RATINGS FOR DIFFERENT INTERNATI0NAL CUISINES

Jeetu Kumar (B18732) And Satish Kumar Keshri (B18743)

May 1, 2019

Introduction

Here we will write introduction to project The dataset is about rating of traditional dishes from each participating country in 2014 FIFA World Cup as well as eight additional nations. The dataset has been generated by polling 1,373 Americans through SurveyMonkey Audience and they were asked to rate the national cuisines of the 32 teams that qualified for the World Cup, as well as eight additional nations with bad soccer but great food: China, Cuba, Ethiopia, India, Ireland, Thailand, Turkey and Vietnam. The dataset contains knowledge about different cuisines, interest about diffrent cuisines, gender, age, household income, educatipon, location and rating for different cuisines of each respondent.

Each participant has rated the cuisines scoring from 1 to 5 and N/A. The meanings of each rating score is as follows:

5: I love this country's traditional cuisine. I think it's one of the best in the world. 4: I like this country's traditional cuisine. I think it's considerably above average. 3: I'm OK with this county's traditional cuisine. I think it's about average. 2: I dislike this country's traditional cuisine. I think it's considerably below average. 1: I hate this country's traditional cuisine. I think it's one of the worst in the world. N/A: I'm unfamiliar with this country's traditional cuisine.

The dataset can be obtained from : https://www.kaggle.com/fivethirtyeight/fivethirtyeight-food-world-cup-dataset

Including library functions

```
library(tidyr)
library(ggrepel)
```

```
## Loading required package: ggplot2
```

```
library(dbplot)
library(ggplot2)

library(fivethirtyeight)

library(sparklyr)
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(mice)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:tidyr':
##
##     complete
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(tibble)
```

Importing Data Set

Creating spark connection; And Reading the data set to spark connection

```
sc <- spark_connect(master = "local", version = "2.0.0")

food_world_cup <- spark_read_csv(sc,"food_world_cup","/home/sysadm/Desktop/WETBDC/mainfoodworldcupdata.

food_world_cup
```

```
## # Source: spark<food_world_cup> [?? x 48]
##    respondent_id knowledge interest gender age   household_income education
##            <dbl> <chr>     <chr>    <chr>  <chr> <chr>            <chr>
##  1    3308895255 Intermed~ Some     Male   18-29 $100,000 - $149~ Less tha~
##  2    3308891308 Novice    Some     Male   18-29 $100,000 - $149~ Some col~
##  3    3308891135 Intermed~ A lot    Male   30-44 $50,000 - $99,9~ Graduate~
##  4    3308879091 Novice    Not much Male   45-60 $0 - $24,999     Less tha~
##  5    3308871671 Novice    Not much Male   30-44 $25,000 - $49,9~ High sch~
##  6    3308871406 Advanced  A lot    Female 30-44 $50,000 - $99,9~ Graduate~
##  7    3308866182 Novice    Some     Male   45-60 NA               High sch~
##  8    3308857114 Advanced  A lot    Male   45-60 $0 - $24,999     Some col~
##  9    3308856510 Novice    Not much Female 30-44 $50,000 - $99,9~ Some col~
## 10    3308846915 Novice    Some     NA     NA    NA               NA
## # ... with more rows, and 41 more variables: location <chr>,
## #   algeria <chr>, argentina <chr>, australia <chr>, belgium <chr>,
## #   bosnia_and_herzegovina <chr>, brazil <chr>, cameroon <chr>,
## #   chile <chr>, china <chr>, colombia <chr>, costa_rica <chr>,
## #   croatia <chr>, cuba <chr>, ecuador <chr>, england <chr>,
```

```
## #   ethiopia <chr>, france <chr>, germany <chr>, ghana <chr>,
## #   greece <chr>, honduras <chr>, india <chr>, iran <chr>, ireland <chr>,
## #   italy <chr>, ivory_coast <chr>, japan <chr>, mexico <chr>,
## #   nigeria <chr>, portugal <chr>, russia <chr>, south_korea <chr>,
## #   spain <chr>, switzerland <chr>, thailand <chr>, the_netherlands <chr>,
## #   turkey <chr>, united_states <chr>, uruguay <chr>, vietnam <chr>
```

Replacing missing values

The dataset contains three type of missing values, viz. NA, N/A and white space. Here we will replace missing values NA, N/A and white space to NA and then overwrite the data to spark connection.

```
food_world_cup <- food_world_cup %>%

  mutate(belgium=ifelse(belgium=="N/A", "NA", belgium)) %>%
  mutate(belgium=ifelse(belgium=="", "NA", belgium))%>%

  mutate(bosnia_and_herzegovina=ifelse(bosnia_and_herzegovina=="N/A", "NA", bosnia_and_herzegovina)) %>%
  mutate(bosnia_and_herzegovina=ifelse(bosnia_and_herzegovina=="", "NA", bosnia_and_herzegovina))%>%

  mutate(brazil=ifelse(brazil=="N/A", "NA", brazil)) %>%
  mutate(brazil=ifelse(brazil=="", "NA", brazil))%>%

  mutate(cameroon=ifelse(cameroon=="N/A", "NA", cameroon)) %>%
  mutate(cameroon=ifelse(cameroon=="", "NA", cameroon))%>%

  mutate(chile=ifelse(chile=="N/A", "NA", chile)) %>%
  mutate(chile=ifelse(chile=="", "NA", chile))%>%

  mutate(china=ifelse(china=="N/A", "NA", china)) %>%
  mutate(china=ifelse(china=="", "NA", china))%>%

  mutate(colombia=ifelse(colombia=="N/A", "NA", colombia)) %>%
  mutate(colombia=ifelse(colombia=="", "NA", colombia))%>%

  mutate(costa_rica=ifelse(costa_rica=="N/A", "NA", costa_rica)) %>%
  mutate(costa_rica=ifelse(costa_rica=="", "NA", costa_rica))%>%

  mutate(croatia=ifelse(croatia=="N/A", "NA", croatia)) %>%
  mutate(croatia=ifelse(croatia=="", "NA", croatia))%>%

  mutate(cuba=ifelse(cuba=="N/A", "NA", cuba)) %>%
  mutate(cuba=ifelse(cuba=="", "NA", cuba))%>%

  mutate(ecuador=ifelse(ecuador=="N/A", "NA", ecuador)) %>%

  mutate(england=ifelse(england=="N/A", "NA", england)) %>%
  mutate(england=ifelse(england=="", "NA", england))%>%

  mutate(ethiopia=ifelse(ethiopia=="N/A", "NA", ethiopia)) %>%
  mutate(ethiopia=ifelse(ethiopia=="", "NA", ethiopia))%>%

  mutate(france=ifelse(france=="N/A", "NA", france)) %>%
```

```
  mutate(france=ifelse(france=="", "NA", france))%>%

  mutate(germany=ifelse(germany=="N/A", "NA", germany)) %>%
  mutate(germany=ifelse(germany=="", "NA", germany))%>%

  mutate(ghana=ifelse(ghana=="N/A", "NA", ghana)) %>%
  mutate(ghana=ifelse(ghana=="", "NA", ghana))%>%

  mutate(greece=ifelse(greece=="N/A", "NA", greece)) %>%
  mutate(greece=ifelse(greece=="", "NA", greece))%>%

  mutate(honduras=ifelse(honduras=="N/A", "NA", honduras)) %>%
  mutate(honduras=ifelse(honduras=="", "NA", honduras))%>%

  mutate(india=ifelse(india=="N/A", "NA", india)) %>%
  mutate(india=ifelse(india=="", "NA", india))%>%

  mutate(iran=ifelse(iran=="N/A", "NA", iran)) %>%
  mutate(iran=ifelse(iran=="", "NA", iran))%>%

  mutate(ireland=ifelse(ireland=="N/A", "NA", ireland)) %>%
  mutate(ireland=ifelse(ireland=="", "NA", ireland))%>%

  mutate(italy=ifelse(italy=="N/A", "NA", italy)) %>%
  mutate(italy=ifelse(italy=="", "NA", italy))


spark_write_csv(food_world_cup,"Sparkoutput",mode="overwrite")

food_world_cup <- spark_read_csv(sc,"food_world_cup","Sparkoutput", overwrite = TRUE)

food_world_cup <- food_world_cup %>%
  mutate(ivory_coast=ifelse(ivory_coast=="N/A", "NA", ivory_coast)) %>%
  mutate(ivory_coast=ifelse(ivory_coast=="", "NA", ivory_coast))%>%

  mutate(japan=ifelse(japan=="N/A", "NA", japan)) %>%
  mutate(japan=ifelse(japan=="", "NA", japan))%>%

  mutate(mexico=ifelse(mexico=="N/A", "NA", mexico)) %>%
  mutate(mexico=ifelse(mexico=="", "NA", mexico))%>%

  mutate(nigeria=ifelse(nigeria=="N/A", "NA", nigeria)) %>%
  mutate(nigeria=ifelse(nigeria=="", "NA", nigeria))%>%

  mutate(portugal=ifelse(portugal=="N/A", "NA", portugal)) %>%
  mutate(portugal=ifelse(portugal=="", "NA", portugal))%>%

  mutate(russia=ifelse(russia=="N/A", "NA", russia)) %>%
  mutate(russia=ifelse(russia=="", "NA", russia))%>%

  mutate(south_korea=ifelse(south_korea=="N/A", "NA", south_korea)) %>%
  mutate(south_korea=ifelse(south_korea=="", "NA", south_korea))%>%
```

```
  mutate(spain=ifelse(spain=="N/A", "NA", spain)) %>%
  mutate(spain=ifelse(spain=="", "NA", spain))%>%

  mutate(switzerland=ifelse(switzerland=="N/A", "NA", switzerland)) %>%
  mutate(switzerland=ifelse(switzerland=="", "NA", switzerland))%>%

  mutate(thailand=ifelse(thailand=="N/A", "NA", thailand)) %>%
  mutate(thailand=ifelse(thailand=="", "NA", thailand))%>%

  mutate(the_netherlands=ifelse(the_netherlands=="N/A", "NA", the_netherlands)) %>%
  mutate(the_netherlands=ifelse(the_netherlands=="", "NA", the_netherlands))%>%

  mutate(turkey=ifelse(turkey=="N/A", "NA", turkey)) %>%
  mutate(turkey=ifelse(turkey=="", "NA", turkey))


spark_write_csv(food_world_cup,"Sparkoutput",mode="overwrite")

food_world_cup <- spark_read_csv(sc,"food_world_cup","Sparkoutput", overwrite = TRUE)

food_world_cup <- food_world_cup %>%

  mutate(united_states=ifelse(united_states=="N/A", "NA", united_states)) %>%
  mutate(united_states=ifelse(united_states=="", "NA", united_states))%>%

  mutate(uruguay=ifelse(uruguay=="N/A", "NA", uruguay)) %>%
  mutate(uruguay=ifelse(uruguay=="", "NA", uruguay))%>%

  mutate(vietnam=ifelse(vietnam=="N/A", "NA", vietnam)) %>%
  mutate(vietnam=ifelse(vietnam=="", "NA", vietnam))%>%

  mutate(algeria=ifelse(algeria=="N/A", "NA", algeria)) %>%
  mutate(algeria=ifelse(algeria=="", "NA", algeria))%>%

  mutate(argentina=ifelse(argentina=="N/A", "NA", argentina)) %>%
  mutate(argentina=ifelse(argentina=="", "NA", argentina))%>%

  mutate(australia=ifelse(australia=="N/A", "NA", australia)) %>%
  mutate(australia=ifelse(australia=="", "NA", australia))%>%

  mutate(knowledge=ifelse(knowledge=="N/A", "NA", knowledge)) %>%
  mutate(knowledge=ifelse(knowledge=="", "NA", knowledge))%>%

  mutate(interest=ifelse(interest=="N/A", "NA", interest)) %>%
  mutate(interest=ifelse(interest=="", "NA", interest))%>%

  mutate(gender=ifelse(gender=="N/A", "NA", gender)) %>%
  mutate(gender=ifelse(gender=="", "NA", gender))%>%

  mutate(age=ifelse(age=="N/A", "NA", age)) %>%
  mutate(age=ifelse(age=="", "NA", age))%>%

  mutate(household_income=ifelse(household_income=="N/A", "NA", household_income)) %>%
```

```
  mutate(household_income=ifelse(household_income=="", "NA", household_income))%>%

  mutate(education=ifelse(education=="N/A", "NA", education)) %>%
  mutate(education=ifelse(education=="", "NA", education))%>%

  mutate(location=ifelse(location=="N/A", "NA", location)) %>%
  mutate(location=ifelse(location=="", "NA", location))


spark_write_csv(food_world_cup,"Sparkoutput",mode="overwrite")

food_world_cup <- spark_read_csv(sc,"food_world_cup","Sparkoutput", overwrite = TRUE)

food_world_cup
```

```
## # Source: spark<food_world_cup> [?? x 48]
##    respondent_id knowledge interest gender age   household_income education
##            <dbl> <chr>     <chr>    <chr>  <chr> <chr>            <chr>
##  1    3308895255 Intermed~ Some     Male   18-29 $100,000 - $149~ Less tha~
##  2    3308891308 Novice    Some     Male   18-29 $100,000 - $149~ Some col~
##  3    3308891135 Intermed~ A lot    Male   30-44 $50,000 - $99,9~ Graduate~
##  4    3308879091 Novice    Not much Male   45-60 $0 - $24,999     Less tha~
##  5    3308871671 Novice    Not much Male   30-44 $25,000 - $49,9~ High sch~
##  6    3308871406 Advanced  A lot    Female 30-44 $50,000 - $99,9~ Graduate~
##  7    3308866182 Novice    Some     Male   45-60 NA               High sch~
##  8    3308857114 Advanced  A lot    Male   45-60 $0 - $24,999     Some col~
##  9    3308856510 Novice    Not much Female 30-44 $50,000 - $99,9~ Some col~
## 10    3308846915 Novice    Some     NA     NA    NA               NA
## # ... with more rows, and 41 more variables: location <chr>,
## #   algeria <chr>, argentina <chr>, australia <chr>, belgium <chr>,
## #   bosnia_and_herzegovina <chr>, brazil <chr>, cameroon <chr>,
## #   chile <chr>, china <chr>, colombia <chr>, costa_rica <chr>,
## #   croatia <chr>, cuba <chr>, ecuador <chr>, england <chr>,
## #   ethiopia <chr>, france <chr>, germany <chr>, ghana <chr>,
## #   greece <chr>, honduras <chr>, india <chr>, iran <chr>, ireland <chr>,
## #   italy <chr>, ivory_coast <chr>, japan <chr>, mexico <chr>,
## #   nigeria <chr>, portugal <chr>, russia <chr>, south_korea <chr>,
## #   spain <chr>, switzerland <chr>, thailand <chr>, the_netherlands <chr>,
## #   turkey <chr>, united_states <chr>, uruguay <chr>, vietnam <chr>
```

```
colnames(food_world_cup)
```

```
##  [1] "respondent_id"          "knowledge"
##  [3] "interest"               "gender"
##  [5] "age"                    "household_income"
##  [7] "education"              "location"
##  [9] "algeria"                "argentina"
## [11] "australia"              "belgium"
## [13] "bosnia_and_herzegovina" "brazil"
## [15] "cameroon"               "chile"
## [17] "china"                  "colombia"
## [19] "costa_rica"             "croatia"
```

