Software Documentation

1.Python(Use google colab to run the code)

1.Per person value of consumption of ceral and pulses in 30 days.ipynb

#Data used - Value (Rs.) of consumption of cereals and pulses per person.

#Load the dataset in google colab and run the following code.

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

from plotly.subplots import make_subplots

pip install --upgrade "kaleido==0.1.*"

from google.colab import files

df=pd.read_excel("/content/Value (Rs.) of consumption of cereals and pulses per person for a period of 30 days for each fractile class of MPCE.xlsx",index_col=[0,1])

def plot_all_class_bar(df,state):

 fig=make_subplots(rows=1,cols=2,subplot_titles=("Rural","Urban"))


 fig1=px.bar(df.loc["Rural"].sort_values(by="All Classes",ascending=True),

     y="Item description",x="All Classes",orientation="h",

     )


 fig1.update_traces(

    hovertemplate="<b>Value in Rs:</b> %{x}<br>"

                  "<b>Item description:</b> %{y}<br>"
```

```
)

for traces in fig1.data:
  fig.add_trace(traces,row=1,col=1)



fig2=px.bar(df.loc["Urban"].sort_values(by="All Classes",ascending=True),
     y="Item description",x="All Classes",orientation="h",
     )

fig2.update_traces(
   hovertemplate="<b>Value in Rs:</b> %{x}<br>"
                 "<b>Item description:</b> %{y}<br>"
)

for traces in fig2.data:
  fig.add_trace(traces,row=1,col=2)

fig.update_layout(title_text=f"Per person value of consumption of cereals and pulses in 30 days in {state}. ",width=1600,height=600)
fig.update_xaxes(title_text="Value in Rs",row=1,col=1)
fig.update_xaxes(title_text="Value in Rs",row=1,col=2)
fig.update_yaxes(title_text="Item description")
fig.show()

fig.write_image(f"{state}.png")
files.download(f"{state}.png")
dfs={}
state_names=df.index.get_level_values(0).unique()
for state in state_names:
```

```
 dfs[state]=df.loc[state]

import time

for (name,df) in dfs.items():

 plot_all_class_bar(df,name)

 time.sleep(3)
```

#This code should download 34 state wise plots.

2. Sex ratio sunburst chart.ipynb

```
#Data used- Estimated number of households and persons by gender and average MPCE for each fractile class of MPCE.

#Load the dataset in google collab and run the code

import pandas as pd

import plotly.express as px

df=pd.read_excel("/content/Estimated number of households and persons by gender and average MPCE for each fractile class of MPCE.xlsx",index_col=[0,1])

df["Total_Males"]=df.Adults_Male+df.Children_Male

df["Total_Females"]=df.Adults_Female+df.Children_Female

df.drop(index="All-India",inplace=True)

df=df[["Fractile class of MPCE","Total_Males","Total_Females"]]

df.reset_index(inplace=True) #multiindex to single index

df

df_long=df.melt(id_vars=["State/UT/All-India","Sector","Fractile class of MPCE"],var_name="Gender",value_name="Count")

df_long

df_long.Gender=df_long.Gender.map({"Total_Males":"Male","Total_Females":"Female"})

df_long

from google.colab import files

fig=px.sunburst(df_long,path=["State/UT/All-India","Sector","Fractile class of MPCE","Gender"],values="Count",branchvalues="total")

fig.update_traces(
```

```
    insidetextorientation='radial',
    textinfo='label+percent parent',
    marker=dict(line=dict(width=1)) ,

    hovertemplate="<b>%{label}</b><br>"
                  "Total people sampled: %{value}<br>",


    )
fig.update_layout(
    margin=dict(t=10, l=10, r=10, b=10),
    width=1900,
    height=800,
)

fig.write_html("sunburst.html",full_html=True)
files.download("sunburst.html")

#THis should generate and download a html file of sunburst chart
```

2. R

1. Average MPCE in rupees for fractile class

```
#Run the following code to generate state wise chart of avg mpce for different fractile class

df <- read.csv("D:\Users\Documents\MOSPI_hackathon\Ananda\1\2\data_adjusted.csv", check.names = FALSE)

dim(df)

str(df)

View(df)

df <- df[,-3]

View(df)
```

```
my_func <- function(state){

temp <- df[which(df$State/UT/All-India == state), -1]

library(tidyverse)

melted_temp <- temp %>% pivot_longer(cols = names(temp)[-1], names_to = "Family Size", values_to = "Avg. MPCE")

p <- melted_temp %>% ggplot(aes(x = Family Size, y = Avg. MPCE, fill = Sector)) + geom_col(position = "dodge") + labs(title = paste(state, ": Average MPCE in Rupees")) + theme(legend.position = "top")

ggsave(path = "D:\Users\Documents\MOSPI_hackathon\Ananda\1\2\", plot = p, width = 10, height = 8, device='png', dpi=1000, filename = paste(state, ".png", sep = ""), units = "in", bg = "white") }

for (i in unique(df$State/UT/All-India)) { my_func(i)
 }
```

2. Family size for different fractile class

