FML64060_ASSIGNMENT_1

NIHARIKA MATSA

2023-09-09

Dataset reference

I have downloaded the dataset from Kaggle which has both qualitative and quantitative data. Please find the dataset below - https://www.kaggle.com/datasets/pyatakov/india-agriculture-crop-production

using readxl package

As our dataset is in excel format, we need to use the library called “readxl” so that we can load our dataset.

```
library(readxl)
library(knitr)
```

Import the dataset

```
agri_data <- read_excel("India Agriculture Crop Production.xlsx")
agri_data_df <- data.frame(agri_data)
```

Display the data

```
### I just want to display all the columns in the dataset. Hence using
options().
options(tibble.width = Inf)

### In order to display the data in a table format, I have used kable.
kable(head(agri_data_df), format = "markdown")
```

State	District	Crop	Year	Season	Area	Area.Units	Production	Production.Units	Yield
Andaman and Nicobar Islands	NICOBARS	Arecanut	2001-02	Kharif	1254	Hectare	2061	Tonnes	1.643541
Andaman and Nicobar Islands	NICOBARS	Arecanut	2002-03	Whole Year	1258	Hectare	2083	Tonnes	1.655803
Andam	NICOBA	Areca	200	Who	12	Hectar	1525	Tonnes	1.2093

State	District	Crop	Year	Season	Area	Area.Units	Production	Production.Units	Yield
an and Nicobar Islands	RS	nut	3-04	le Year	61	e			58
Andaman and Nicobar Islands	NORTH AND MIDDLE ANDAMAN	Arecanut	2001-02	Kharif	3100	Hectare	5239	Tonnes	1.690000
Andaman and Nicobar Islands	SOUTH ANDAMANS	Arecanut	2002-03	Whole Year	3105	Hectare	5267	Tonnes	1.696296
Andaman and Nicobar Islands	SOUTH ANDAMANS	Arecanut	2003-04	Whole Year	3118	Hectare	5182	Tonnes	1.661963

Summary of the data

Summary displays the class and mode of each column if it is a qualitative data and it displays the min, max, mean, median, etc., if it is a quantitative data.

```
summary(agri_data_df)
```

```
##     State             District             Crop                 Year          
##  Length:345407      Length:345407      Length:345407      Length:345407     
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##     Season               Area          Area.Units          Production       
##  Length:345407      Min.   :      0   Length:345407      Min.   :0.000e+00  
##  Class :character   1st Qu.:     74   Class :character   1st Qu.:8.700e+01  
##  Mode  :character   Median :    532   Mode  :character   Median :7.170e+02  
##                     Mean   :  11670                      Mean   :9.584e+05  
##                     3rd Qu.:   4110                      3rd Qu.:7.176e+03  
##                     Max.   :8580100                      Max.   :1.598e+09  
##                     NA's   :33                           NA's   :4993       
##  Production.Units       Yield         
##  Length:345407      Min.   :    0.00  
##  Class :character   1st Qu.:    0.55  
##  Mode  :character   Median :    1.00  
##                     Mean   :   79.41  
##                     3rd Qu.:    2.47  
##                     Max.   :43958.33  
##                     NA's   :33        
```

Descriptive Statistics for the quantitative data from the dataset

Calculate Mean

Add all the values from column and divide it by total number of values.

```
### I see that there are few missing values in the Area, Yield and Production column.
colMeans(agri_data_df[,c('Area', 'Production','Yield')],na.rm = TRUE)

##         Area   Production        Yield 
##  11670.19126 958371.14866     79.40757 
```

Calculate Median

To find the median, first arrange the values in the ascending order and then pick the middle number. a. If the total count is odd then we can have one median value (i.e., Middle number) b. If the total count is even then we will have two middle numbers, in order to find median for them, we have to calculate mean for those two numbers and the result will be our median.

```
median(agri_data_df$Area, na.rm = TRUE)

## [1] 532

median(agri_data_df$Production, na.rm = TRUE)

## [1] 717

median(agri_data_df$Yield, na.rm = TRUE)

## [1] 1
```

Calculate Min value for Area, Production and Yield

```
min(agri_data_df$Area, na.rm = TRUE)

## [1] 0.004

min(agri_data_df$Production, na.rm = TRUE)

## [1] 0

min(agri_data_df$Yield, na.rm = TRUE)

## [1] 0
```

Calculate Max value for Area, Production and Yield

```
max(agri_data_df$Area, na.rm = TRUE)

## [1] 8580100

max(agri_data_df$Production, na.rm =TRUE)

## [1] 1597800000

max(agri_data_df$Yield, na.rm = TRUE)

## [1] 43958.33
```

Descriptive Statistics for the qualitative data from the dataset

Calculate Mode

To find mode, first arrange the values in the ascending order and find the response which occurs most frequently. Dataset can have no mode, one mode or more than one mode.

```
### Calculate the mode for State
mode_result <- as.data.frame(sort(table(agri_data_df$State), decreasing = 
TRUE))

### Rename columns for clarity
colnames(mode_result) <- c("State", "Mode")

### Display the result in table format
kable(mode_result, format = "markdown")
```