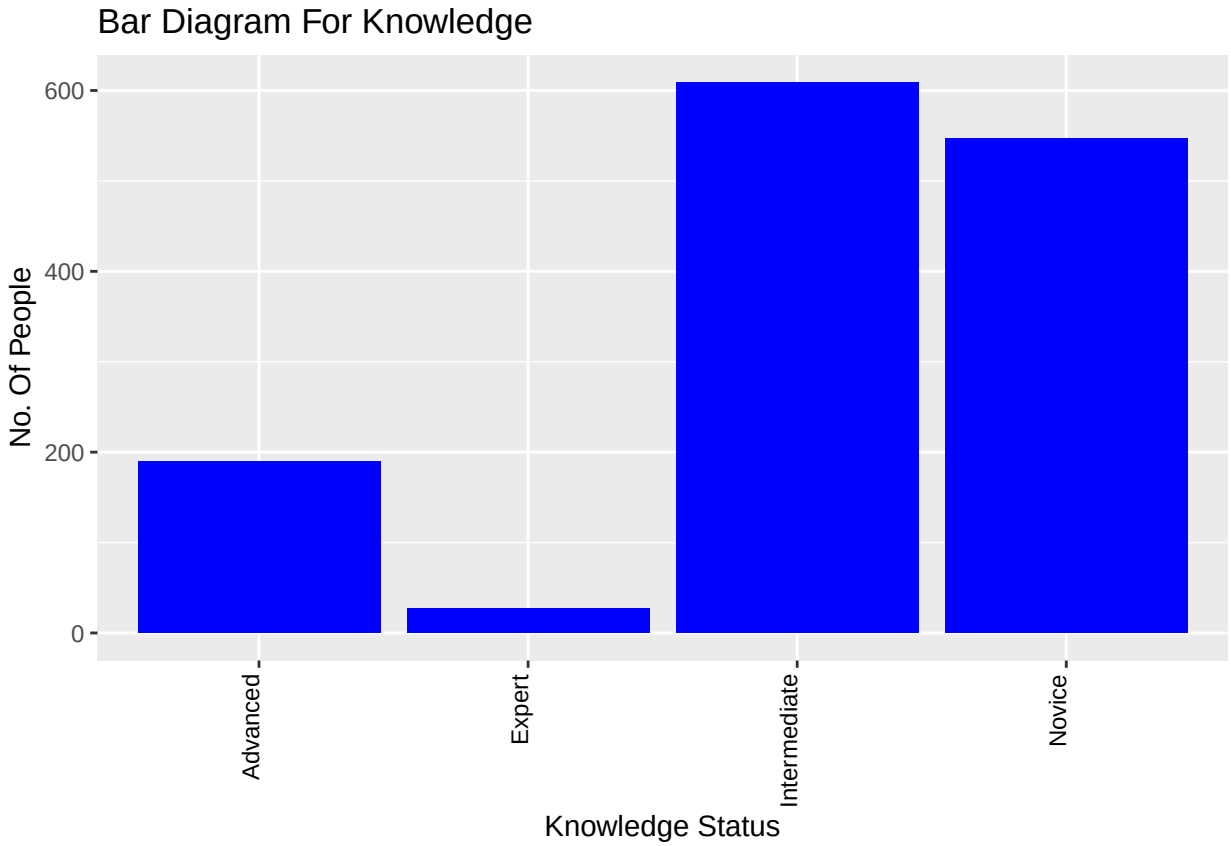
```
## [21] "cuba"                 "ecuador"
## [23] "england"              "ethiopia"
## [25] "france"               "germany"
## [27] "ghana"                "greece"
## [29] "honduras"             "india"
## [31] "iran"                 "ireland"
## [33] "italy"                "ivory_coast"
## [35] "japan"                "mexico"
## [37] "nigeria"              "portugal"
## [39] "russia"               "south_korea"
## [41] "spain"                "switzerland"
## [43] "thailand"             "the_netherlands"
## [45] "turkey"               "united_states"
## [47] "uruguay"              "vietnam"
```

Analysis of Distribution of Missing Values for Selected Columns

Here we will analize the distribution of missing values for these columns: knowedge, interest, gender, age, household_income, education, location, algeria, china, india, spain, switzerland, england, mexico. The we will replace them with appropriate values.

For Column 'knowledge'

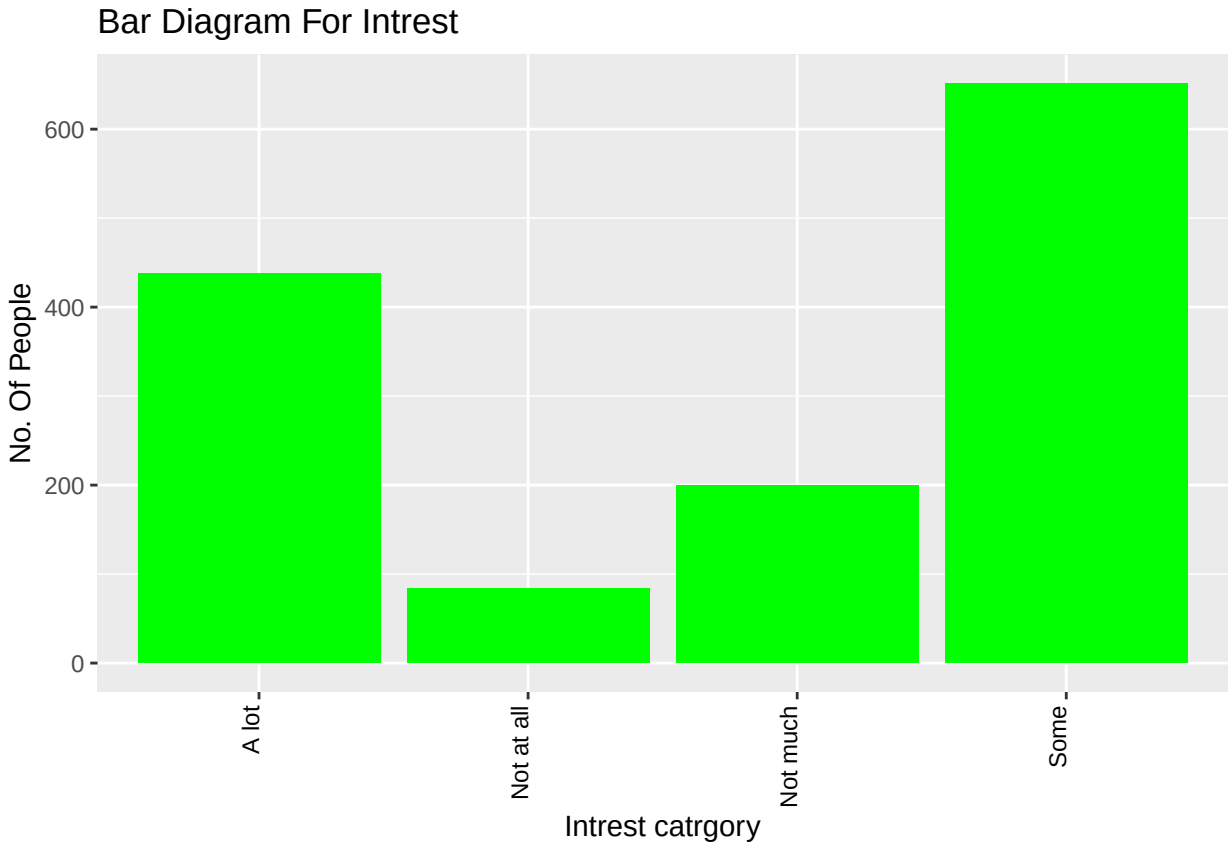
```
food_world_cup %>%
  ggplot(aes(x=factor(knowledge)))+
  geom_bar(stat="count", fill = 'blue')+
  xlab("Knowledge Status")+
  ylab("No. Of People")+
  ggtitle("Bar Diagram For Knowledge")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'black'))
```



We can see that there is no 'NA' in 'knowledge' column, so we do not needs any replacement. Also we can see that count of 'Advanced' category is very low and class 'Expert' even suffering in count, where as class 'Intermediate' is high and class 'Novice' in competition with 'Intermediate'.

For Column 'interest'

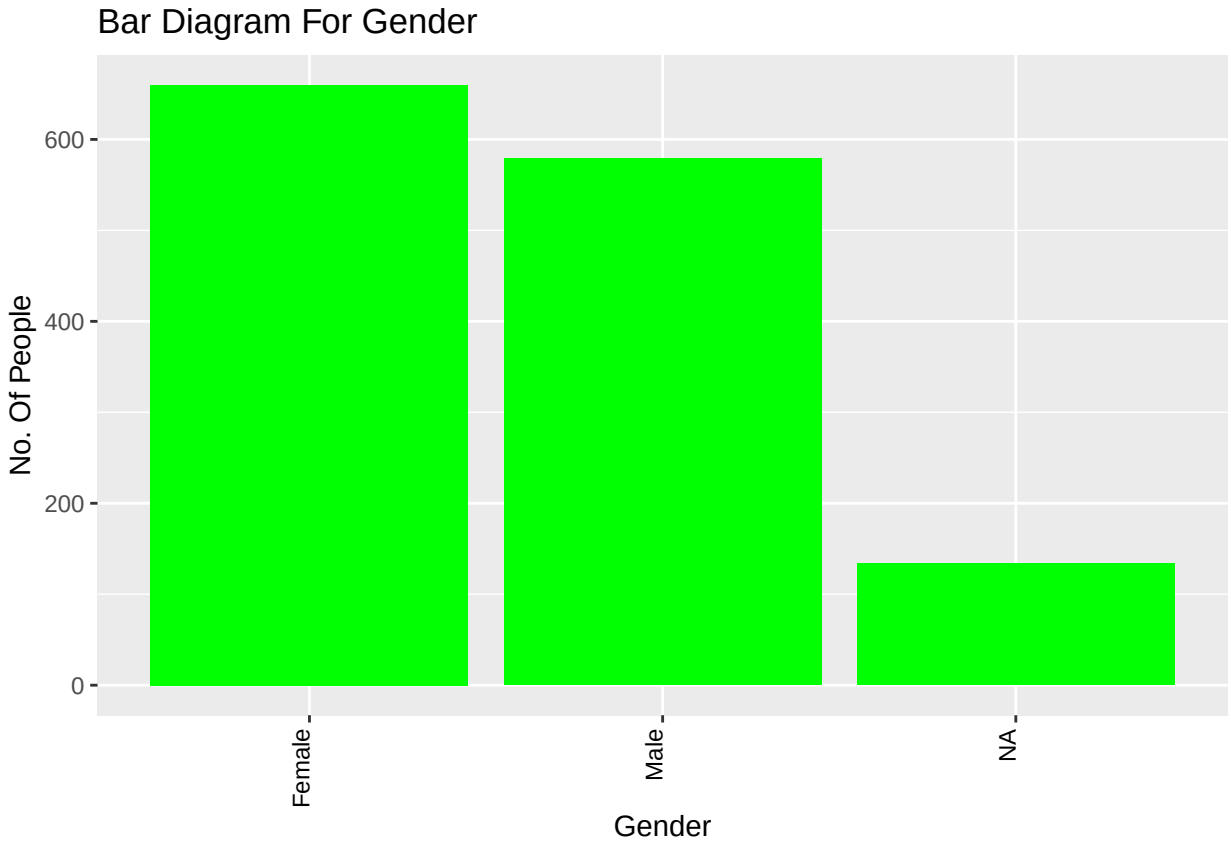
```
food_world_cup %>%
  ggplot(aes(x=factor(interest)))+
  geom_bar(stat="count", fill = 'green')+
  xlab("Intrest catrgory")+
  ylab("No. Of People")+
  ggtitle("Bar Diagram For Intrest")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'black'))
```



Here Also we can see that there is no 'NA' in 'intrest' column, so we do not needs any replacement. Also we can see that interst category 'Some' have highest count and count of category 'A lot' is adequate, as expected the category 'Not at all' have lowest count but not completely low.

For Column 'gender'

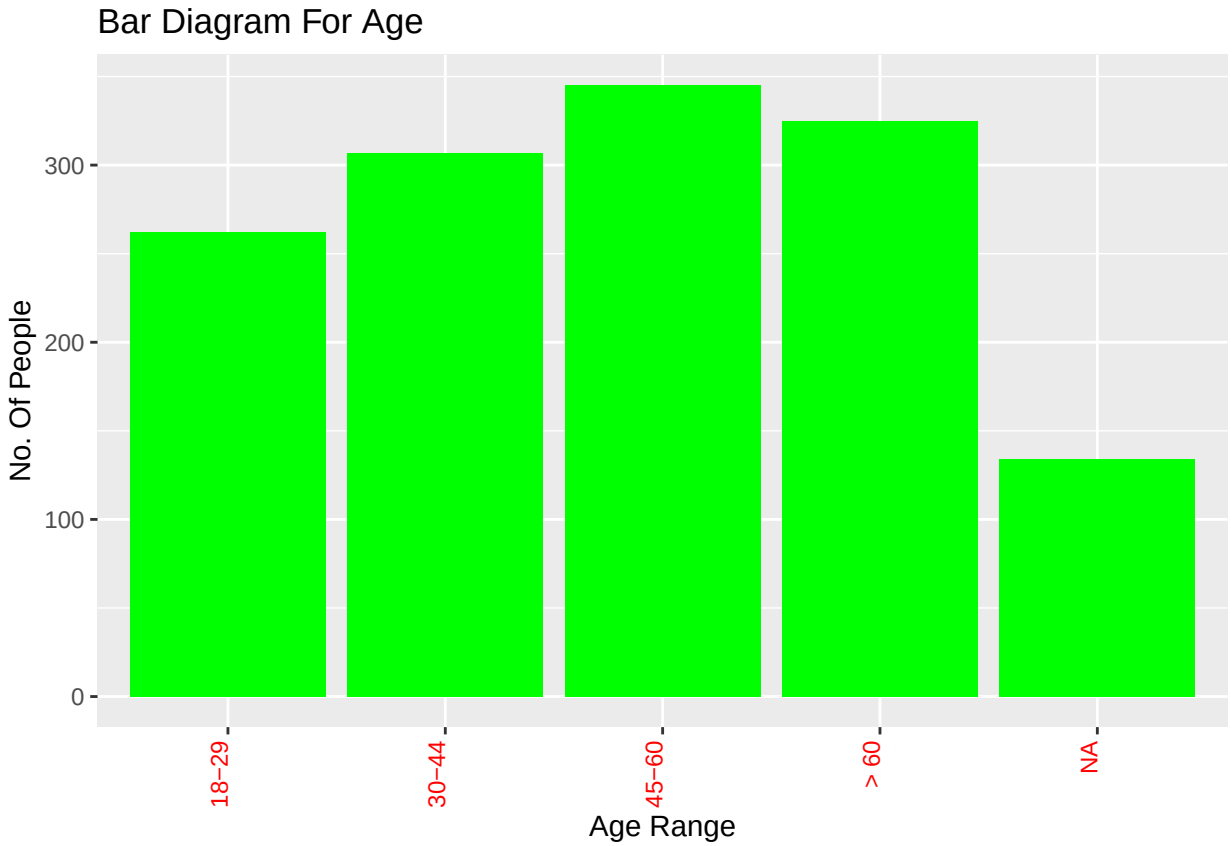
```
food_world_cup %>%
  ggplot(aes(x=factor(gender)))+
  geom_bar(stat="count", fill = 'green')+
  xlab("Gender")+
  ylab("No. Of People")+
  ggtitle("Bar Diagram For Gender")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'black'))
```



Here we can see that there is 'NA' values. But we wil not replace it, as we cannot determine gender by just looking on some numbers, also we have less information about data so it may be the case of 'thrid gender'. And we can see that number of 'Female' participant is more then number of 'Male' participant.

