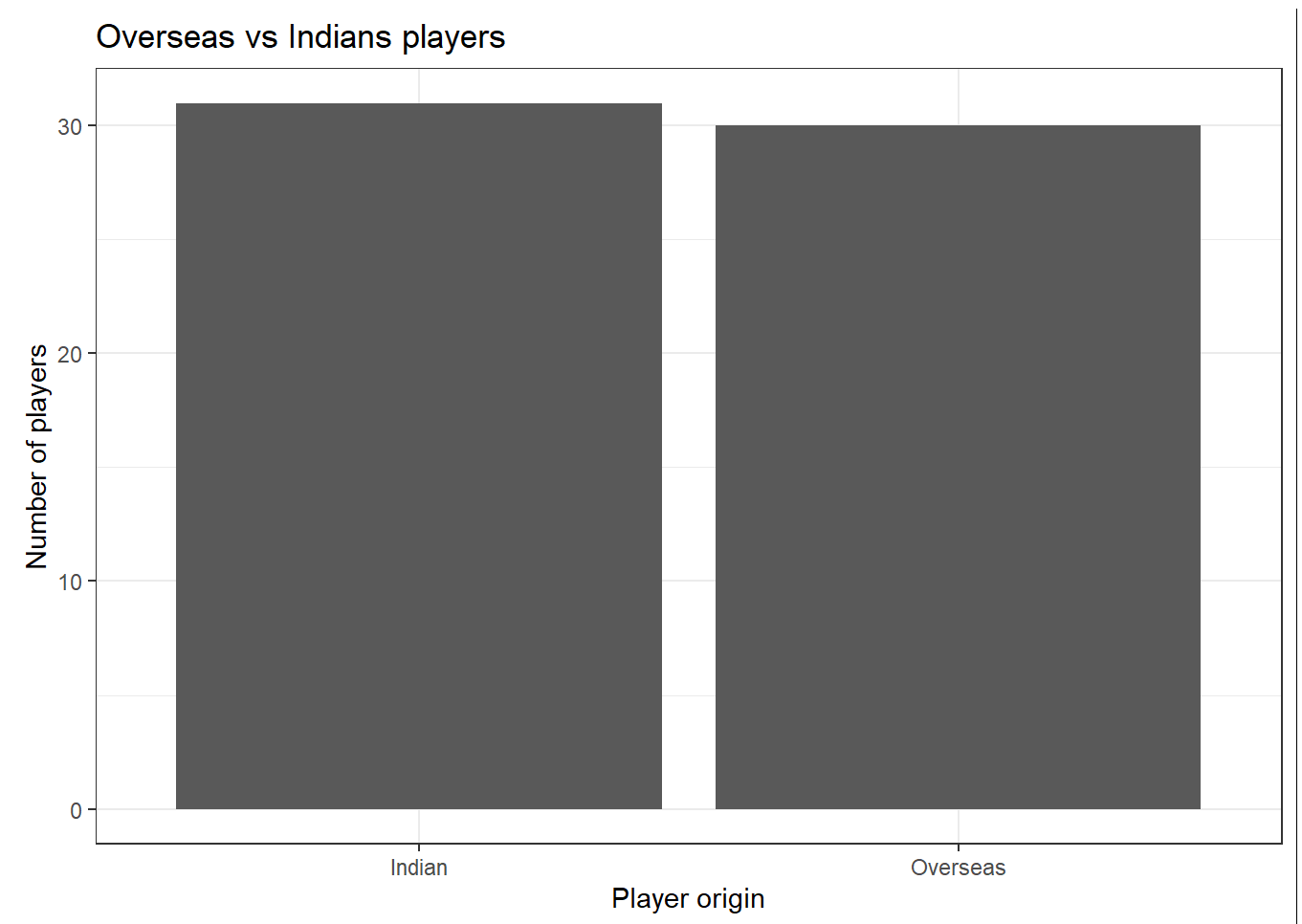
2020

Most expensive players-

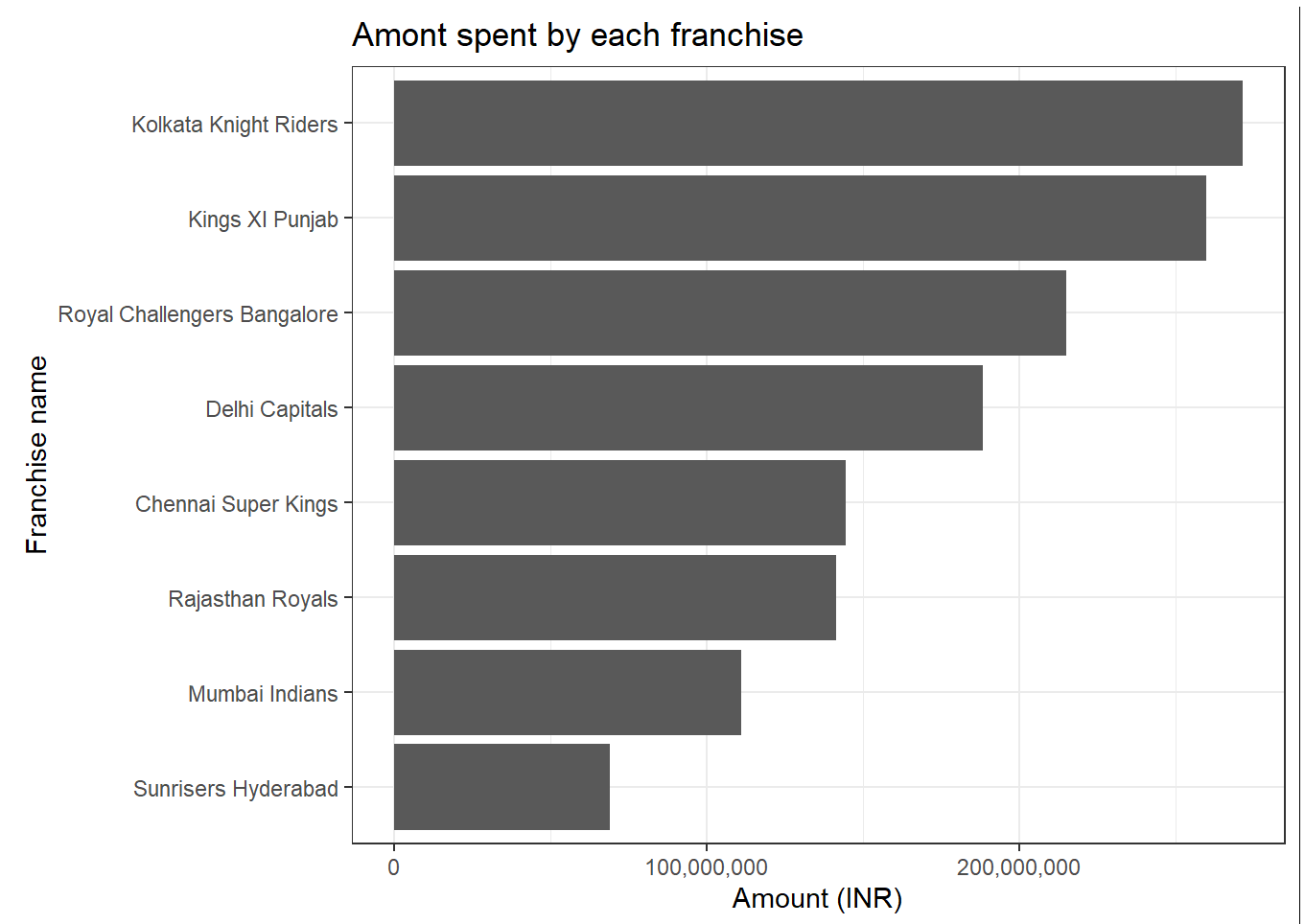
``summarise()`` has grouped output by 'player'. You can override using the ``.groups`` argument.