State	Mode
Uttar Pradesh	44781
Madhya Pradesh	29906
Karnataka	27493
Bihar	24697
Rajasthan	20363
Tamil Nadu	18525
Assam	18186
Maharashtra	17922
Andhra Pradesh	16363
Odisha	16153
Chhattisgarh	15285
Gujarat	14053
West Bengal	12596
Haryana	8305

State	Mode
Uttarakhand	6702
Nagaland	5676
Himachal Pradesh	5043
Jharkhand	5004
Kerala	4870
Telangana	4704
Jammu and Kashmir	4348
Arunachal Pradesh	4345
Meghalaya	4322
Punjab	4142
Manipur	3120
Tripura	2557
Mizoram	2112
Puducherry	1127
Sikkim	876
Andaman and Nicobar Islands	728
Goa	399
Dadra and Nagar Haveli	332
Delhi	203
Chandigarh	124
Daman and Diu	44
Laddakh	1

```r
### Calculate the mode for Season
season_result <- as.data.frame(sort(table(agri_data_df$Season), decreasing =
TRUE))

### Rename columns for clarity
colnames(season_result) <- c("Season", "Mode")

### Display the result in table format
kable(season_result, format = "markdown")
```

Season	Mode
Kharif	138400
Rabi	100977
Whole Year	68689
Summer	22101
Winter	8250

Season	Mode
Autumn	6989
nan	1

```r
### Calculate the mode for Crop
crop_result <- as.data.frame(sort(table(agri_data_df$Crop), decreasing =
TRUE))

### Rename columns for clarity
colnames(crop_result) <- c("Crop", "Mode")

### Display the result in table format
kable(crop_result, format = "markdown")
```

Crop	Mode
Rice	21611
Maize	20507
Moong(Green Gram)	15101
Urad	14581
Sesamum	13049
Groundnut	12586
Wheat	11248
Rapeseed &Mustard	11034
Sugarcane	10942
Arhar/Tur	10895
Potato	10756
Onion	10675
Gram	10474
Jowar	9769
Dry chillies	8971
Bajra	7796
Peas & beans (Pulses)	7266
Sunflower	7244
Small millets	6985
Cotton(lint)	6475
Masoor	6383
Turmeric	5953
Linseed	5892
Barley	5891
Ragi	5757

Crop	Mode
Sweet potato	5742
Other Kharif pulses	5720
Horse-gram	5424
Coriander	5037
Garlic	5032
Soyabean	4988
Other Rabi pulses	4866
Ginger	4686
Castor seed	4681
Banana	4509
Tobacco	3590
Sannhamp	3017
Coconut	2927
Niger seed	2792
Mesta	2406
Tapioca	2268
Arecanut	2192
Guar seed	2088
Jute	1913
Safflower	1764
Cowpea(Lobia)	1761
Khesari	1759
Cashewnut	1573
Black pepper	1417
Moth	1408
Other Cereals	1387
other oilseeds	1240
Oilseeds total	702
Cardamom	575
Other Summer Pulses	67
Dry Ginger	3

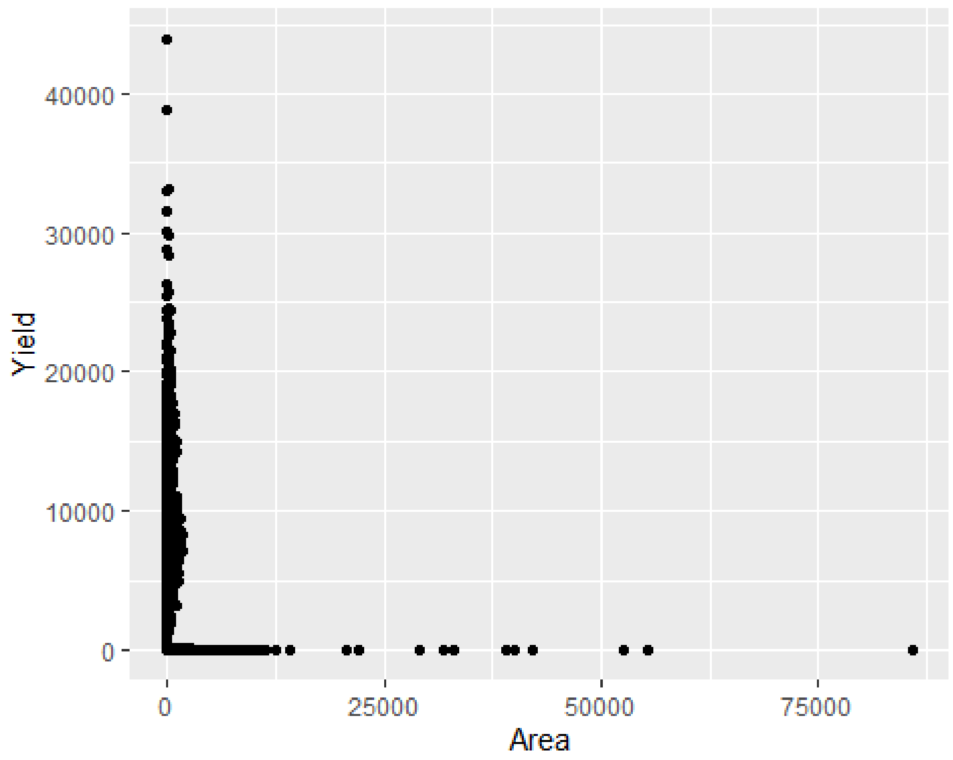
Transform atleast one variable

```r
### We need dplyr library to rename the column name
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

### Rename the columns
agri_data_df <- agri_data_df %>%
  rename(
    Production_Units = Production.Units,
    Area_Units = Area.Units
  )

### We are converting Hectares to Kilometers for Area_Units column
agri_data_df <- agri_data_df %>%
  mutate(
    Area = Area/100
  )

### Displays the number of rows in the dataset
nrow(agri_data_df)

## [1] 345407

### Assigning the Area Units value as Kilometers as we have converted Hectare
to Kilometers
agri_data_df$Area_Units <- 'Kilometers'

### Round off the values for Area, Production and Yield
agri_data_df$Yield <- round(agri_data_df$Yield, digits=2)
agri_data_df$Production <- round(agri_data_df$Production, digits=2)
agri_data_df$Area <- round(agri_data_df$Area, digits=2)
```