For Column age

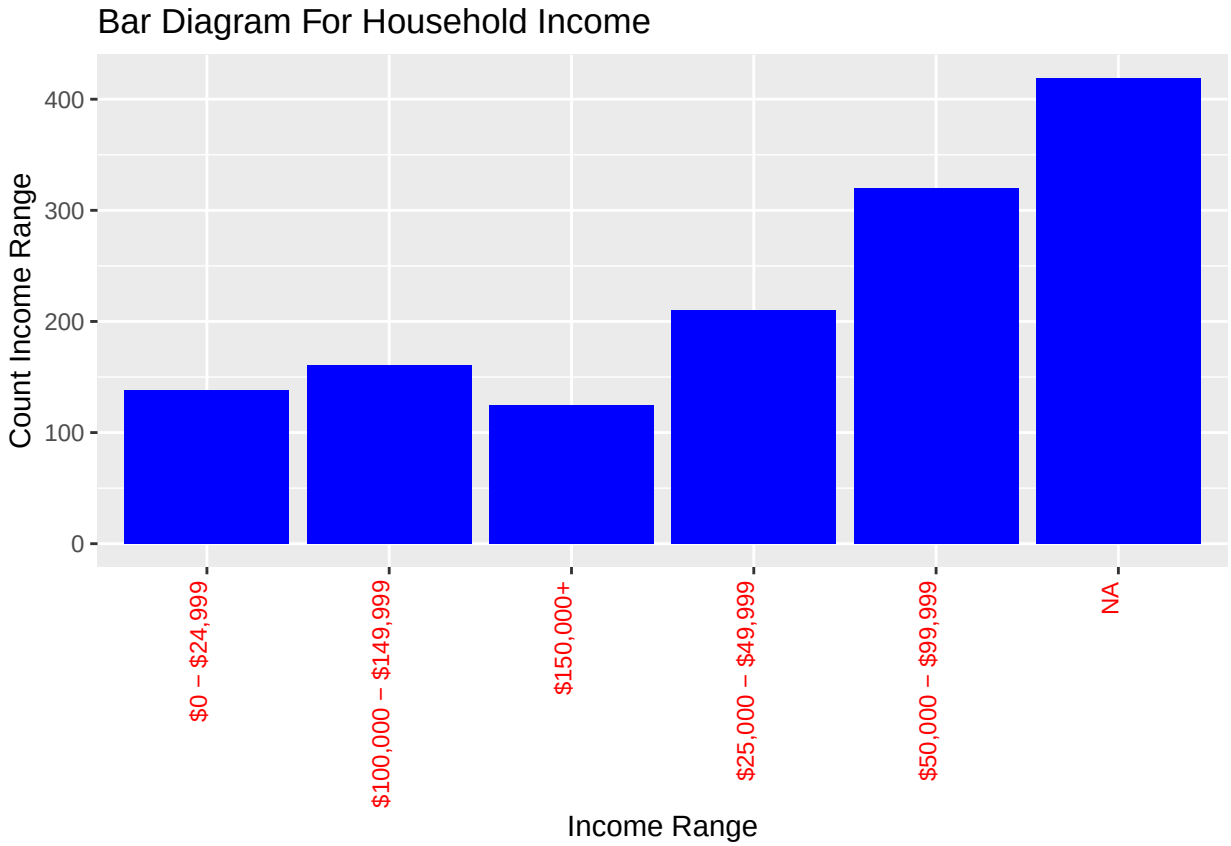
```
food_world_cup %>%
  ggplot(aes(x=factor(age)))+
  geom_bar(stat="count", fill = 'green')+
  xlab("Age Range")+
  ylab("No. Of People")+
  ggtitle("Bar Diagram For Age")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'Red'))
```



Here we can see that age group '45-60' is leading and age group '>60' is second leading category and category '18-19' are 4th leading gropu. While this column also have 'NA' category. But we are not going to replace it because leading category are not reliable in the contex of data, so we wil treate 'NA' as sperate category until we do not arrives at any conclusion.

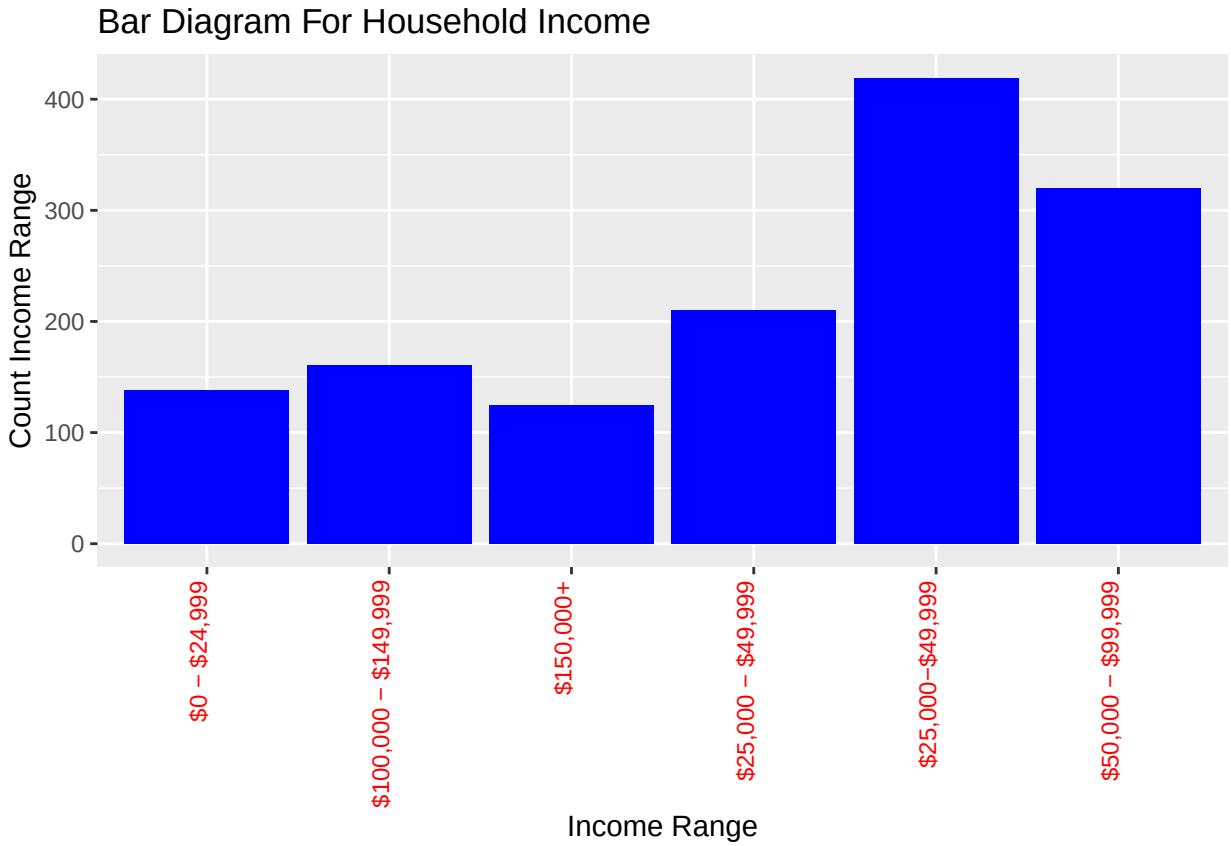
For column 'household_income'

```
food_world_cup %>%
  ggplot(aes(x=factor(household_income)))+
  geom_bar(stat="count", fill = 'blue')+
  xlab("Income Range")+
  ylab("Count Income Range")+
  ggtitle("Bar Diagram For Household Income")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'Red'))
```



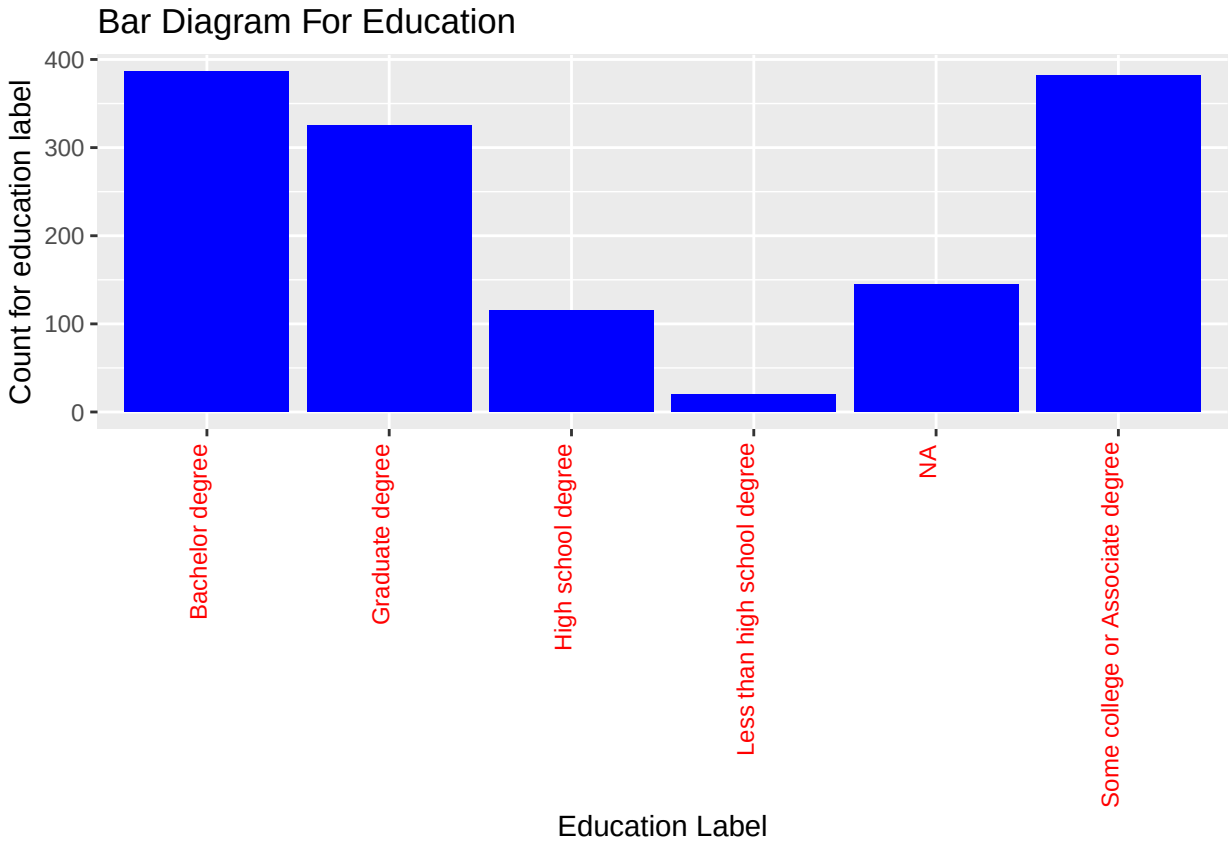
Here we can see that "NA" class is leading and highest second leading income class is just above the average income class and count in creamy-layer income class is low but not few. we will replace 'NA' class by class '$25,000-$49,999' as this makes the distribution of income in data symetric.

```
food_world_cup <- food_world_cup %>%
  mutate(household_income=ifelse(household_income=="NA", "$25,000-$49,999", household_income))

spark_write_csv(food_world_cup,"Sparkoutput",mode="overwrite")

food_world_cup <- spark_read_csv(sc,"food_world_cup","Sparkoutput", overwrite = TRUE)

food_world_cup %>%
  ggplot(aes(x=factor(household_income)))+
  geom_bar(stat="count", fill = 'blue')+
  xlab("Income Range")+
  ylab("Count Income Range")+
  ggtitle("Bar Diagram For Household Income")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'Red'))
```



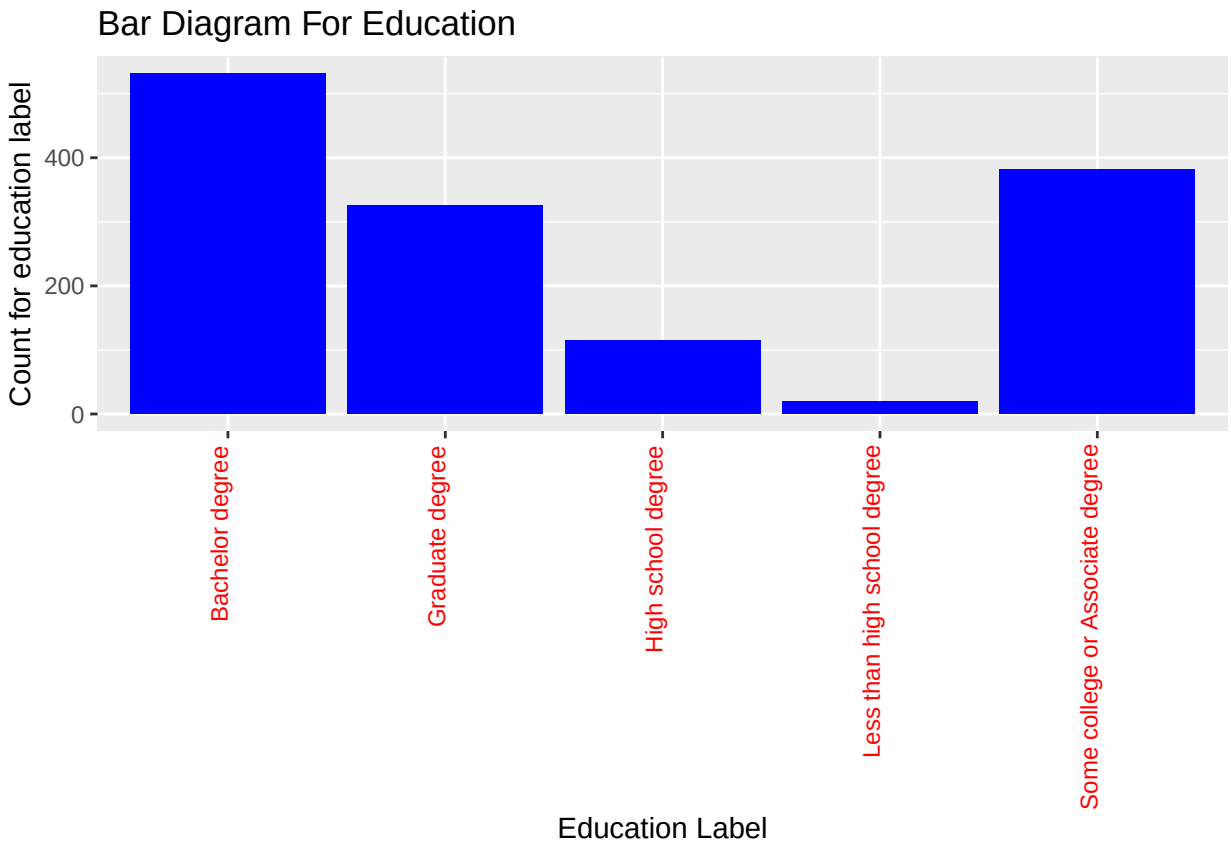
For column 'education'

```
food_world_cup %>%
  ggplot(aes(x=factor(education)))+
  geom_bar(stat="count", fill = 'blue')+
  xlab("Education Label")+
  ylab("Count for education label")+
  ggtitle("Bar Diagram For Education")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'Red'))
```