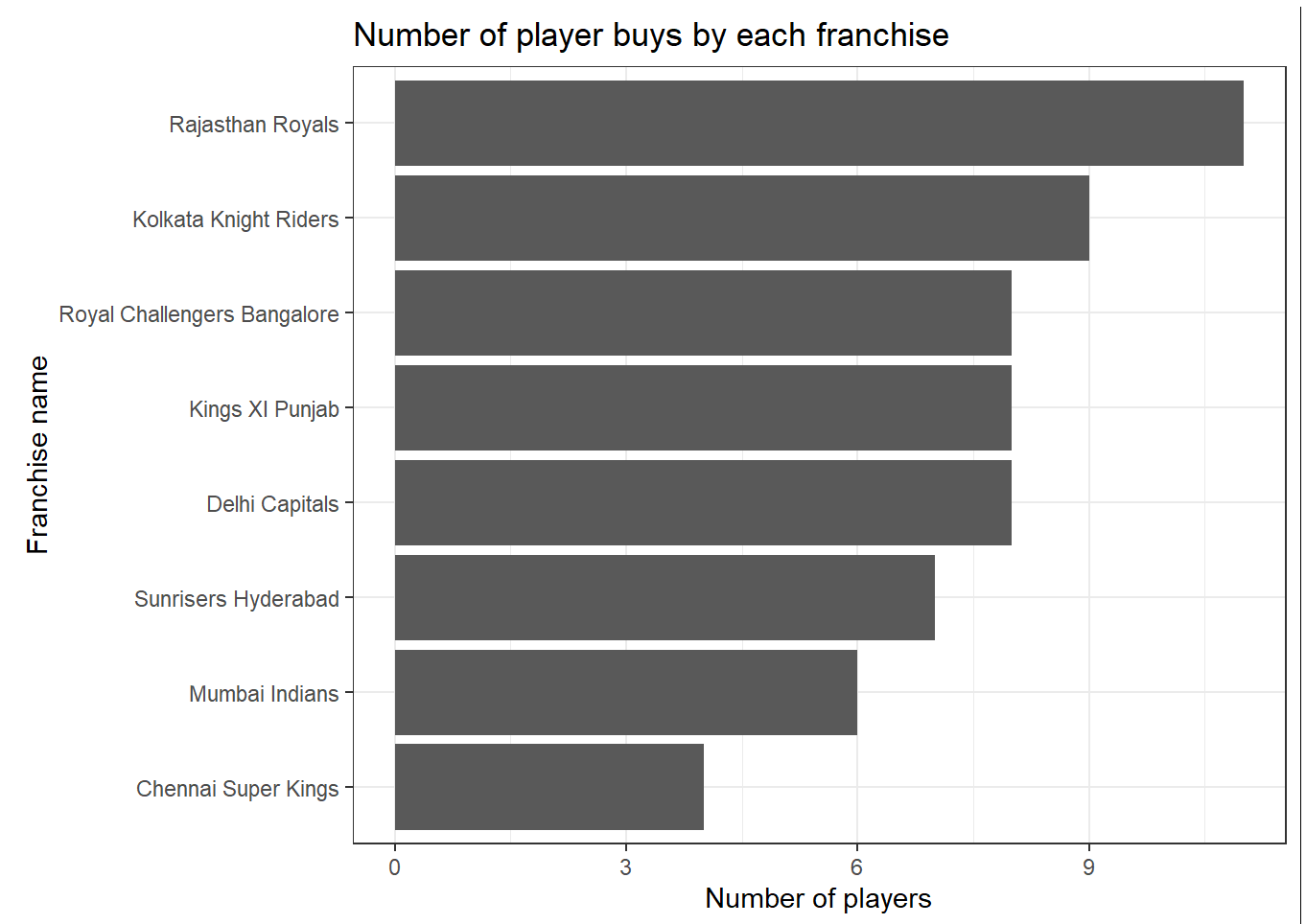
Overseas vs Indian players-



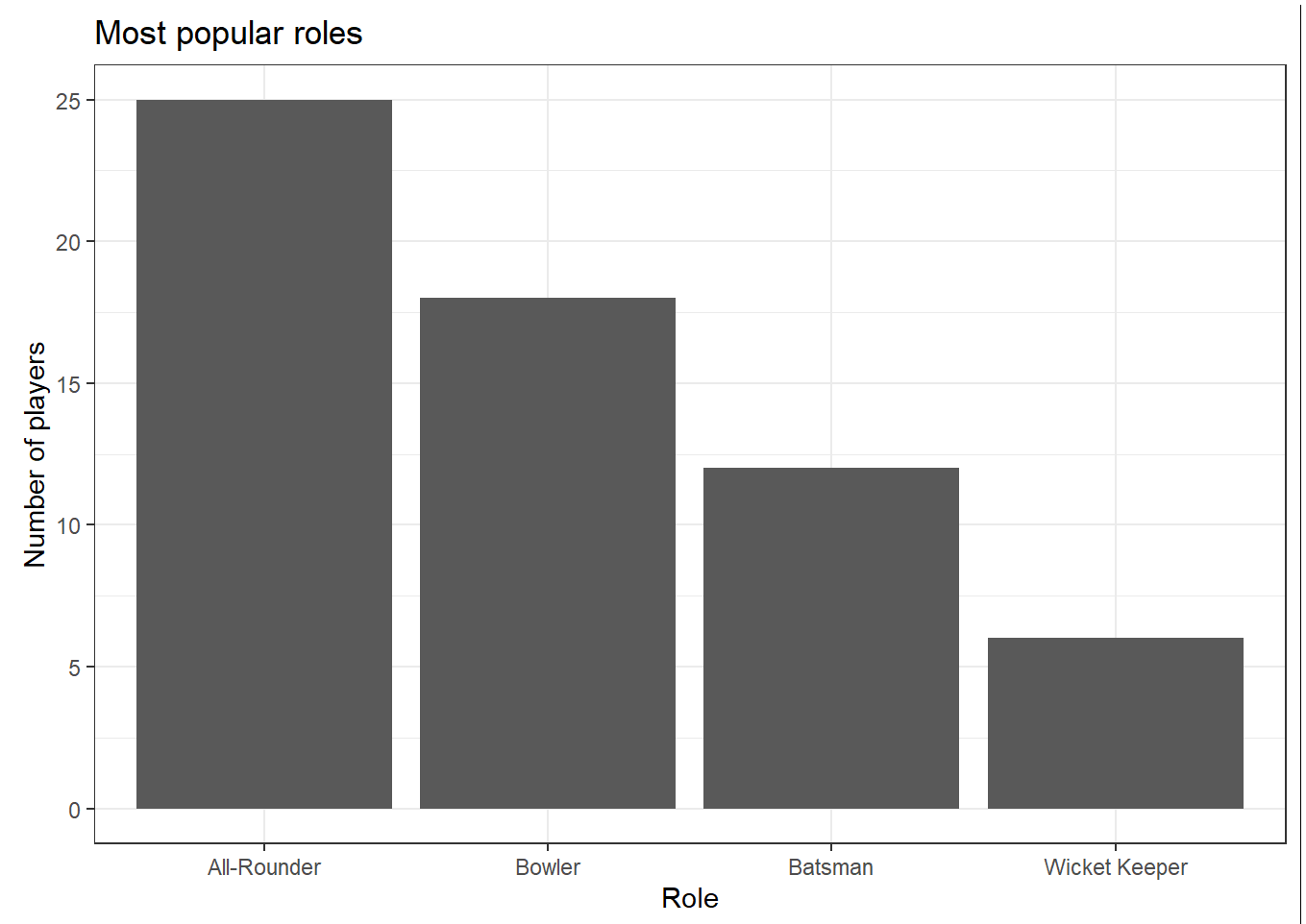
Amount spent by each franchise-



