```
install.packages("tidyverse")
library(tidyverse)
library(dplyr)
library(ggplot2)
food_data <- read.csv("Food_Texture_Data.csv")
```

```
# Using View() function to preview the data frame
View(food_data)
```

```
str(food_data)
'data.frame':   50 obs. of  6 variables:
 $ X       : chr  "B110" "B136" "B171" "B192" ...
 $ Oil     : num  16.5 17.7 16.2 16.7 16.3 19.1 18.4 17.5 15.7 16.4 ...
 $ Density : int  2955 2660 2870 2920 2975 2790 2750 2770 2955 2945 ...
 $ Crispy  : int  10 14 12 10 11 13 13 10 11 11 ...
 $ Fracture: int  23 9 17 31 26 16 17 26 23 24 ...
 $ Hardness: int  97 139 143 95 143 189 114 63 123 132 ...
```

```
# summary() or glimpse() any function can be used in this case
summary(food_data)
      X                  Oil          Density         Crispy         Fracture
 Length:50          Min.   :13.7   Min.   :2570   Min.   : 7.00   Min.   : 9.
00
 Class :character   1st Qu.:16.3   1st Qu.:2772   1st Qu.:10.00   1st Qu.:17.
00
 Mode  :character   Median :16.9   Median :2868   Median :12.00   Median :21.
00
                    Mean   :17.2   Mean   :2858   Mean   :11.52   Mean   :20.
86
                    3rd Qu.:18.1   3rd Qu.:2945   3rd Qu.:13.00   3rd Qu.:25.
00
                    Max.   :21.2   Max.   :3125   Max.   :15.00   Max.   :33.
00
    Hardness
 Min.   : 63.0
 1st Qu.:107.2
 Median :126.0  
 Mean   :128.2  
 3rd Qu.:143.8  
 Max.   :192.0  
```

Hide

```
# Using nrow() and ncol() function to get the number of rows and columns
nrow(food_data)
```

```
[1] 50
```

Hide

```
ncol(food_data)
```

```
[1] 6
```

Hide

```
# Using colnames() function to get the variable names
colnames(food_data)
```

```
[1] "X"         "Oil"       "Density"  "Crispy"   "Fracture" "Hardness"
```

Hide

```
# We use head() function to get first few rows
head(food_data,5)
```

	X	Oil	Density	Crispy	Fracture	Hardness
	<chr>	<dbl>	<int>	<int>	<int>	<int>
1	B110	16.5	2955	10	23	97
2	B136	17.7	2660	14	9	139
3	B171	16.2	2870	12	17	143
4	B192	16.7	2920	10	31	95
5	B225	16.3	2975	11	26	143

5 rows

Hide

```
# Here we use mutate() function to create a column and store it in another va
riable
food_data_01 <- food_data %>% mutate(Price=Density/(Oil+Crispy+Fracture+Hardn
ess))
head(food_data_01)
```

	X	Oil	Density	Crispy	Fracture	Hardness	Price
	<chr>	<dbl>	<int>	<int>	<int>	<int>	<dbl>
1	B110	16.5	2955	10	23	97	20.17065
2	B136	17.7	2660	14	9	139	14.80245
3	B171	16.2	2870	12	17	143	15.24973
4	B192	16.7	2920	10	31	95	19.12246
5	B225	16.3	2975	11	26	143	15.15537
6	B237	19.1	2790	13	16	189	11.76719

6 rows

Hide

```
# Here we rename the column Oil to Oil Percentage and store it in a data frame

food_data_02 <- food_data_01 %>% rename(Oil_Percentage=Oil,Crispyness=Crispy,
Price_in_dollar=Price,Product=X)

head(food_data_02)
```

	Product	Oil_Percentage	Density	Crispyness	Fracture	Hardness	Price_in_dollar
	<chr>	<dbl>	<int>	<int>	<int>	<int>	<dbl>
1	B110	16.5	2955	10	23	97	20.17065
2	B136	17.7	2660	14	9	139	14.80245
3	B171	16.2	2870	12	17	143	15.24973
4	B192	16.7	2920	10	31	95	19.12246
5	B225	16.3	2975	11	26	143	15.15537
6	B237	19.1	2790	13	16	189	11.76719

6 rows

Hide

```
# Here we use mutate() function

food_data_03 <- food_data_02 %>% mutate(Price_in_Rs= 80*Price_in_dollar)

head(food_data_03)
```

	Product	Oil_Percentage	Density	Crispyness	Fracture	Hardness	Price_in_dollar	Price_in_Rs
	<chr>	<dbl>	<int>	<int>	<int>	<int>	<dbl>	<dbl>
1	B110	16.5	2955	10	23	97	20.17065	1613.6519
2	B136	17.7	2660	14	9	139	14.80245	1184.1959
3	B171	16.2	2870	12	17	143	15.24973	1219.9787