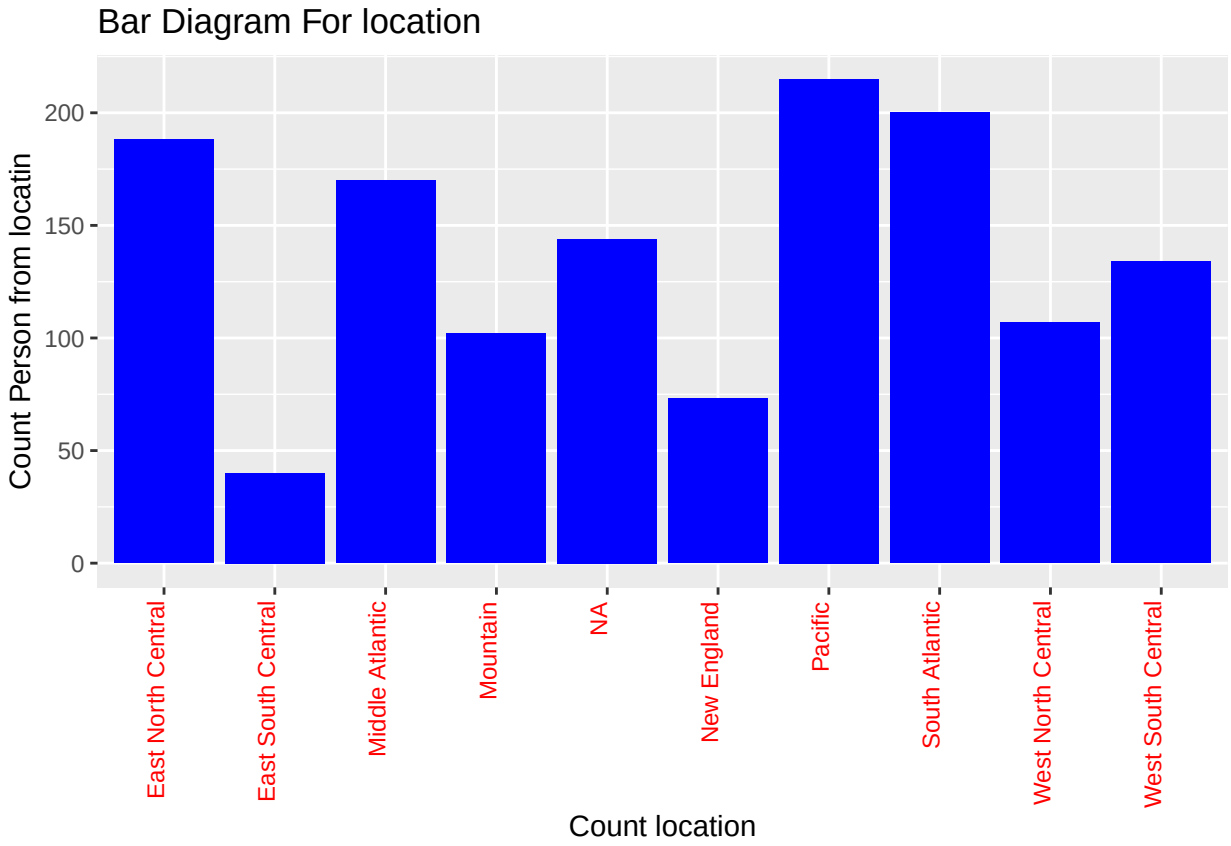
Here we can see that 'Bachelor degree' as expected are leading and 'Some college or associated degree'(others UG) are also in contest (second highest count); And count of 'Graduate degree' class are also highly close to leading one. Here also we can see that "NA" values are there and we will like to replace it by 'Bachelor degree' class. We can Note that count for lower education level is very low.

```
food_world_cup <- food_world_cup %>%
  mutate(education=ifelse(education=="NA", "Bachelor degree", education))

spark_write_csv(food_world_cup,"Sparkoutput",mode="overwrite")

food_world_cup <- spark_read_csv(sc,"food_world_cup","Sparkoutput", overwrite = TRUE)

food_world_cup %>%
  ggplot(aes(x=factor(education)))+
  geom_bar(stat="count", fill = 'blue')+
  xlab("Education Label")+
  ylab("Count for education label")+
  ggtitle("Bar Diagram For Education")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'Red'))
```



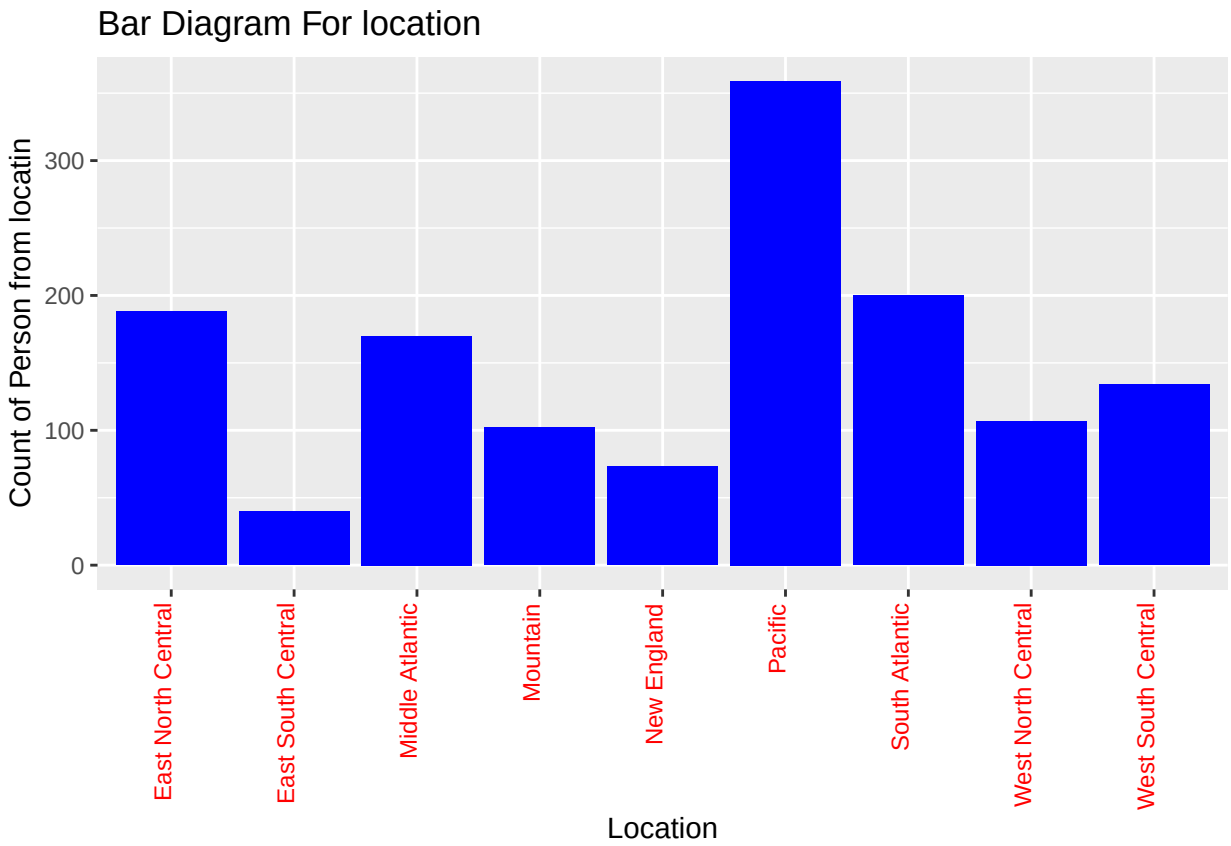
For column 'location'

```
food_world_cup %>%
  ggplot(aes(x=factor(location)))+
  geom_bar(stat="count", fill = 'blue')+
  xlab("Count location")+
  ylab("Count Person from locatin")+
  ggtitle("Bar Diagram For location")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'Red'))
```



Here we can see that participant from 'Pacific' is leading from the distribution we can see that participation of person from different rason is almost fair in nature. We see that class 'NA' is also there better replacement for them is that 'replace by distributing eually them to all others' but spark data frame does not allow this kind of replacement in form of direct syntex, so we like to replace them by leading class.

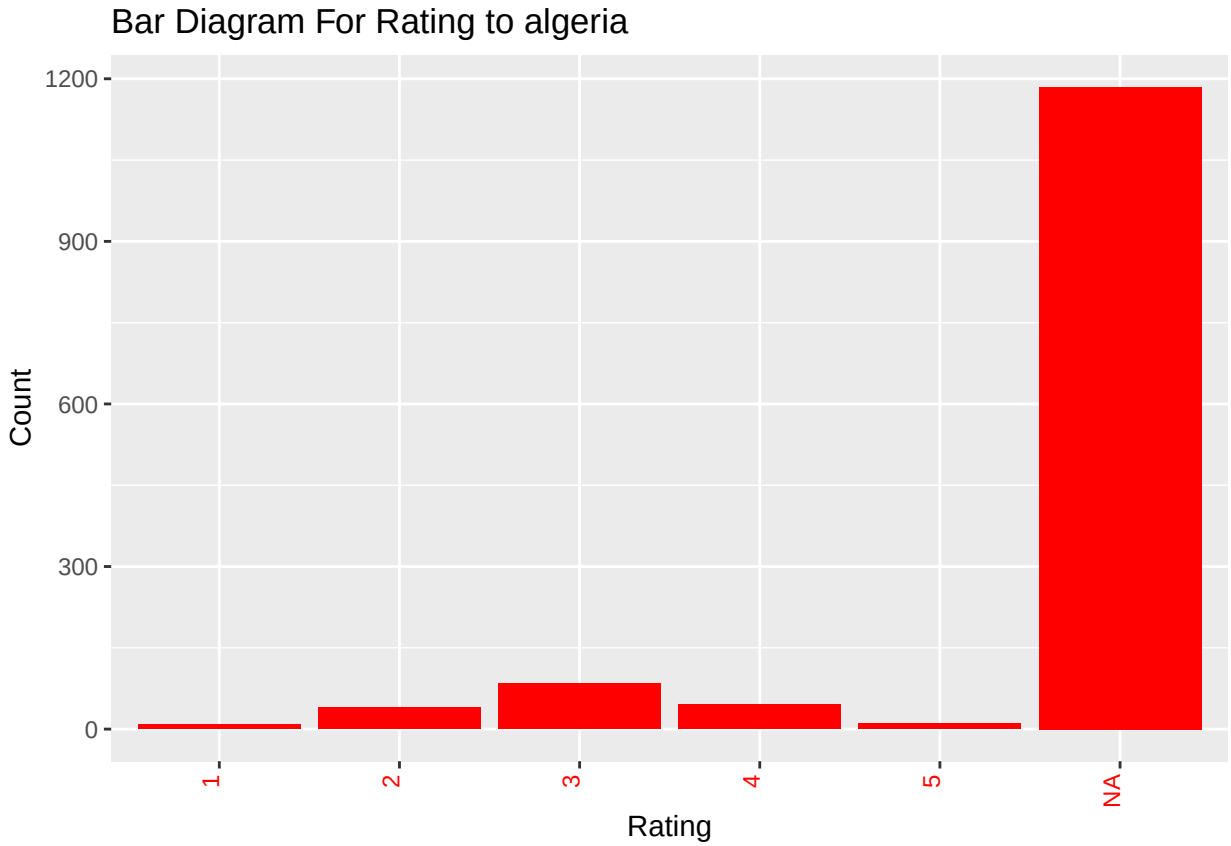
```
food_world_cup <- food_world_cup %>%
  mutate(location=ifelse(location=="NA", "Pacific", location))

spark_write_csv(food_world_cup,"Sparkoutput",mode="overwrite")

food_world_cup <- spark_read_csv(sc,"food_world_cup","Sparkoutput", overwrite = TRUE)

food_world_cup %>%
  ggplot(aes(x=factor(location)))+
  geom_bar(stat="count", fill = 'blue')+
  xlab("Location")+
  ylab("Count of Person from locatin")+
  ggtitle("Bar Diagram For location")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'Red'))
```



Missing value analysis for Rating to cuisines of different countries Here after class "NA" in contex of Rating have a meaning. From the data, participant are supposed to give the rating to dishes as 1, 2, 3 , 4, 5, and 'NA' where 1 means 'dissatisfied', 5 means 'highly satisfied' and 'NA' mean 'I don't know or don't want to commwnt'. And in order to perform mathematical operation we will just covert 'NA' to '0' as factor; so in contex of 'Rating' will not do any replacement of 'NA', well be treat them as seprate class as also it decleared in data set.

For Rating To dishes of country algeria

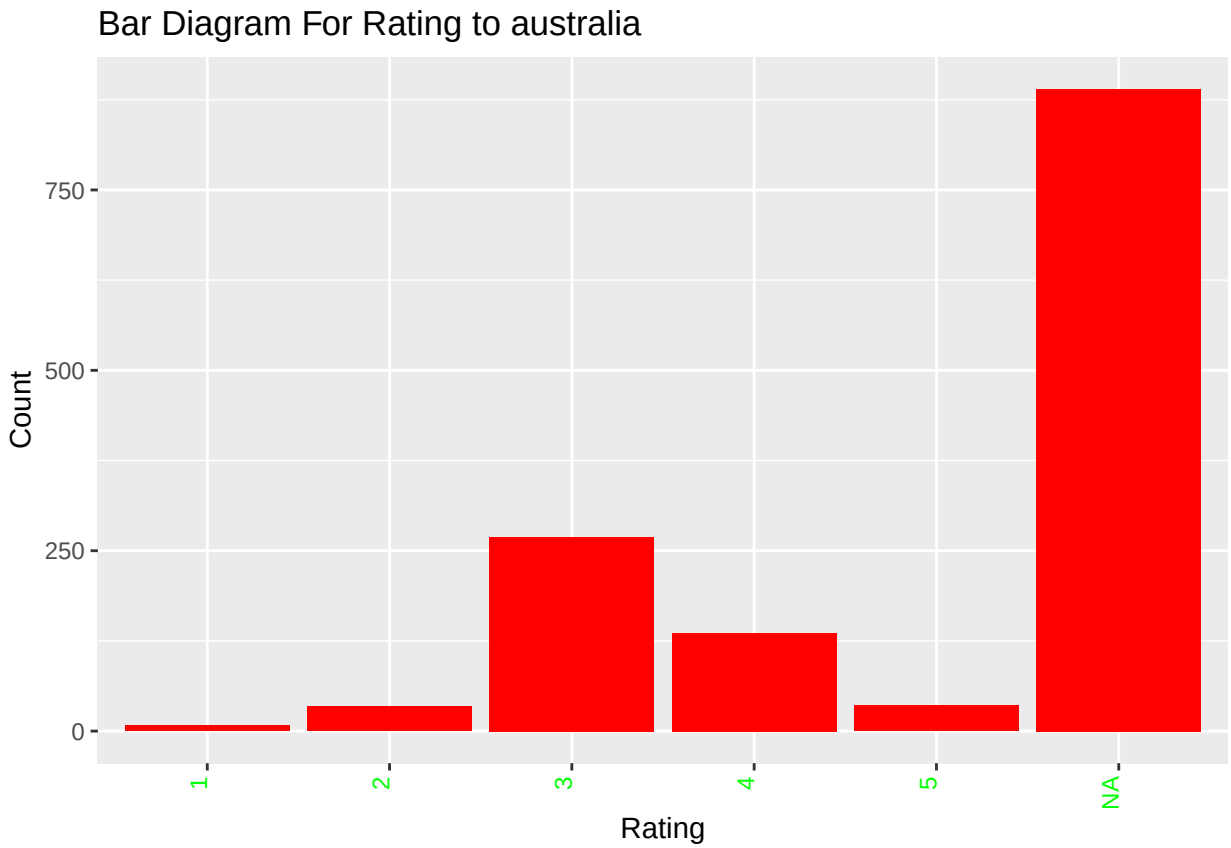
```
food_world_cup %>%
  ggplot(aes(x=factor(algeria)))+
  geom_bar(stat="count", fill = 'red')+
  xlab("Rating")+
  ylab("Count")+
  ggtitle("Bar Diagram For Rating to algeria")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'Red'))
```



Here we can see that among the Rating 3 is leading class symmetrically, and 'NA' is of highest count class.