Franchises with most buys-



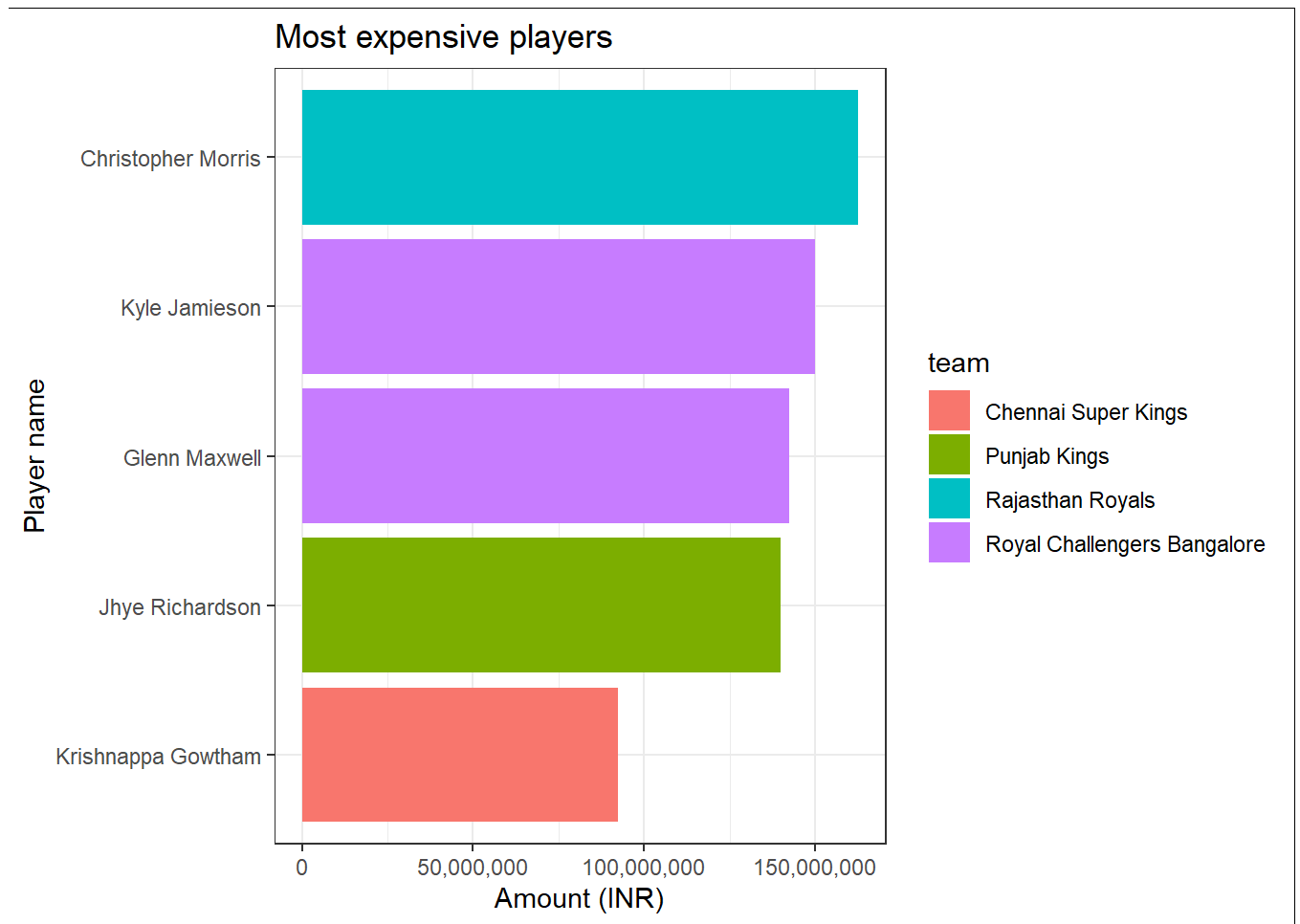
Most popular roles-



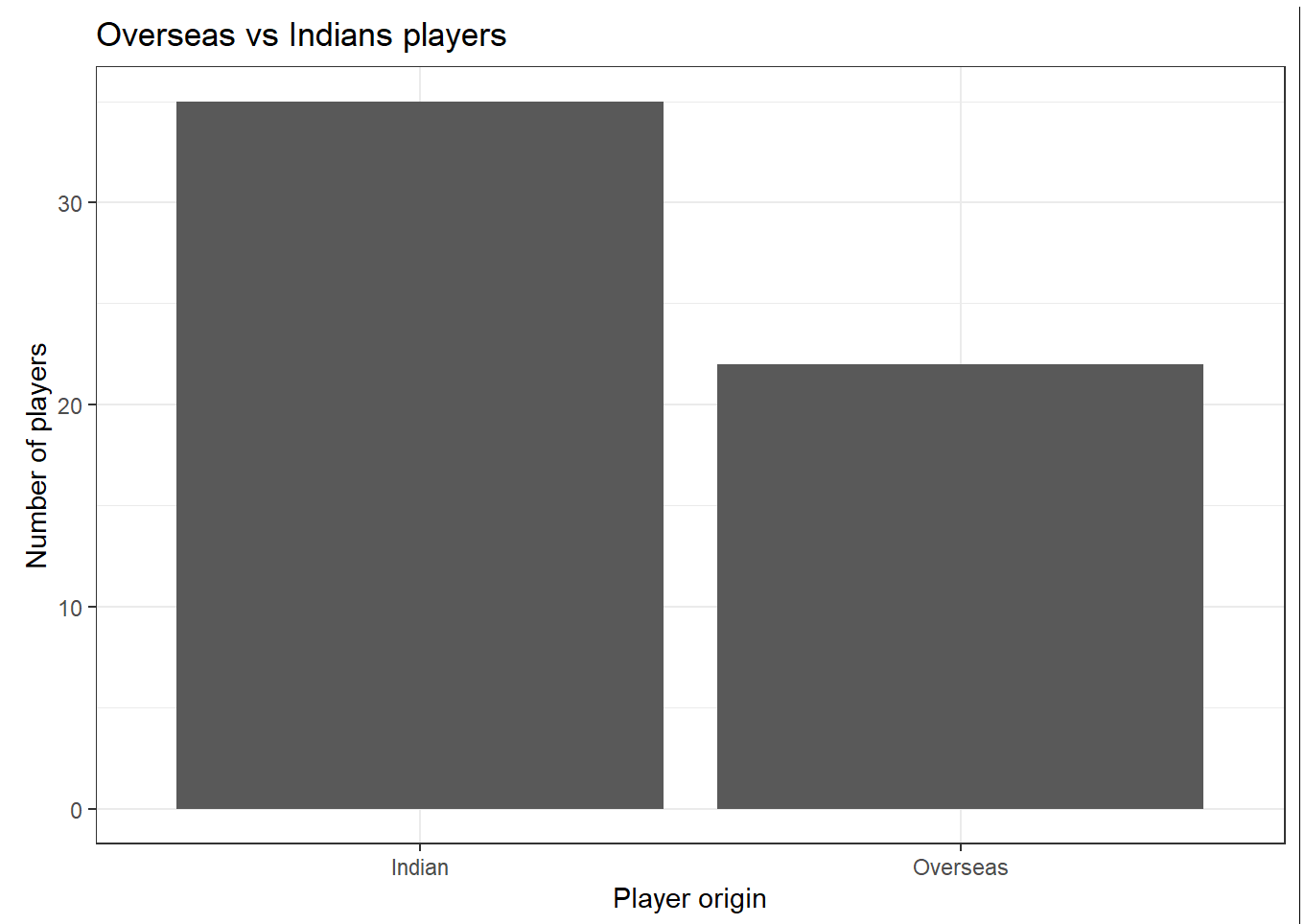
2021

Most expensive players-