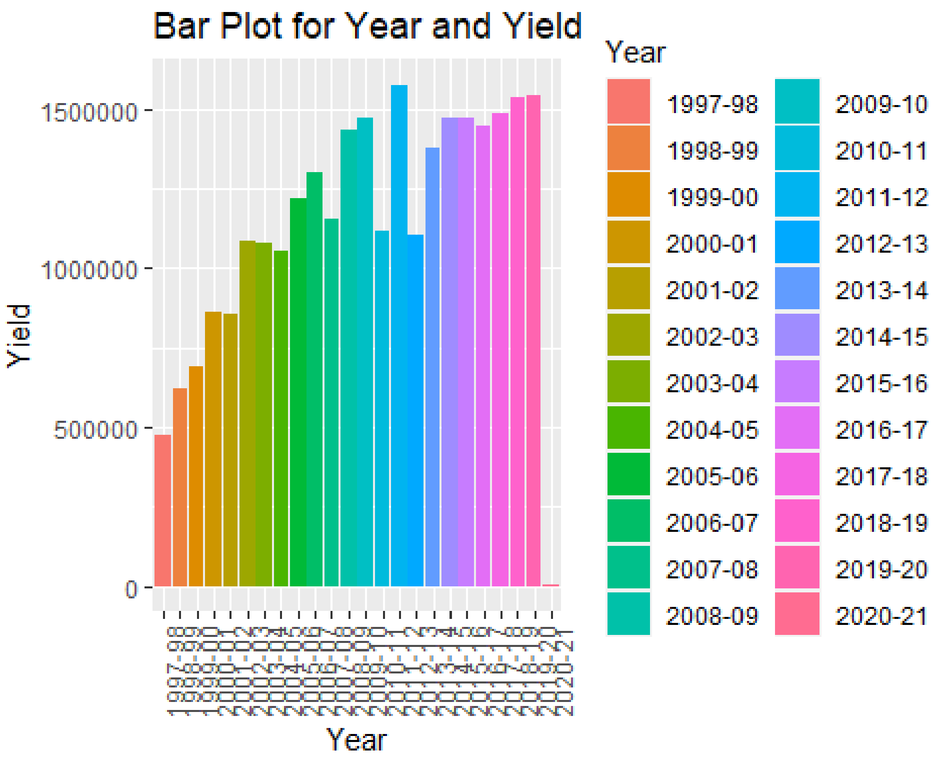
Scatter plot for Production and Yield

```
#### Load the ggplot2 package
library(ggplot2)
# Create a scatter plot
scatter_plot <- ggplot(agri_data_df, aes(x = Area, y = Yield)) +
  geom_point()

# Display the scatter plot
print(scatter_plot)

## Warning: Removed 33 rows containing missing values (`geom_point()`).
```



Bar Plot for Year and Yield

```
# Create a bar plot
colors <- c("red","green", "blue","purple", "orange")
ggplot(agri_data_df, aes(x = Year, y = Yield, fill= Year)) +
  geom_bar(stat = "identity") +
  labs(title = "Bar Plot for Year and Yield") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

## Warning: Removed 33 rows containing missing values (`position_stack()`).
```


Bar Plot for Year and Yield
Yield
1500000
1000000
500000
0
Year
1997-98
1998-99
1999-00
2000-01
2001-02
2002-03
2003-04
2004-05
2005-06
2006-07
2007-08
2008-09
2009-10
2010-11
2011-12
2012-13
2013-14
2014-15
2015-16
2016-17
2017-18
2018-19
2019-20
2020-21
Year
1997-98
1998-99
1999-00
2000-01
2001-02
2002-03
2003-04
2004-05
2005-06
2006-07
2007-08
2008-09
2009-10
2010-11
2011-12
2012-13
2013-14
2014-15
2015-16
2016-17
2017-18
2018-19
2019-20
2020-21