For Rating To dishes of country australia

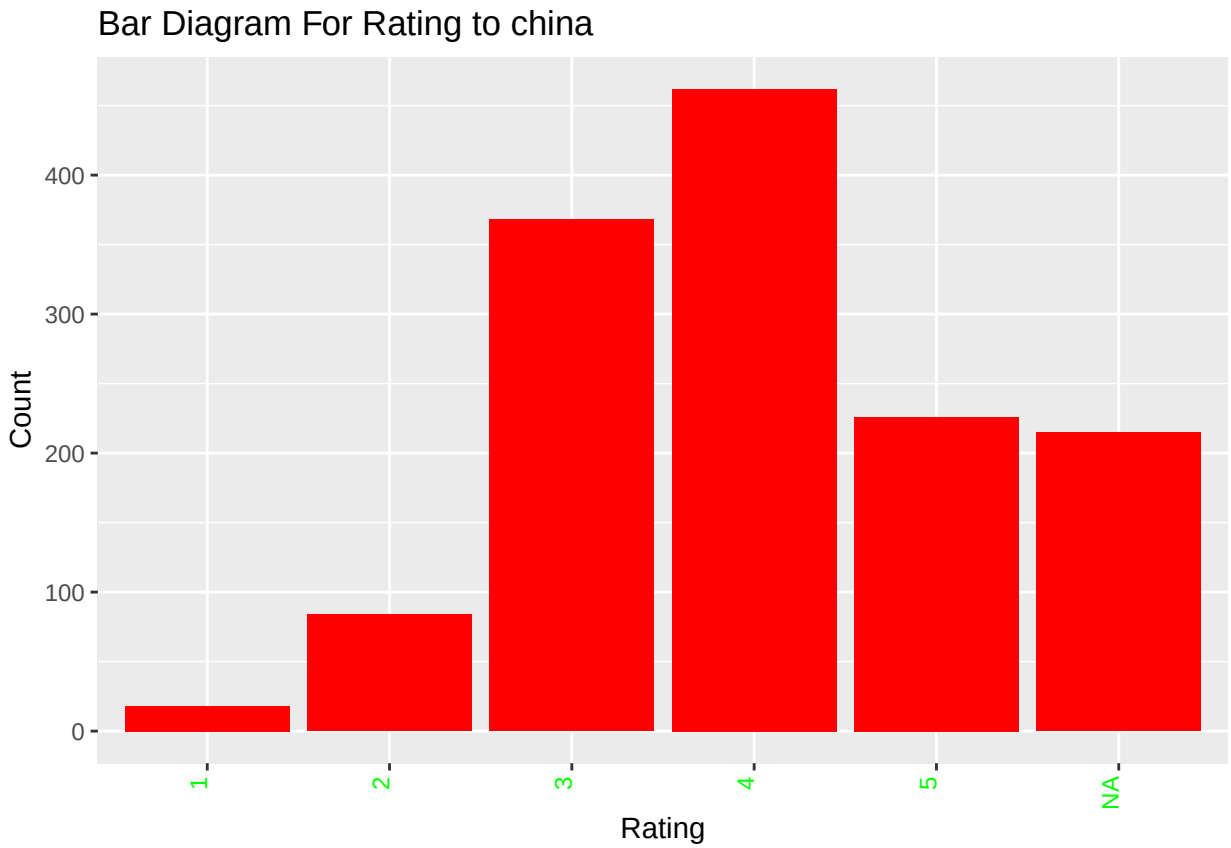
```
food_world_cup %>%
  ggplot(aes(x=factor(australia)))+
  geom_bar(stat="count", fill = 'red')+
  xlab("Rating")+
  ylab("Count")+
  ggtitle("Bar Diagram For Rating to australia")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'green'))
```



Here also we can see that among the Rating 3 is leading class, and 'NA' is of highest count class.

For Rating To dishes of country china

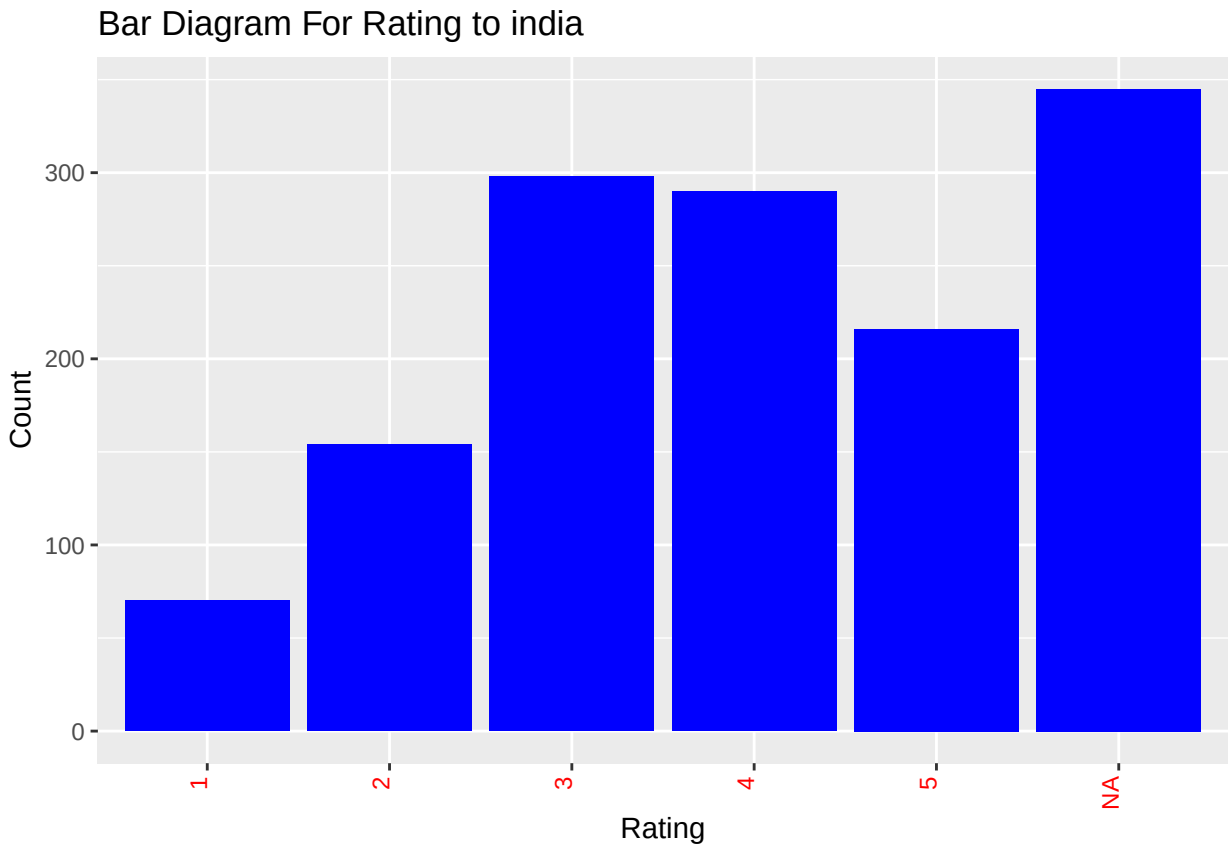
```
food_world_cup %>%
  ggplot(aes(x=factor(china)))+
  geom_bar(stat="count", fill = 'red')+
  xlab("Rating")+
  ylab("Count")+
  ggtitle("Bar Diagram For Rating to china")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'green'))
```



Here also we can see that among the Rating 4 is leading as expected and distribution is completely positively squed, and 'NA' has 3rd highest count. It tells that chainies food is famous to people throught the world.

For Rating To dishes of country india

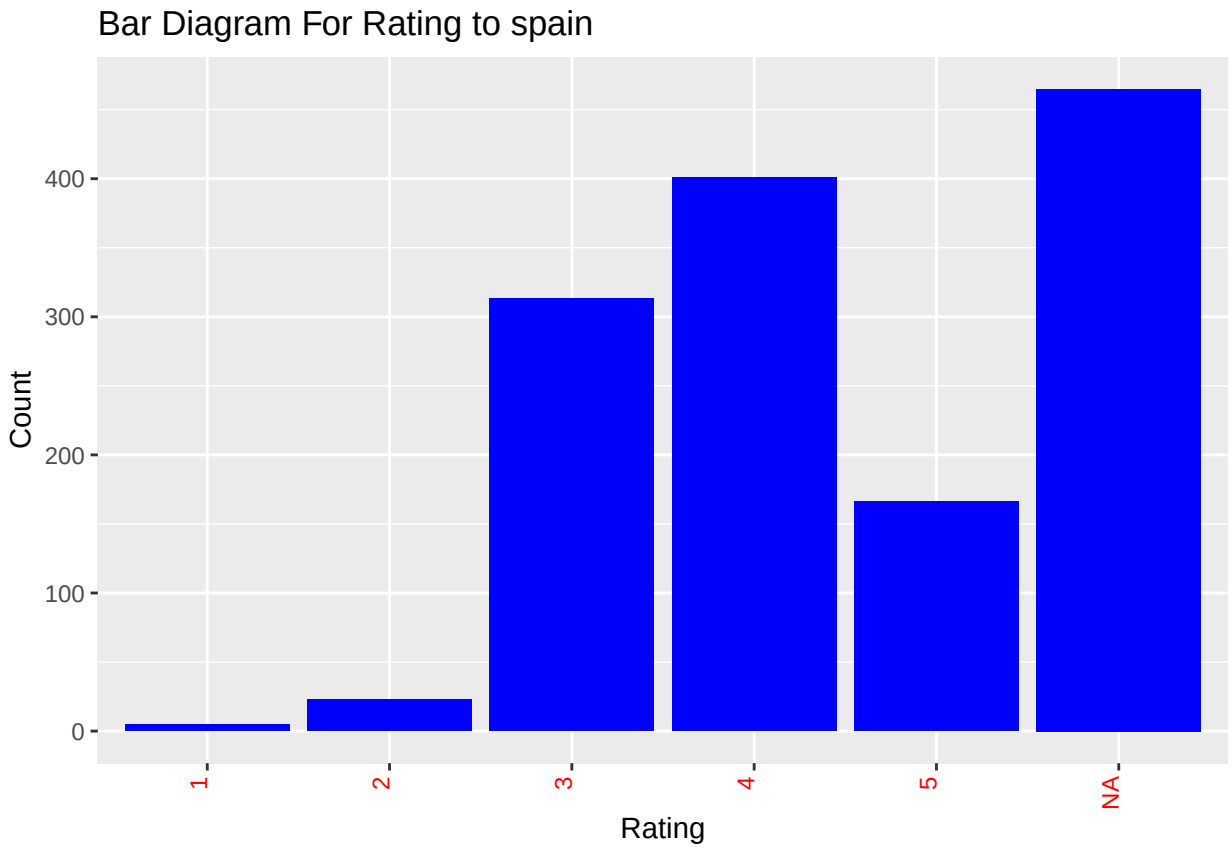
```
food_world_cup %>%
  ggplot(aes(x=factor(india)))+
  geom_bar(stat="count", fill = 'blue')+
  xlab("Rating")+
  ylab("Count")+
  ggtitle("Bar Diagram For Rating to india")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'red'))
```



Here also we can see that among the Rating 3 is leading as expected and clas 4 and 5 both are in contest and distribution is completely positively squed, and 'NA' has 3rd highest count. It tells that indian food is also liked by the people throught the world.

For Rating To dishes of country spain

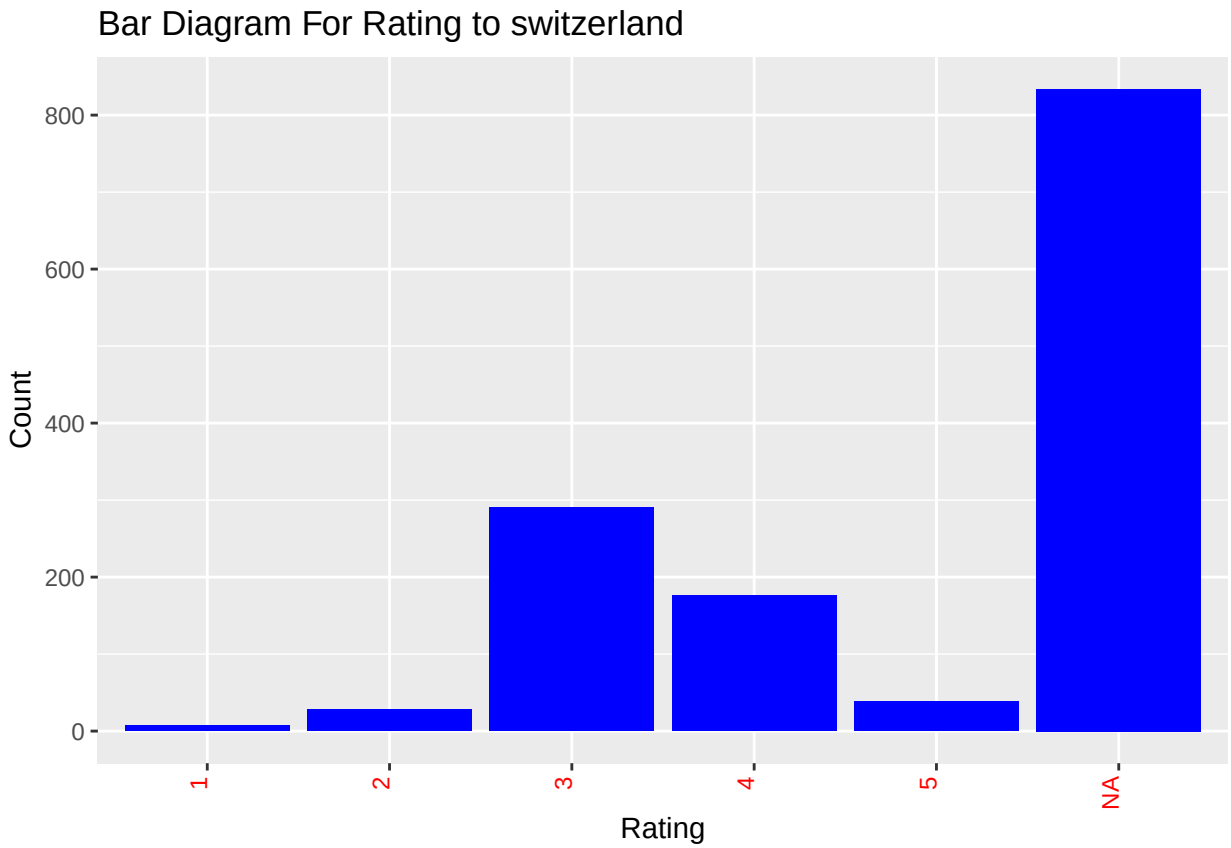
```
food_world_cup %>%
  ggplot(aes(x=factor(spain)))+
  geom_bar(stat="count", fill = 'blue')+
  xlab("Rating")+
  ylab("Count")+
  ggtitle("Bar Diagram For Rating to spain")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'red'))
```



Here we can see that clas 4 is leading then class 3 and distribution highly positevily.