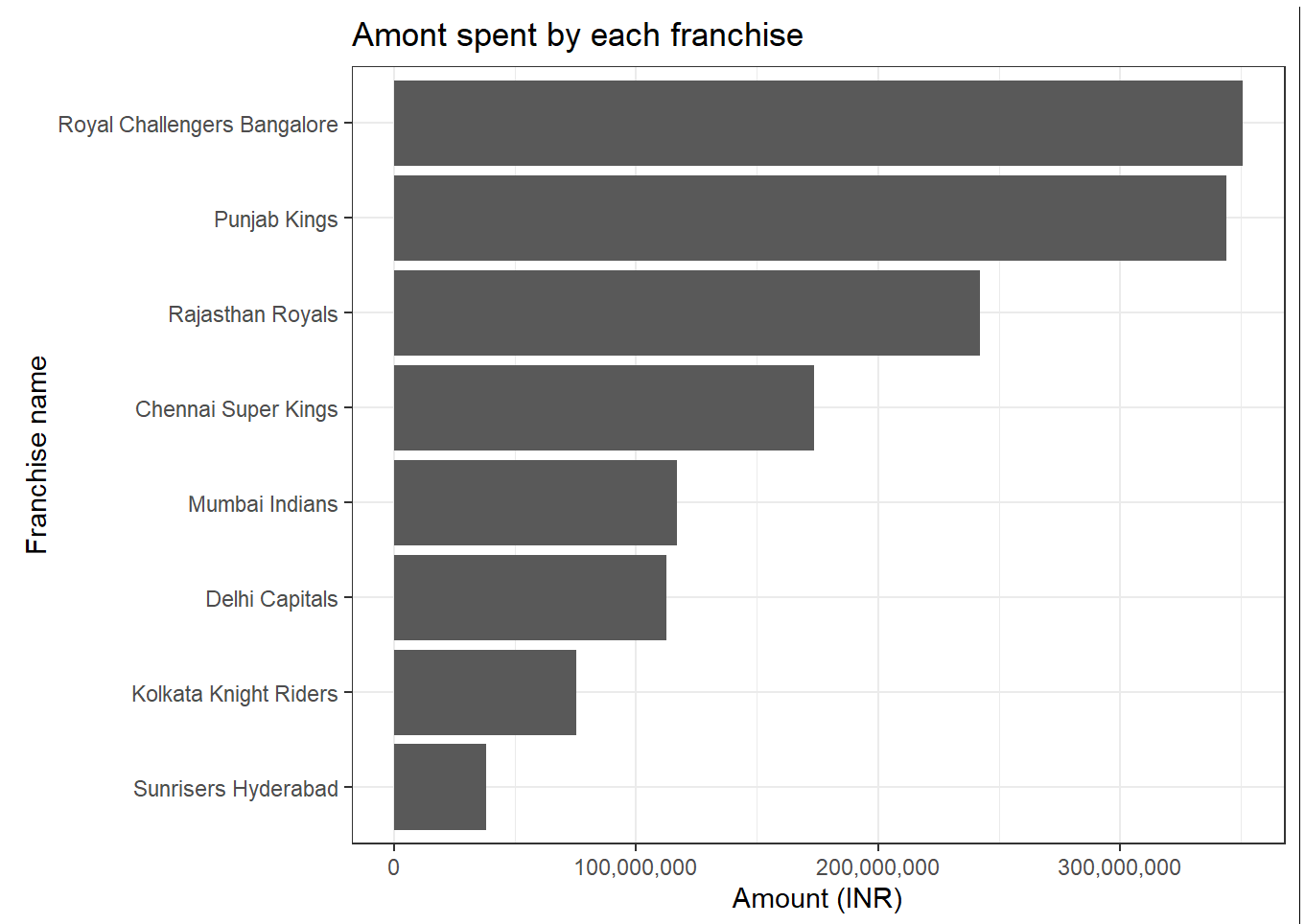
``summarise()`` has grouped output by 'player'. You can override using the ``.groups`` argument.