```
#Run the following code  to generate statwise chart

data_1 <- read.csv("D:\\Users\\Documents\\MOSPI_hackathon\\Ananda\\1\\3\\data_adjusted.csv", check.names = FALSE)



dim(data_1)



names(data_1)



str(data_1)



data_1$`9` <- as.numeric(data_1$`9`)

data_1$`10+` <- as.numeric(data_1$`10+`)



str(data_1)



View(data_1)
```

```
any(is.na(data_1) == TRUE)


col_na <- c()


col_na <- apply(data_1, 2, FUN = is.na)


table(col_na[,1])

table(col_na[,2])

table(col_na[,3])

table(col_na[,4])

table(col_na[,5])

table(col_na[,6])

table(col_na[,7])

table(col_na[,8])

table(col_na[,9])

table(col_na[,10])

table(col_na[,11])

table(col_na[,12])


unique(data_1[which(col_na[, 12] == TRUE), 1])


table(col_na[,13])


unique(data_1[which(col_na[, 13] == TRUE), 1])


na_states <- c("Chandigarh", "Puducherry", "Goa")
```

```
states <- unique(data_1$`State/UT/All-India`)



my_func_1 <- function(state){


 df_1 <- data_1[which(data_1$`State/UT/All-India` == state),]


 df_11 <- df_1[which(df_1$Sector == "Rural"), -(1:2)]

 df_12 <- df_1[which(df_1$Sector == "Urban"), -(1:2)]


 library(tidyverse)


 df_melted_11 <- df_11 %>%

  pivot_longer(cols = names(df_11)[-1],

               names_to = "size",

               values_to = "percentage")


 p1 <- df_melted_11 %>%

  ggplot(aes(x = Fractile.class.of.MPCE, y = percentage, fill = size)) +

  geom_col(position = "stack") +

  labs(x = "Fracticle Class of MPCE", y = "",

       title = paste(state, ": Rural"), fill = "Family Size") +

  theme(axis.text.x = element_text(angle = 45),

        axis.text.y = element_blank(),

        axis.ticks.y = element_blank(),

        legend.position = "top") +
```

```
  guides(fill = guide_legend(nrow = 1))


df_melted_12 <- df_12 %>%
  pivot_longer(cols = names(df_12)[-1],
               names_to = "size",
               values_to = "percentage")


p2 <- df_melted_12 %>%
  ggplot(aes(x = Fractile.class.of.MPCE, y = percentage, fill = size)) +
  geom_col(position = "stack") +
  labs(x = "Fracticle Class of MPCE", y = "",
       title = paste(state, ": Urban"), fill = "Family Size") +
  theme(axis.text.x = element_text(angle = 45),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        legend.position = "top") +
  guides(fill = guide_legend(nrow = 1))


library(gridExtra)


plot <- grid.arrange(p1, p2, nrow = 1, ncol = 2)


ggsave(path = "D:\\Users\\Documents\\MOSPI_hackathon\\Ananda\\1\\3\\",
       plot = plot,
       width = 16,
       height = 8,
       device='png',
```

```
        dpi=1000,
        filename = paste(state, ".png", sep = ""),
        units = "in",
        bg = "white")
}

for (i in states) {
  my_func_1(i)
}
```

3. Employment distribution state wise

#Run the following code to generate statewise employment distribution pie chart

```
df <- read.csv("D:\\Users\\Documents\\MOSPI_hackathon\\Ananda\\4\\3\\data_adjusted.csv", check.names = FALSE)

dim(df)

str(df)

df$`Regular wage/salary earning in agriculture` <- as.numeric(df$`Regular wage/salary earning in agriculture`)

str(df)

View(df)

any(is.na(df) == TRUE)

x <- which(is.na(df) == TRUE, arr.ind = TRUE); x
```

```
df[x] <- 0


for(i in 1:nrow(df)){


 jpeg(paste("D:\\Users\\Documents\\MOSPI_hackathon\\Ananda\\4\\3\\", df[i, 1], ".jpg", sep = ""),
      width = 7000, height = 4000,
      res = 500)


 values <- as.numeric(df[i, -1])


 percentages <- round(( values / sum(values) ) * 100, digits = 2)


 pie(values,
     paste(names(df)[-1], ":", percentages, "%"),
     col = c("#FF5733", "#33FF57", "#3357FF", "#FF33A8", "#FFC733", "#33FFF5", "#A833FF"),
     main = paste("Employment Distribution in", df[i, 1]))


 dev.off()
}
```