For Rating To dishes of country switzerland

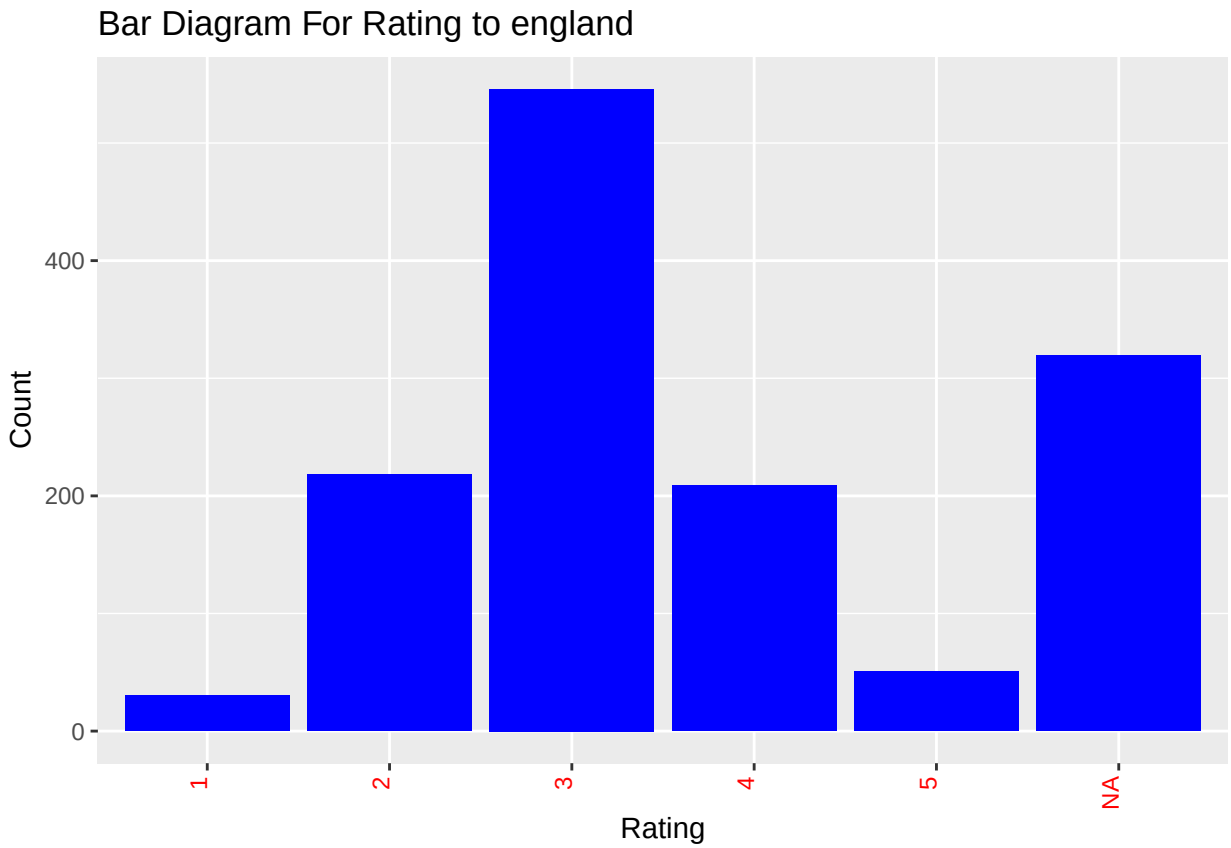
```
food_world_cup %>%
  ggplot(aes(x=factor(switzerland)))+
  geom_bar(stat="count", fill = 'blue')+
  xlab("Rating")+
  ylab("Count")+
  ggtitle("Bar Diagram For Rating to switzerland")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'red'))
```



Here we can see that clas 3 is leading as expected and distribution positevily. But clas 'NA' has highest count.

For Rating To dishes of country england

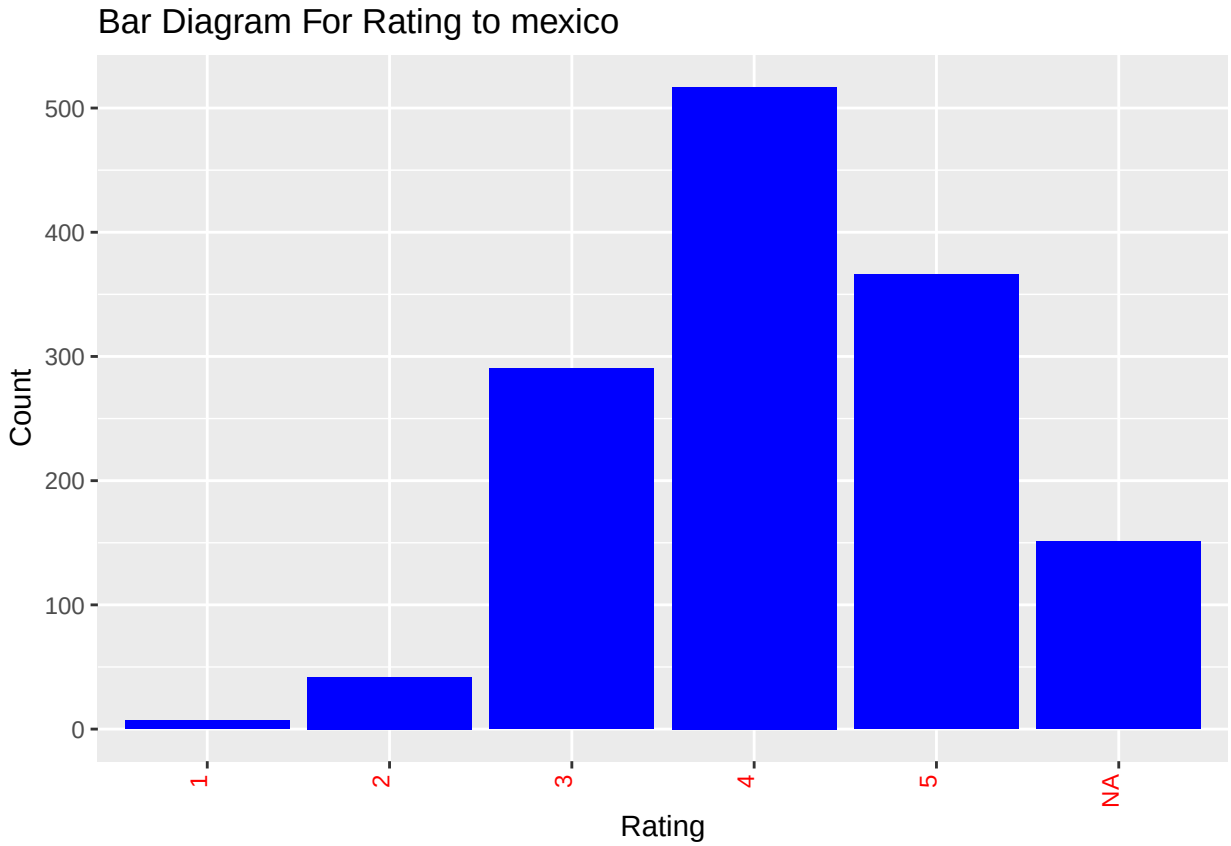
```
food_world_cup %>%
  ggplot(aes(x=factor(england)))+
  geom_bar(stat="count", fill = 'blue')+
  xlab("Rating")+
  ylab("Count")+
  ggtitle("Bar Diagram For Rating to england")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'red'))
```



Here we can see that clas 3 is of highest count as expected. And clas 'NA' has second highest count.

For Rating To dishes of country mexico

```
food_world_cup %>%
  ggplot(aes(x=factor(mexico)))+
  geom_bar(stat="count", fill = 'blue')+
  xlab("Rating")+
  ylab("Count")+
  ggtitle("Bar Diagram For Rating to mexico")+
  theme(axis.text.x = element_text(angle = 90,hjust= 0.99,vjust = 0.0,color = 'red'))
```



Here we can see that clas 4 is of highest count and distribution is highly positively squed. And clas 4 has second highest count.

Writting these changes to spark connection

```
food_world_cup <- food_world_cup %>%
  mutate(algeria=ifelse(algeria=="NA",0,algeria))%>%
  mutate(argentina=ifelse(argentina=="NA",0,argentina))%>%
  mutate(australia=ifelse(australia=="NA",0,australia))%>%
  mutate(belgium=ifelse(belgium=="NA",0,belgium))%>%
  mutate(bosnia_and_herzegovina=ifelse(bosnia_and_herzegovina=="NA",0,bosnia_and_herzegovina))%>%
  mutate(brazil=ifelse(brazil=="NA",0,brazil))%>%
  mutate(cameroon=ifelse(cameroon=="NA",0,cameroon))%>%
  mutate(chile=ifelse(chile=="NA",0,chile))%>%
  mutate(china=ifelse(china=="NA",0,china))%>%
  mutate(colombia=ifelse(colombia=="NA",0,colombia))%>%
```

```
  mutate(costa_rica=ifelse(costa_rica=="NA",0,costa_rica))%>%
  mutate(croatia=ifelse(croatia=="NA",0,croatia))%>%
  mutate(cuba=ifelse(cuba=="NA",0,cuba))%>%
  mutate(ecuador=ifelse(ecuador=="NA",0,ecuador))%>%
  mutate(england=ifelse(england=="NA",0,england))%>%
  mutate(ethiopia=ifelse(ethiopia=="NA",0,ethiopia))%>%
  mutate(france=ifelse(france=="NA",0,france))

spark_write_csv(food_world_cup,"Sparkoutput",mode="overwrite")
food_world_cup <- spark_read_csv(sc,"food_world_cup","Sparkoutput", overwrite = TRUE)
food_world_cup <- food_world_cup %>%
  mutate(germany=ifelse(germany=="NA",0,germany))%>%
  mutate(ghana=ifelse(ghana=="NA",0,ghana))%>%
  mutate(greece=ifelse(greece=="NA",0,greece))%>%
  mutate(honduras=ifelse(honduras=="NA",0,honduras))%>%
  mutate(india=ifelse(india=="NA",0,india))%>%
  mutate(iran=ifelse(iran=="NA",0,iran))%>%
  mutate(ireland=ifelse(ireland=="NA",0,ireland))%>%
  mutate(italy=ifelse(italy=="NA",0,italy))
spark_write_csv(food_world_cup,"Sparkoutput",mode="overwrite")
food_world_cup <- spark_read_csv(sc,"food_world_cup","Sparkoutput", overwrite = TRUE)
food_world_cup <- food_world_cup %>%
  mutate(ivory_coast=ifelse(ivory_coast=="NA",0,ivory_coast))%>%
  mutate(japan=ifelse(japan=="NA",0,japan))%>%
  mutate(mexico=ifelse(mexico=="NA",0,mexico))%>%
  mutate(nigeria=ifelse(nigeria=="NA",0,nigeria))%>%
  mutate(portugal=ifelse(portugal=="NA",0,portugal))%>%
  mutate(russia=ifelse(russia=="NA",0,russia))%>%
```

```
  mutate(south_korea=ifelse(south_korea=="NA",0,south_korea))%>%

  mutate(spain=ifelse(spain=="NA",0,spain))%>%

  mutate(switzerland=ifelse(switzerland=="NA",0,switzerland))%>%

  mutate(thailand=ifelse(thailand=="NA",0,thailand))%>%

  mutate(the_netherlands=ifelse(the_netherlands=="NA",0,the_netherlands))%>%

  mutate(turkey=ifelse(turkey=="NA",0,turkey))%>%

  mutate(united_states=ifelse(united_states=="NA",0,united_states))%>%

  mutate(uruguay=ifelse(uruguay=="NA",0,uruguay))%>%

  mutate(vietnam=ifelse(vietnam=="NA",0,vietnam))

spark_write_csv(food_world_cup,"Sparkoutput",mode="overwrite")

food_world_cup <- spark_read_csv(sc,"food_world_cup","Sparkoutput", overwrite = TRUE)

food_world_cup
```

```
## # Source: spark<food_world_cup> [?? x 48]
##    respondent_id knowledge interest gender age   household_income education
##            <dbl> <chr>     <chr>    <chr>  <chr> <chr>            <chr>
##  1    3308895255 Intermed~ Some     Male   18-29 $100,000 - $149~ Less tha~
##  2    3308891308 Novice    Some     Male   18-29 $100,000 - $149~ Some col~
##  3    3308891135 Intermed~ A lot    Male   30-44 $50,000 - $99,9~ Graduate~
##  4    3308879091 Novice    Not much Male   45-60 $0 - $24,999     Less tha~
##  5    3308871671 Novice    Not much Male   30-44 $25,000 - $49,9~ High sch~
##  6    3308871406 Advanced  A lot    Female 30-44 $50,000 - $99,9~ Graduate~
##  7    3308866182 Novice    Some     Male   45-60 $25,000-$49,999  High sch~
##  8    3308857114 Advanced  A lot    Male   45-60 $0 - $24,999     Some col~
##  9    3308856510 Novice    Not much Female 30-44 $50,000 - $99,9~ Some col~
## 10    3308846915 Novice    Some     NA     NA    $25,000-$49,999  Bachelor~
## # ... with more rows, and 41 more variables: location <chr>,
## #   algeria <dbl>, argentina <dbl>, australia <dbl>, belgium <dbl>,
## #   bosnia_and_herzegovina <dbl>, brazil <dbl>, cameroon <dbl>,
## #   chile <dbl>, china <dbl>, colombia <dbl>, costa_rica <dbl>,
## #   croatia <dbl>, cuba <dbl>, ecuador <dbl>, england <dbl>,
## #   ethiopia <dbl>, france <dbl>, germany <dbl>, ghana <dbl>,
## #   greece <dbl>, honduras <dbl>, india <dbl>, iran <dbl>, ireland <dbl>,
## #   italy <dbl>, ivory_coast <dbl>, japan <dbl>, mexico <dbl>,
## #   nigeria <dbl>, portugal <dbl>, russia <dbl>, south_korea <dbl>,
## #   spain <dbl>, switzerland <dbl>, thailand <dbl>, the_netherlands <dbl>,
## #   turkey <dbl>, united_states <dbl>, uruguay <dbl>, vietnam <dbl>
```

Collecting Data To local system for descriptive plots

```
food_world_cup <- food_world_cup %>%
  collect()
food_world_cup
```

```
## # A tibble: 1,373 x 48
##    respondent_id knowledge interest gender age   household_income education
##            <dbl> <chr>     <chr>    <chr>  <chr> <chr>            <chr>
##  1    3308895255 Intermed~ Some     Male   18-29 $100,000 - $149~ Less tha~
##  2    3308891308 Novice    Some     Male   18-29 $100,000 - $149~ Some col~
##  3    3308891135 Intermed~ A lot    Male   30-44 $50,000 - $99,9~ Graduate~
##  4    3308879091 Novice    Not much Male   45-60 $0 - $24,999     Less tha~
##  5    3308871671 Novice    Not much Male   30-44 $25,000 - $49,9~ High sch~
##  6    3308871406 Advanced  A lot    Female 30-44 $50,000 - $99,9~ Graduate~
##  7    3308866182 Novice    Some     Male   45-60 $25,000-$49,999  High sch~
##  8    3308857114 Advanced  A lot    Male   45-60 $0 - $24,999     Some col~
##  9    3308856510 Novice    Not much Female 30-44 $50,000 - $99,9~ Some col~
## 10    3308846915 Novice    Some     NA     NA    $25,000-$49,999  Bachelor~
## # ... with 1,363 more rows, and 41 more variables: location <chr>,
## #   algeria <dbl>, argentina <dbl>, australia <dbl>, belgium <dbl>,
## #   bosnia_and_herzegovina <dbl>, brazil <dbl>, cameroon <dbl>,
## #   chile <dbl>, china <dbl>, colombia <dbl>, costa_rica <dbl>,
## #   croatia <dbl>, cuba <dbl>, ecuador <dbl>, england <dbl>,
## #   ethiopia <dbl>, france <dbl>, germany <dbl>, ghana <dbl>,
## #   greece <dbl>, honduras <dbl>, india <dbl>, iran <dbl>, ireland <dbl>,
## #   italy <dbl>, ivory_coast <dbl>, japan <dbl>, mexico <dbl>,
## #   nigeria <dbl>, portugal <dbl>, russia <dbl>, south_korea <dbl>,
## #   spain <dbl>, switzerland <dbl>, thailand <dbl>, the_netherlands <dbl>,
## #   turkey <dbl>, united_states <dbl>, uruguay <dbl>, vietnam <dbl>
```