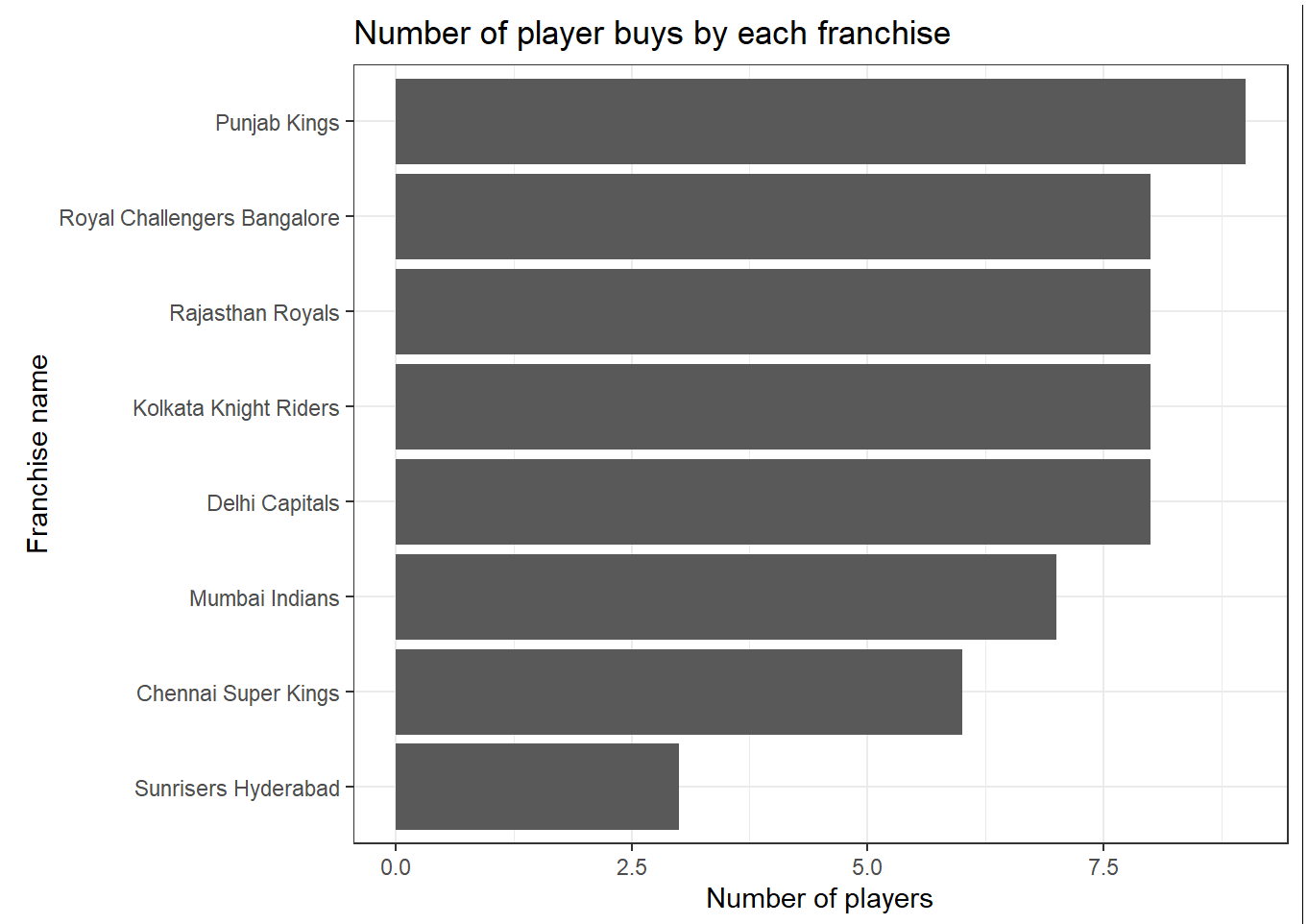
Overseas vs Indian players-



Amount spent by each franchise-

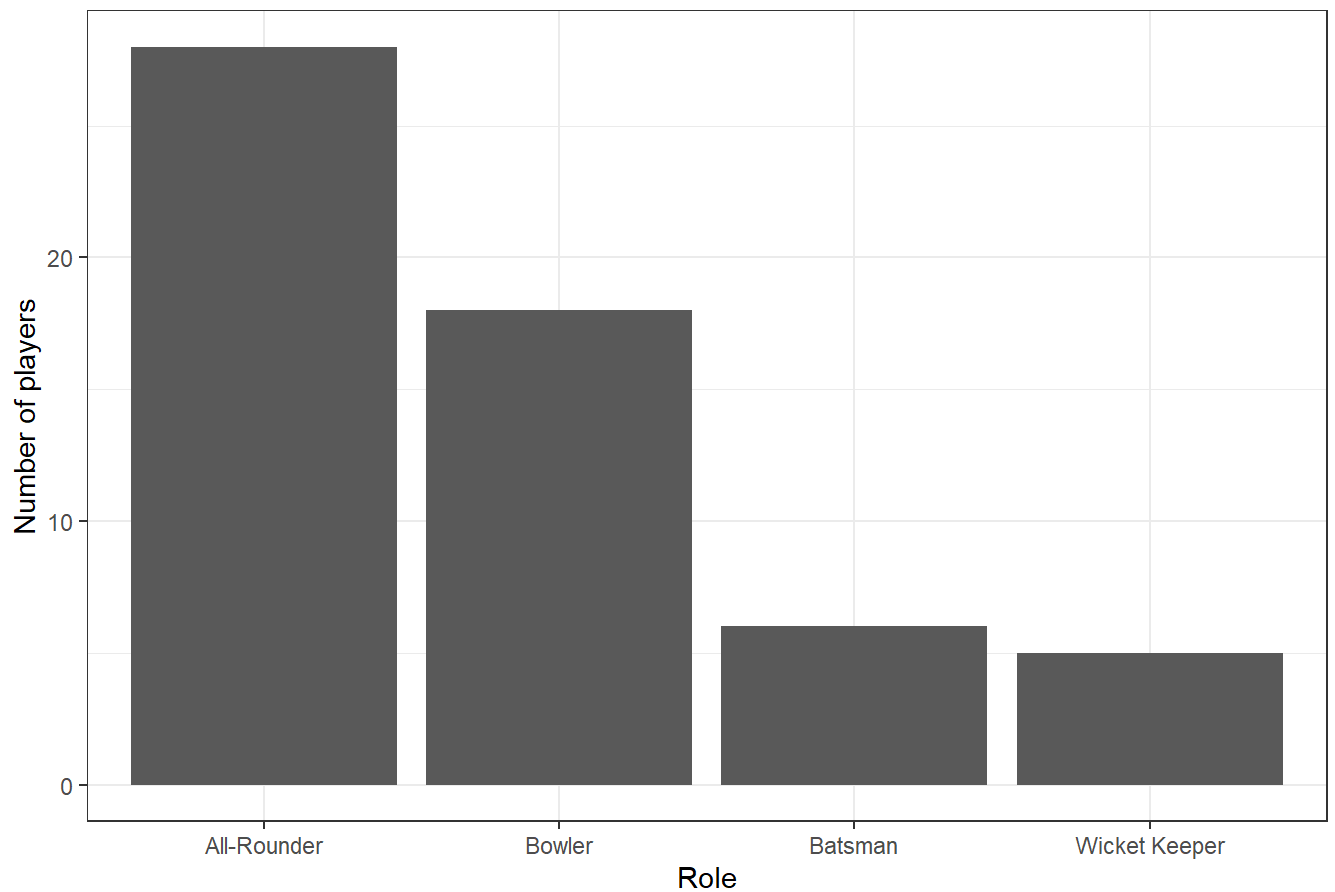


Franchises with most buys-



Most popular roles-