Creating Plot theme

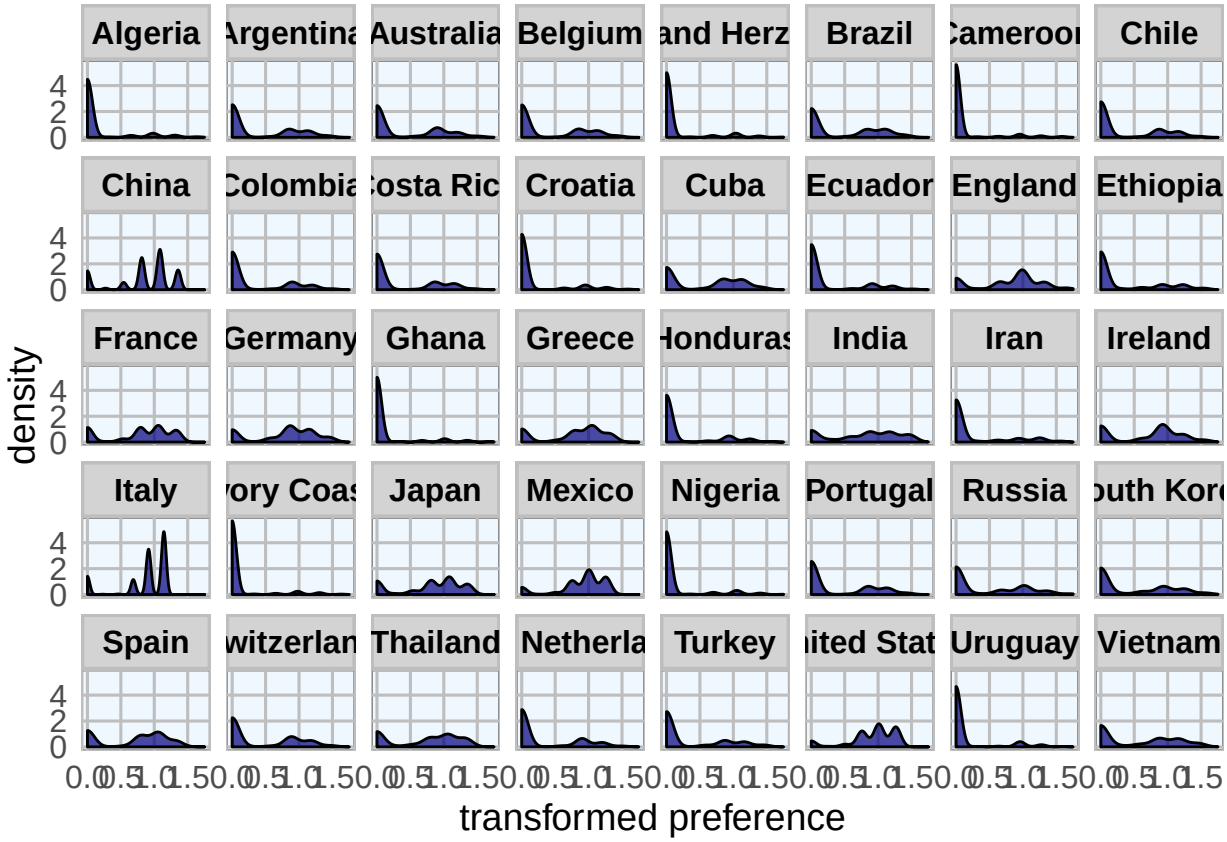
```
my_theme <- function(base_size = 12, base_family = "sans"){
  theme_minimal(base_size = base_size, base_family = base_family) +
    theme(
      axis.text = element_text(size = 12),
      axis.title = element_text(size = 14),
      panel.grid.major = element_line(color = "grey"),
      panel.grid.minor = element_blank(),
      panel.background = element_rect(fill = "aliceblue"),
      strip.background = element_rect(fill = "lightgrey", color = "grey", size = 1),
      strip.text = element_text(face = "bold", size = 12, color = "black"),
      legend.position = "right",
      legend.justification = "top",
      panel.border = element_rect(color = "grey", fill = NA, size = 0.5)
    )
}
```

Creating Bar Plot For "Male" and 'Female' Participatent

```
dataset_impute <- mice(food_world_cup[, -c(1, 2)],  print = FALSE)
```

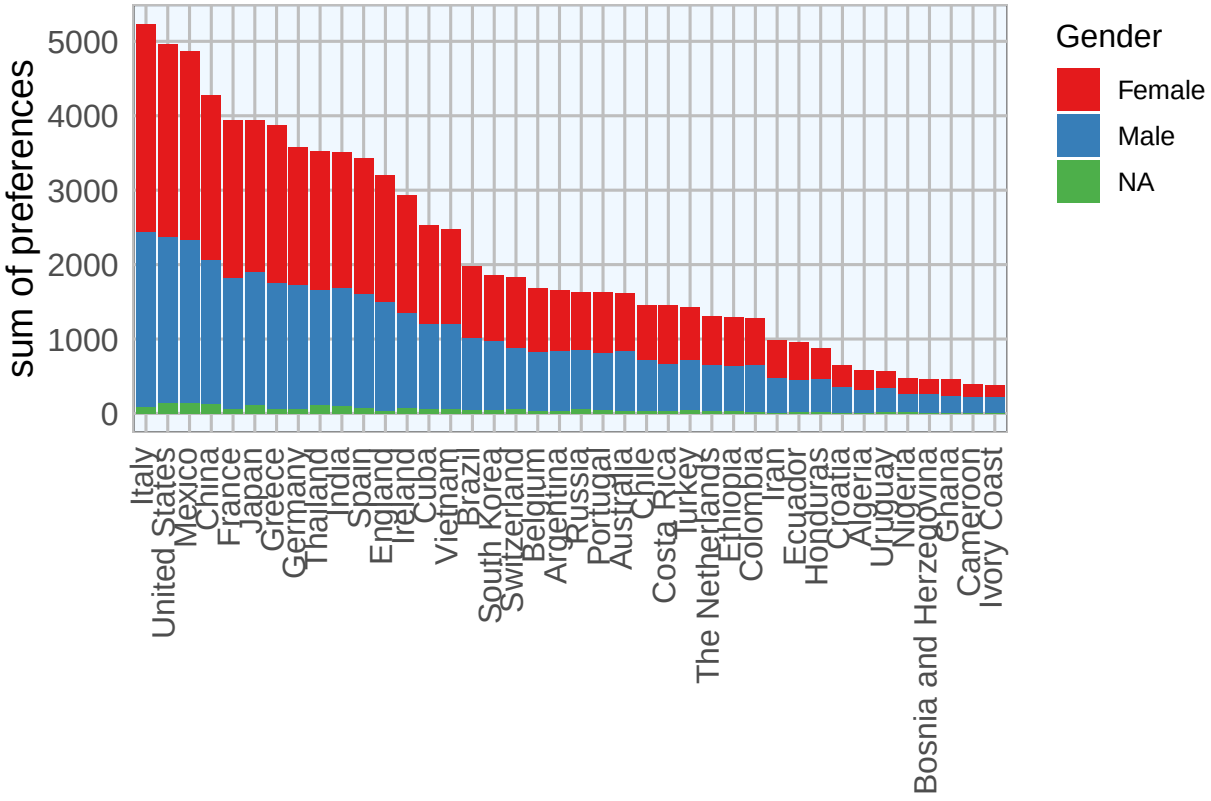
```
## Warning: Number of logged events: 6
```

```
food_world_cup <- cbind(food_world_cup[, 2, drop = FALSE], mice::complete(dataset_impute, 1))


food_world_cup[8:47] <- lapply(food_world_cup[8:47], as.numeric)

countries <- paste(colnames(food_world_cup)[-c(1:7)])

for (response in countries) {
  food_world_cup[paste(response, "trans", sep = "_")] <- food_world_cup[response] / mean(food_world_cup
}


food_world_cup %>%
  gather(x, y, algeria_trans:vietnam_trans) %>%
  mutate(x_2 = gsub("_trans", "", x)) %>%
  mutate(x_2 = gsub("_", " ", x_2)) %>%
  mutate(x_2 = gsub("(^|[[:space:]])([[:alpha:]])", "\\1\\U\\2", x_2, perl = TRUE)) %>%
  mutate(x_2 = gsub("And", "and", x_2)) %>%
  ggplot(aes(y)) +
  geom_density(fill = "navy", alpha = 0.7) +
  my_theme() +
  facet_wrap(~ x_2, ncol = 8) +
  labs(x = "transformed preference")
```



##

```
food_world_cup_gather <- food_world_cup %>%
  collect %>%
  gather(country, value, algeria:vietnam)

food_world_cup_gather$value <- as.numeric(food_world_cup_gather$value)
food_world_cup_gather$country <- as.factor(food_world_cup_gather$country)

food_world_cup_gather <- food_world_cup_gather %>%
  mutate(x_2 = gsub("_", " ", country)) %>%
  mutate(x_2 = gsub("(^|[[:space:]])([[:alpha:]])", "\\1\\U\\2", x_2, perl = TRUE)) %>%
  mutate(x_2 = gsub("And", "and", x_2))

order <- aggregate(food_world_cup_gather$value, by = list(food_world_cup_gather$x_2), FUN = sum)

food_world_cup_gather %>%
  mutate(x_2 = factor(x_2, levels = order$Group.1[order(order$x, decreasing = TRUE)])) %>%
  ggplot(aes(x = x_2, y = value, fill = gender)) +
  geom_bar(stat = "identity") +
  scale_fill_brewer(palette = "Set1") +
  my_theme() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.3, hjust = 1)) +
  labs(fill = "Gender",
       x = "",
       y = "sum of preferences")
```



Sending data to spark connection for ml related stuff

```
food_world_cup <- copy_to(sc, food_world_cup, 'food_world_cup', overwrite = TRUE)
food_world_cup
```

```
## # Source: spark<food_world_cup> [?? x 87]
##    knowledge interest gender age   household_income education location
##    <chr>     <chr>    <chr>  <chr> <chr>            <chr>     <chr>
##  1 Intermed~ Some     Male   18-29 $100,000 - $149~ Less tha~ West So~
##  2 Novice    Some     Male   18-29 $100,000 - $149~ Some col~ West So~
##  3 Intermed~ A lot    Male   30-44 $50,000 - $99,9~ Graduate~ Pacific
##  4 Novice    Not much Male   45-60 $0 - $24,999     Less tha~ New Eng~
##  5 Novice    Not much Male   30-44 $25,000 - $49,9~ High sch~ Pacific
##  6 Advanced  A lot    Female 30-44 $50,000 - $99,9~ Graduate~ East No~
##  7 Novice    Some     Male   45-60 $25,000-$49,999  High sch~ West So~
##  8 Advanced  A lot    Male   45-60 $0 - $24,999     Some col~ South A~
##  9 Novice    Not much Female 30-44 $50,000 - $99,9~ Some col~ South A~
## 10 Novice    Some     NA     NA    $25,000-$49,999  Bachelor~ Pacific
## # ... with more rows, and 80 more variables: algeria <dbl>,
## #   argentina <dbl>, australia <dbl>, belgium <dbl>,
## #   bosnia_and_herzegovina <dbl>, brazil <dbl>, cameroon <dbl>,
## #   chile <dbl>, china <dbl>, colombia <dbl>, costa_rica <dbl>,
## #   croatia <dbl>, cuba <dbl>, ecuador <dbl>, england <dbl>,
## #   ethiopia <dbl>, france <dbl>, germany <dbl>, ghana <dbl>,
## #   greece <dbl>, honduras <dbl>, india <dbl>, iran <dbl>, ireland <dbl>,
## #   italy <dbl>, ivory_coast <dbl>, japan <dbl>, mexico <dbl>,
```

```
## #   nigeria <dbl>, portugal <dbl>, russia <dbl>, south_korea <dbl>,
## #   spain <dbl>, switzerland <dbl>, thailand <dbl>, the_netherlands <dbl>,
## #   turkey <dbl>, united_states <dbl>, uruguay <dbl>, vietnam <dbl>,
## #   algeria_trans <dbl>, argentina_trans <dbl>, australia_trans <dbl>,
## #   belgium_trans <dbl>, bosnia_and_herzegovina_trans <dbl>,
## #   brazil_trans <dbl>, cameroon_trans <dbl>, chile_trans <dbl>,
## #   china_trans <dbl>, colombia_trans <dbl>, costa_rica_trans <dbl>,
## #   croatia_trans <dbl>, cuba_trans <dbl>, ecuador_trans <dbl>,
## #   england_trans <dbl>, ethiopia_trans <dbl>, france_trans <dbl>,
## #   germany_trans <dbl>, ghana_trans <dbl>, greece_trans <dbl>,
## #   honduras_trans <dbl>, india_trans <dbl>, iran_trans <dbl>,
## #   ireland_trans <dbl>, italy_trans <dbl>, ivory_coast_trans <dbl>,
## #   japan_trans <dbl>, mexico_trans <dbl>, nigeria_trans <dbl>,
## #   portugal_trans <dbl>, russia_trans <dbl>, south_korea_trans <dbl>,
## #   spain_trans <dbl>, switzerland_trans <dbl>, thailand_trans <dbl>,
## #   the_netherlands_trans <dbl>, turkey_trans <dbl>,
## #   united_states_trans <dbl>, uruguay_trans <dbl>, vietnam_trans <dbl>
```