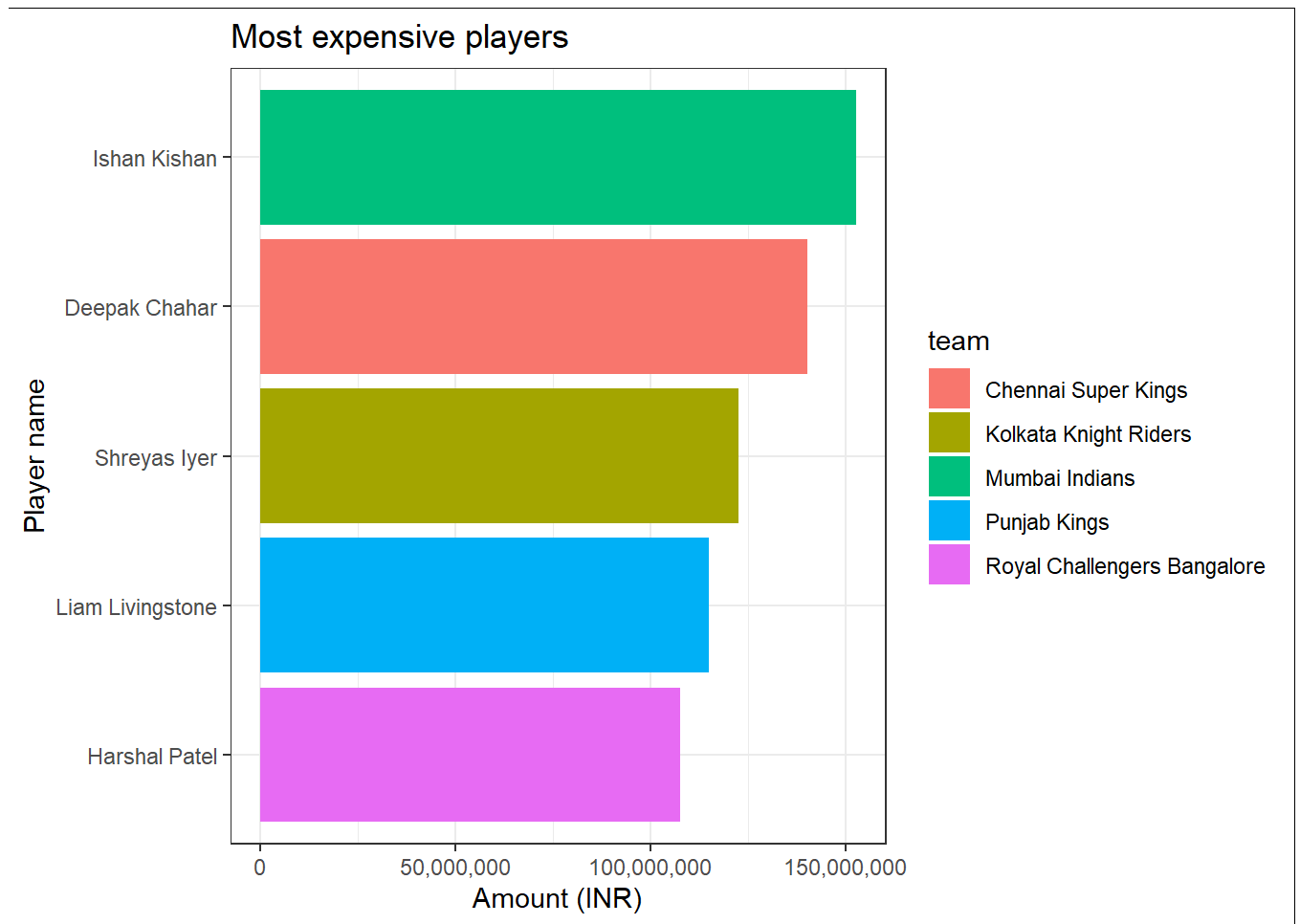
Most popular roles



2022

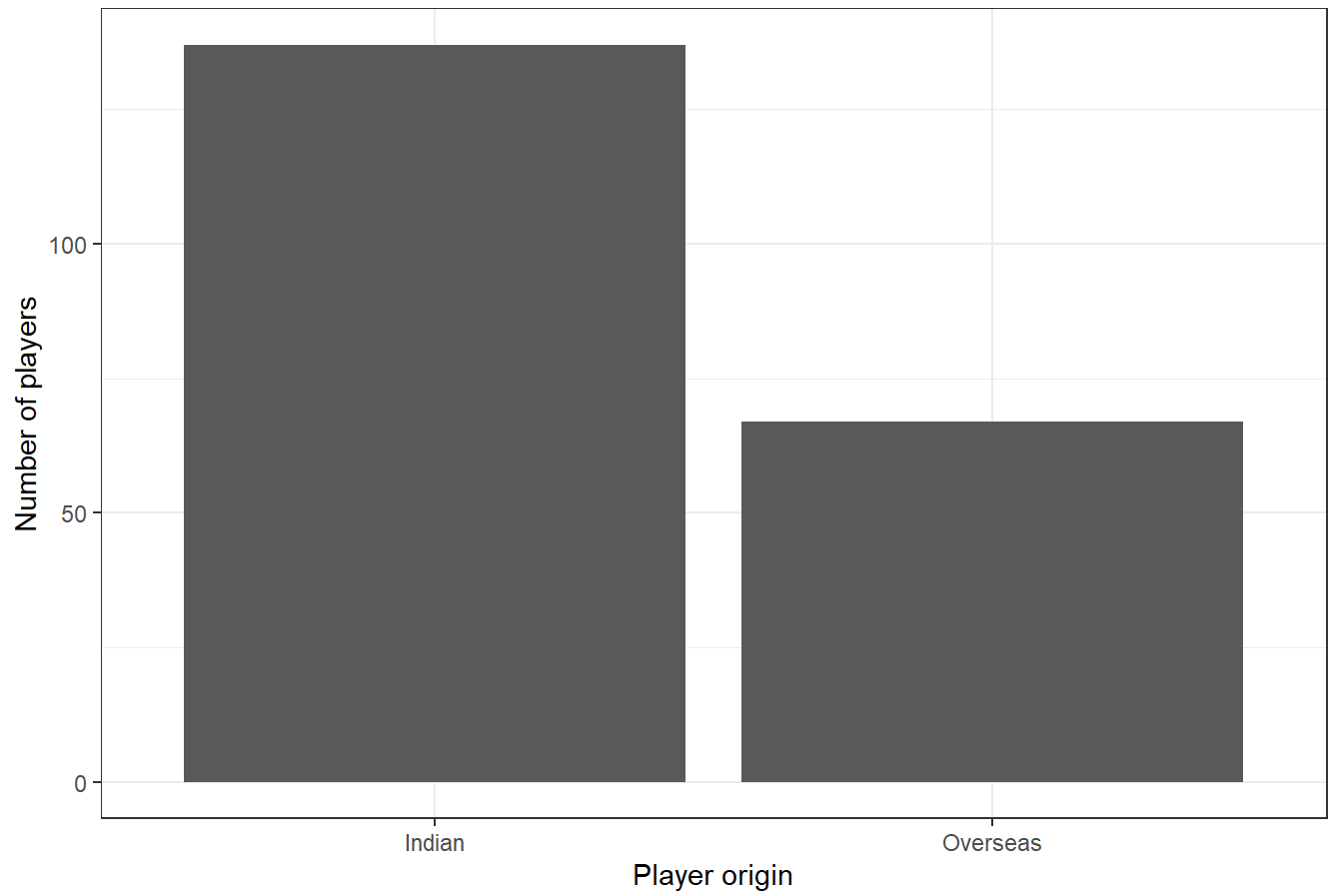
Most expensive players-

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

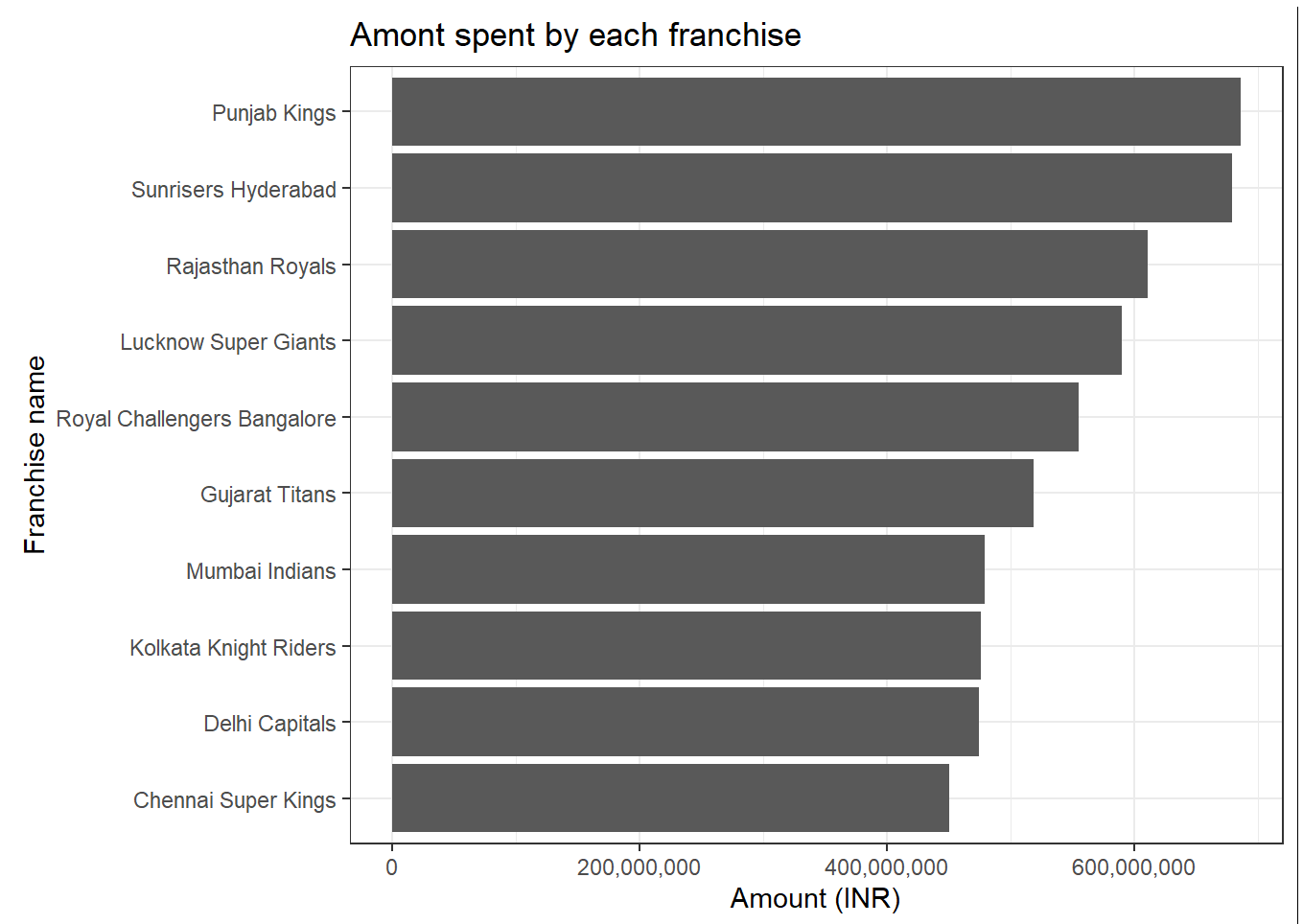


Overseas vs Indian players-

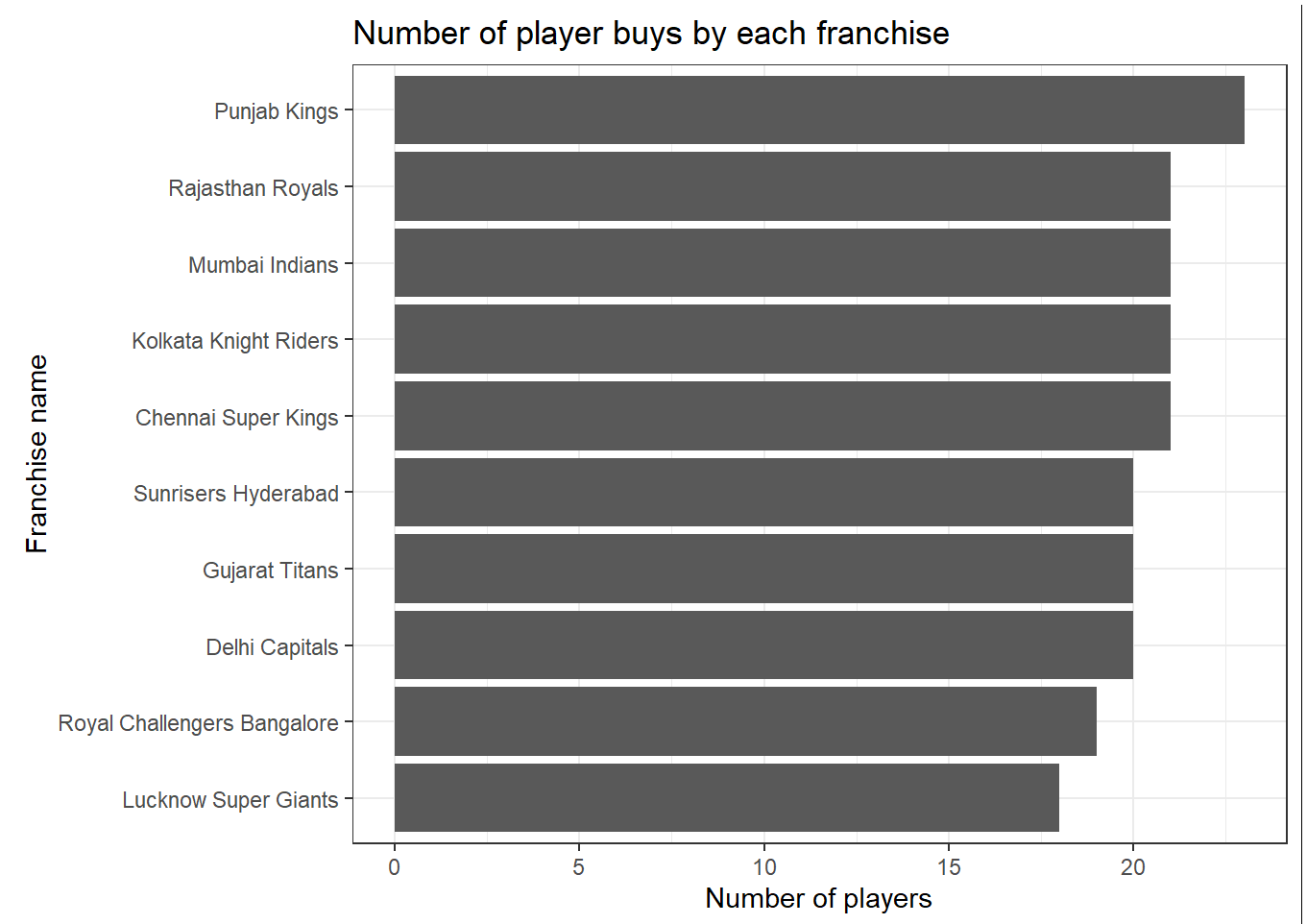
Overseas vs Indians players



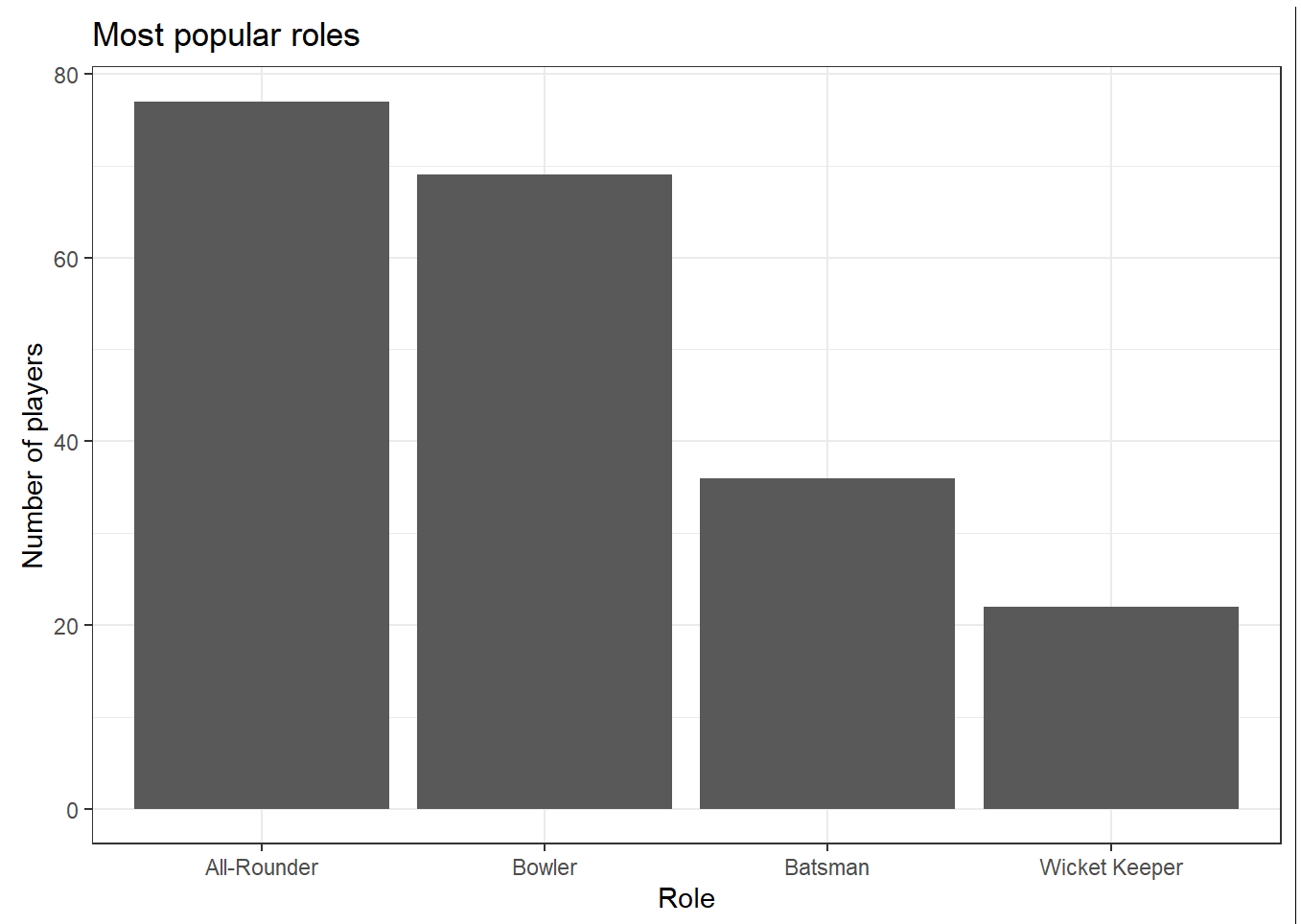