Creating And adding pca columns for cluster analysis using ml_pca

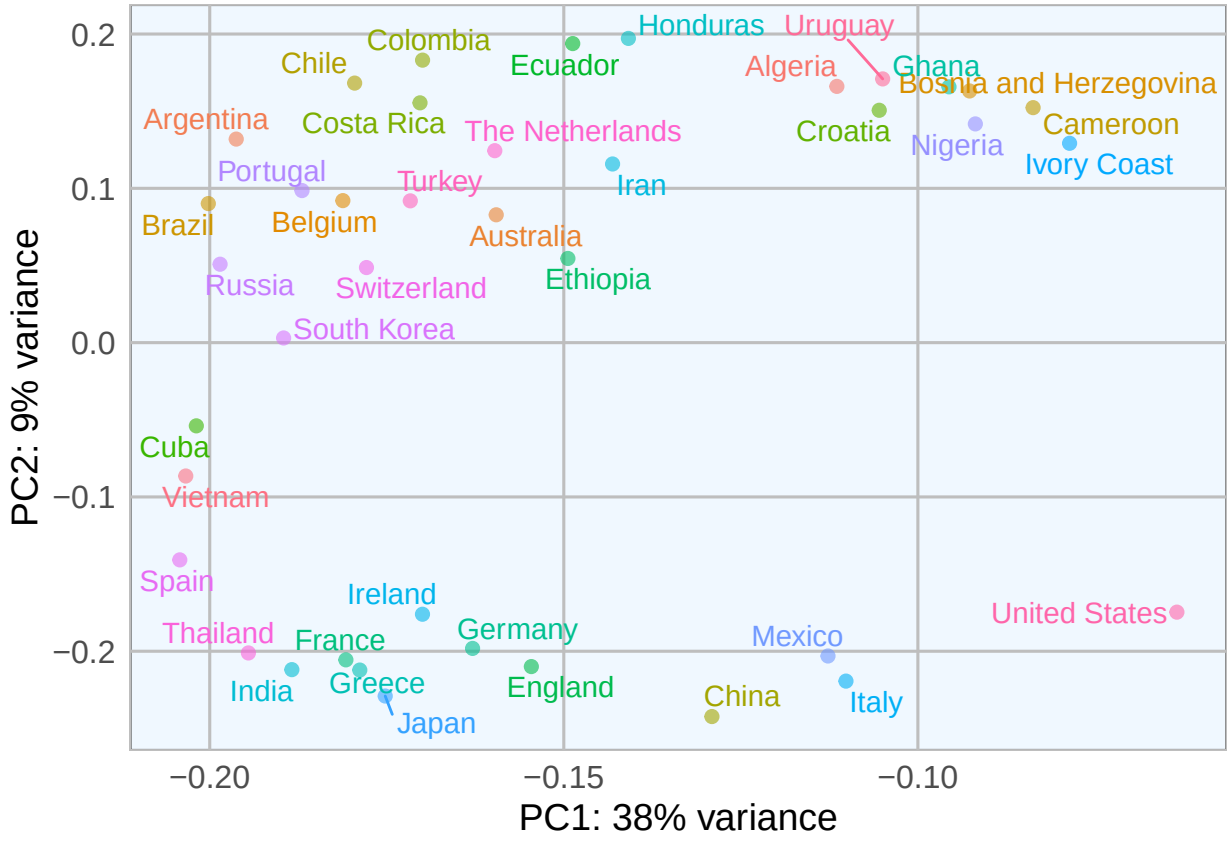
```
pca <- food_world_cup %>%
  mutate_each_(funs(as.numeric), countries) %>%
  ml_pca(features = paste(colnames(food_world_cup)[-c(1:47)]))
```

```
## Warning: mutate_each() is deprecated
## Please use mutate_if(), mutate_at(), or mutate_all() instead:
##
##   - To map `funs` over all variables, use mutate_all()
##   - To map `funs` over a selection of variables, use mutate_at()
## This warning is displayed once per session.

## Warning: funs() is soft deprecated as of dplyr 0.8.0
## please use list() instead
##
## # Before:
## funs(name = f(.)
##
## # After:
## list(name = ~f(.))
## This warning is displayed once per session.
```

Ploting Cluster Output using pca

```
as.data.frame(pca$pc) %>%
  rownames_to_column(var = "labels") %>%
  mutate(x_2 = gsub("_trans", "", labels)) %>%
  mutate(x_2 = gsub("_", " ", x_2)) %>%
  mutate(x_2 = gsub("(^|[[:space:]])([[:alpha:]])", "\\1\\U\\2", x_2, perl = TRUE)) %>%
  mutate(x_2 = gsub("And", "and", x_2)) %>%
```

```
  ggplot(aes(x = PC1, y = PC2, color = x_2, label = x_2)) +
  geom_point(size = 2, alpha = 0.6) +
  geom_text_repel() +
  labs(x = paste0("PC1: ", round(pca$explained_variance[1], digits = 2) * 100, "% variance"),
       y = paste0("PC2: ", round(pca$explained_variance[2], digits = 2) * 100, "% variance")) +
  my_theme() +
  guides(fill = FALSE, color = FALSE)
```



Adding index encoded columns using ft_string_indexer for columns : knowledge, interest, gender, age, household_income, education, location

```
food_world_cup <- tbl(sc, "food_world_cup") %>%
  ft_string_indexer(input_col = "knowledge", output_col = "knowledge_idx")%>%
  ft_string_indexer(input_col = "interest", output_col = "interest_idx") %>%
  ft_string_indexer(input_col = "gender", output_col = "gender_idx") %>%
  ft_string_indexer(input_col = "age", output_col = "age_idx") %>%
  ft_string_indexer(input_col = "household_income", output_col = "household_income_idx") %>%
  ft_string_indexer(input_col = "education", output_col = "education_idx") %>%
  ft_string_indexer(input_col = "location", output_col = "location_idx")
```

Adding normalized score column 'Generalized_Rating'

```
food_world_cup <- food_world_cup %>%
  mutate(Generalized_Rating=(algeria+argentina+australia+belgium+bosnia_and_herzegovina+brazil
                             +cameroon+chile+china+colombia+costa_rica+croatia+cuba+
                               ecuador+england+ethiopia+france+germany+ghana+greece+honduras+
                               india+iran+ireland+italy+ivory_coast+japan+mexico+nigeria+portugal+
                               russia+south_korea+spain+switzerland+thailand+the_netherlands+
                               turkey+united_states+uruguay+vietnam)/40)
```

Creating partation of data as train and test data

```
 partitioned <- food_world_cup%>%
   sdf_partition(training = 0.75, test = 0.25, seed = 123)
```

pipeline1 for linear regression using ml_linear_regression

crating the pipeline1

```
pipeline1 <- ml_pipeline(sc) %>%
  ft_dplyr_transformer(
    tbl =food_world_cup
  )%>%
  ft_r_formula(Generalized_Rating ~ interest_idx+ age_idx + household_income_idx + knowledge_idx + loca
  ml_linear_regression()

fitted_pipeline1 <- ml_fit(
  pipeline1,
  partitioned$training
)
```

Getting prediction and tally pipeline1 using ml_transform

```
predictions1 <- ml_transform(
  fitted_pipeline1,
  partitioned$test
)


predictions1<- predictions1%>%
  group_by(Generalized_Rating,prediction)%>%
  tally()
```

Checking Scores for prediction for pipeline1

```
ml_regression_evaluator(predictions1,label_col = "Generalized_Rating", predicted_lbl="predictions1",meti
```

```
## [1] 0.09371004
```

```
#best -->
ml_regression_evaluator(predictions1,label_col = "Generalized_Rating", predicted_lbl="predictions1",meti
```

```
## [1] 0.6980051
```

```
ml_regression_evaluator(predictions1,label_col = "Generalized_Rating", predicted_lbl="predictions1",meti
```

```
## [1] 0.8910534
```

```
ml_regression_evaluator(predictions1,label_col = "Generalized_Rating", predicted_lbl="predictions1",meti
```

```
## [1] 0.7939762
```

```
tbl1 <- tbl_df(predictions1)
```

Creating the pipeline2 for Decision tree classifier using ml_decision_tree_classifier

Fitting pipeline

```
pipeline3 <- ml_pipeline(sc) %>%
  ft_dplyr_transformer(
    tbl =food_world_cup
  )%>%
  ft_binarizer(
    input_col = "Generalized_Rating",
    output_col = "bucketed_rating",
    threshold = 1.67
  ) %>%
  ft_r_formula(bucketed_rating ~ interest_idx+ age_idx + household_income_idx + knowledge_idx + locatio
  ml_decision_tree_classifier(max_depth = 10, max_bins = 32, min_instances_per_node = 2)

fitted_pipeline3 <- ml_fit(
  pipeline3,
  partitioned$training
)
```

Getting prediction and score for pipeline2

```
predictions3 <- ml_transform(
  fitted_pipeline3,
  partitioned$test
)

predictions3 <- predictions3%>%
  group_by(bucketed_rating, prediction)%>%
  tally()

tbl3 <- tbl_df(predictions3)

ml_multiclass_classification_evaluator(predictions3,label_col ="bucketed_rating", predicted_lbl="predic
```

```
## [1] 0.5
```

```
#decision_tree_regressor
```

Creating And Fitting the pipeline3 for

```
pipeline4 <- ml_pipeline(sc) %>%
  ft_dplyr_transformer(
    tbl =food_world_cup
  )%>%

  ft_r_formula(Generalized_Rating ~ interest_idx+ age_idx + household_income_idx + knowledge_idx + loca
  ml_decision_tree_regressor(max_depth = 5,
                             max_bins = 32, min_instances_per_node = 1)

fitted_pipeline4 <- ml_fit(
  pipeline4,
  partitioned$training
)
```

Getting prediction and score for pipeline3

```
predictions4 <- ml_transform(
  fitted_pipeline4,
  partitioned$test
)

predictions4 <- predictions4%>%
  group_by(Generalized_Rating, prediction)%>%
  tally()

tbl4 <- tbl_df(predictions4)

#best -->
ml_regression_evaluator(predictions4,label_col = "Generalized_Rating", predicted_lbl="predictions4",met
```

```
## [1] 0.3246866
```

```
ml_regression_evaluator(predictions4,label_col = "Generalized_Rating", predicted_lbl="predictions4",meti
```

```
## [1] 0.5603684
```

```
ml_regression_evaluator(predictions4,label_col = "Generalized_Rating", predicted_lbl="predictions4",meti
```

```
## [1] 0.7923215
```

```
ml_regression_evaluator(predictions4,label_col = "Generalized_Rating", predicted_lbl="predictions4",meti
```

```
## [1] 0.6277733
```

Fitting the pipeline4

```
pipeline5 <- ml_pipeline(sc) %>%
  ft_dplyr_transformer(
    tbl =food_world_cup
  )%>%
  ft_binarizer(
    input_col = "Generalized_Rating",
    output_col = "bucketed_rating",
    threshold = 1.67
  ) %>%
  ft_r_formula(bucketed_rating ~ interest_idx+ age_idx + household_income_idx + knowledge_idx + locatio
  ml_logistic_regression()

fitted_pipeline5 <- ml_fit(
  pipeline5,
  partitioned$training
)

fitted_pipeline5
```

```
## PipelineModel (Transformer) with 4 stages
## <pipeline_352985ce9c6>
##   Stages
##   |--1 SQLTransformer (Transformer)
##   |    <dplyr_transformer_35297da34e5a>
##   |     (Parameters -- Column Names)
##   |--2 Binarizer (Transformer)
##   |    <binarizer_35296339213b>
##   |     (Parameters -- Column Names)
##   |      input_col: Generalized_Rating
##   |      output_col: bucketed_rating
##   |--3 RFormulaModel (Transformer)
##   |    <r_formula_35296814e96>
##   |     (Parameters -- Column Names)
```

```
##   |      features_col: features
##   |      label_col: label
##   |     (Transformer Info)
##   |      formula:  chr "bucketed_rating ~ interest_idx + age_idx + household_income_idx + knowledge_
##   |--4 LogisticRegressionModel (Transformer)
##   |    <logistic_regression_3529ffdd0ba>
##   |     (Parameters -- Column Names)
##   |      features_col: features
##   |      label_col: label
##   |      prediction_col: prediction
##   |      probability_col: probability
##   |      raw_prediction_col: rawPrediction
##   |     (Transformer Info)
##   |      coefficients:  num [1:5] -0.3355 -0.2877 0.1341 0.2793 -0.0184
##   |      intercept:  num -0.0562
##   |      num_classes:  int 2
##   |      num_features:  int 5
##   |      threshold:  num 0.5
```

```
predictions5 <- ml_transform(
  fitted_pipeline5,
  partitioned$test
)

predictions5
```

```
## # Source: spark<?> [?? x 101]
##    knowledge interest gender age   household_income education location
##    <chr>     <chr>    <chr>  <chr> <chr>            <chr>     <chr>
##  1 Advanced  A lot    Female 18-29 $150,000+        Bachelor~ Mountain
##  2 Advanced  A lot    Female 18-29 $25,000 - $49,9~ Bachelor~ Middle ~
##  3 Advanced  A lot    Female 18-29 $25,000 - $49,9~ Bachelor~ New Eng~
##  4 Advanced  A lot    Female 18-29 $25,000-$49,999  Bachelor~ Pacific
##  5 Advanced  A lot    Female 18-29 $25,000-$49,999  Some col~ Middle ~
##  6 Advanced  A lot    Female 30-44 $100,000 - $149~ Bachelor~ Middle ~
##  7 Advanced  A lot    Female 30-44 $25,000 - $49,9~ Graduate~ West No~
##  8 Advanced  A lot    Female 30-44 $25,000-$49,999  Some col~ Mountain
##  9 Advanced  A lot    Female 30-44 $50,000 - $99,9~ Bachelor~ Middle ~
## 10 Advanced  A lot    Female 30-44 $50,000 - $99,9~ Graduate~ East No~
## # ... with more rows, and 94 more variables: algeria <dbl>,
## #   argentina <dbl>, australia <dbl>, belgium <dbl>,
## #   bosnia_and_herzegovina <dbl>, brazil <dbl>, cameroon <dbl>,
## #   chile <dbl>, china <dbl>, colombia <dbl>, costa_rica <dbl>,
## #   croatia <dbl>, cuba <dbl>, ecuador <dbl>, england <dbl>,
## #   ethiopia <dbl>, france <dbl>, germany <dbl>, ghana <dbl>,
## #   greece <dbl>, honduras <dbl>, india <dbl>, iran <dbl>, ireland <dbl>,
## #   italy <dbl>, ivory_coast <dbl>, japan <dbl>, mexico <dbl>,
## #   nigeria <dbl>, portugal <dbl>, russia <dbl>, south_korea <dbl>,
## #   spain <dbl>, switzerland <dbl>, thailand <dbl>, the_netherlands <dbl>,
## #   turkey <dbl>, united_states <dbl>, uruguay <dbl>, vietnam <dbl>,
## #   algeria_trans <dbl>, argentina_trans <dbl>, australia_trans <dbl>,
## #   belgium_trans <dbl>, bosnia_and_herzegovina_trans <dbl>,
## #   brazil_trans <dbl>, cameroon_trans <dbl>, chile_trans <dbl>,
## #   china_trans <dbl>, colombia_trans <dbl>, costa_rica_trans <dbl>,
```

```
## #   croatia_trans <dbl>, cuba_trans <dbl>, ecuador_trans <dbl>,
## #   england_trans <dbl>, ethiopia_trans <dbl>, france_trans <dbl>,
## #   germany_trans <dbl>, ghana_trans <dbl>, greece_trans <dbl>,
## #   honduras_trans <dbl>, india_trans <dbl>, iran_trans <dbl>,
## #   ireland_trans <dbl>, italy_trans <dbl>, ivory_coast_trans <dbl>,
## #   japan_trans <dbl>, mexico_trans <dbl>, nigeria_trans <dbl>,
## #   portugal_trans <dbl>, russia_trans <dbl>, south_korea_trans <dbl>,
## #   spain_trans <dbl>, switzerland_trans <dbl>, thailand_trans <dbl>,
## #   the_netherlands_trans <dbl>, turkey_trans <dbl>,
## #   united_states_trans <dbl>, uruguay_trans <dbl>, vietnam_trans <dbl>,
## #   knowledge_idx <dbl>, interest_idx <dbl>, gender_idx <dbl>,
## #   age_idx <dbl>, household_income_idx <dbl>, education_idx <dbl>,
## #   location_idx <dbl>, Generalized_Rating <dbl>, bucketed_rating <dbl>,
## #   features <list>, label <dbl>, rawPrediction <list>,
## #   probability <list>, prediction <dbl>
```

```
predictions5 <- predictions5%>%
  group_by(bucketed_rating, prediction)%>%
  tally()

tbl5 <- tbl_df(predictions5)

ml_multiclass_classification_evaluator(predictions5,label_col ="bucketed_rating", predicted_lbl="predic
```

```
## [1] 0.5
```