Amount spent by each franchise-



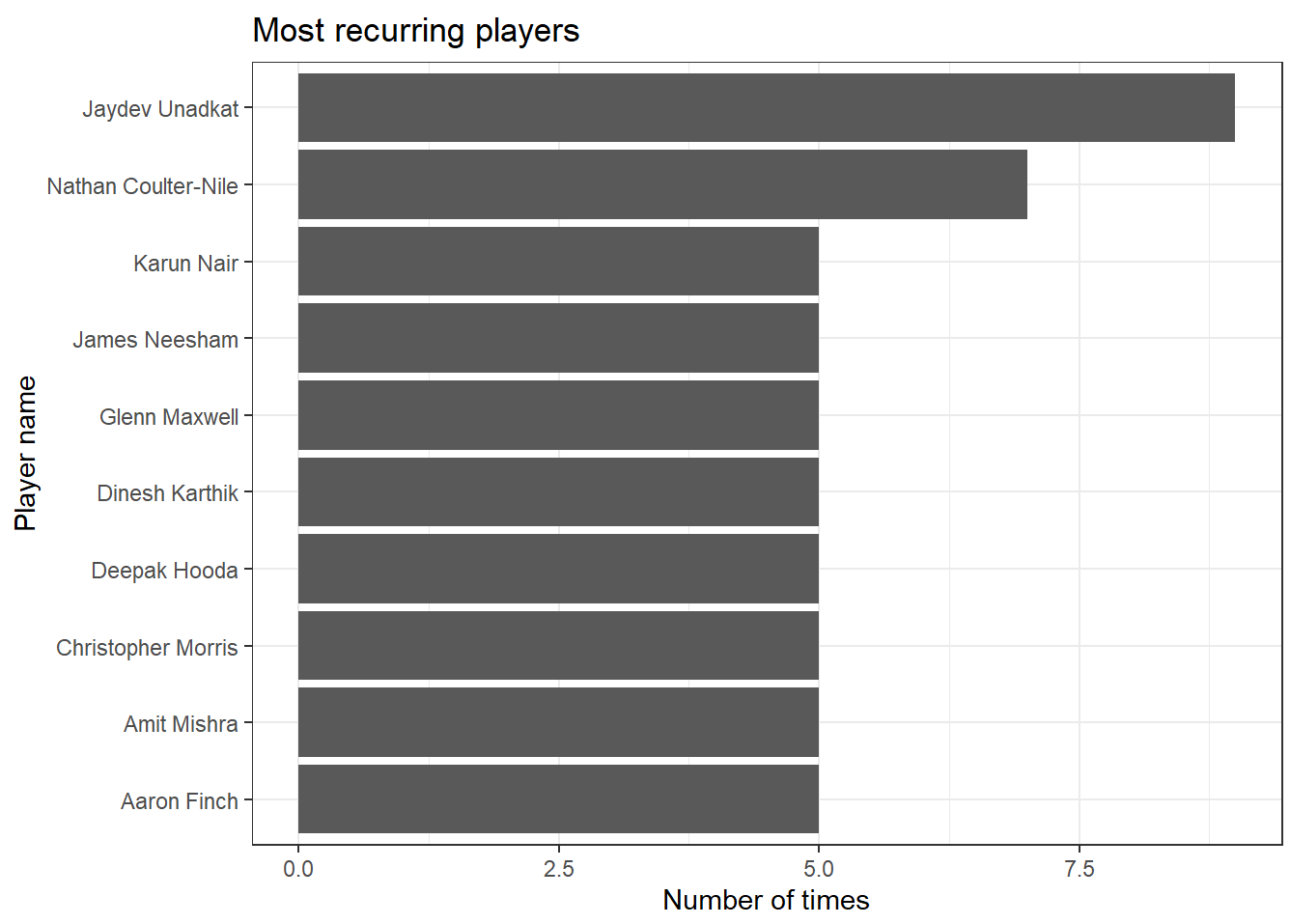
Franchises with most buys-



Most popular roles-



Most recurring players in auctions over the years-



Thank you