	Product	Oil_Percentage	Density	Crispyness	Fracture	Hardness	Price_in_dollar	Price_in_Rs
	<chr>	<dbl>	<int>	<int>	<int>	<int>	<dbl>	<dbl>
4	B192	16.7	2920	10	31	95	19.12246	1529.7970
5	B225	16.3	2975	11	26	143	15.15537	1212.4300
6	B237	19.1	2790	13	16	189	11.76719	941.3749

6 rows

Hide

```
# We use summarize() function to calculate this values

food_data_03 %>% summarize(min(Oil_Percentage),max(Oil_Percentage),mean(Oil_P
ercentage))
```

Hide

```
# We use summarize() function to calculate this values

food_data_03 %>% summarize(min(Density),max(Density),mean(Density))
```

Hide

```
# We use summarize() function to calculate this values

food_data_03 %>% summarize(min(Crispyness),max(Crispyness),mean(Crispyness))
```

Hide

```
# We use summarize() function to calculate this values

food_data_03 %>% summarize(min(Fracture),max(Fracture),mean(Fracture))
```

Hide

```
# We use summarize() function to calculate this values

food_data_03 %>% summarize(min(Hardness),max(Hardness),mean(Hardness))
```

Hide

```
# We use summarize() function to calculate this values

food_data_03 %>% summarize(min(Price_in_Rs),max(Price_in_Rs),mean(Price_in_Rs
))
```

Hide

```
View(food_data_03)
```

Hide

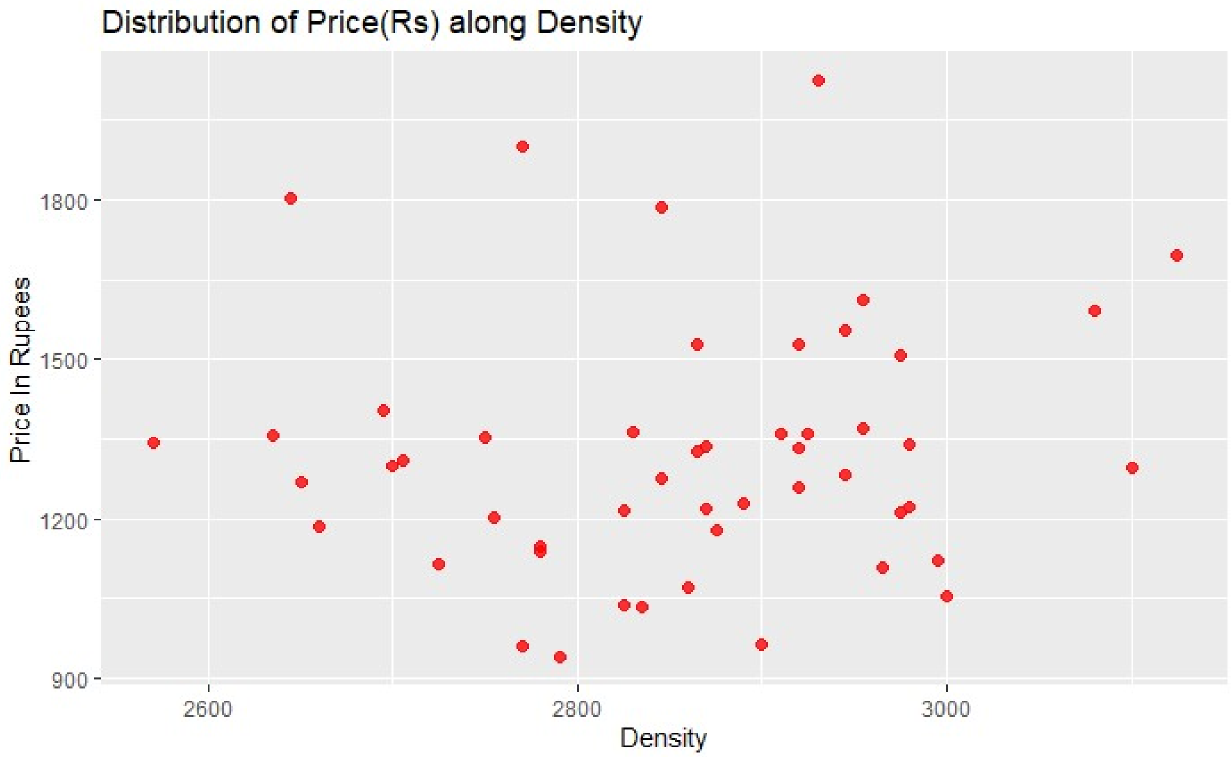
```
# Find the product that has lowest Price_in_Rs

food_data_03 %>% filter(Price_in_Rs == min(Price_in_Rs)) %>% select(Product)
```

Product
<chr>
B237

1 row

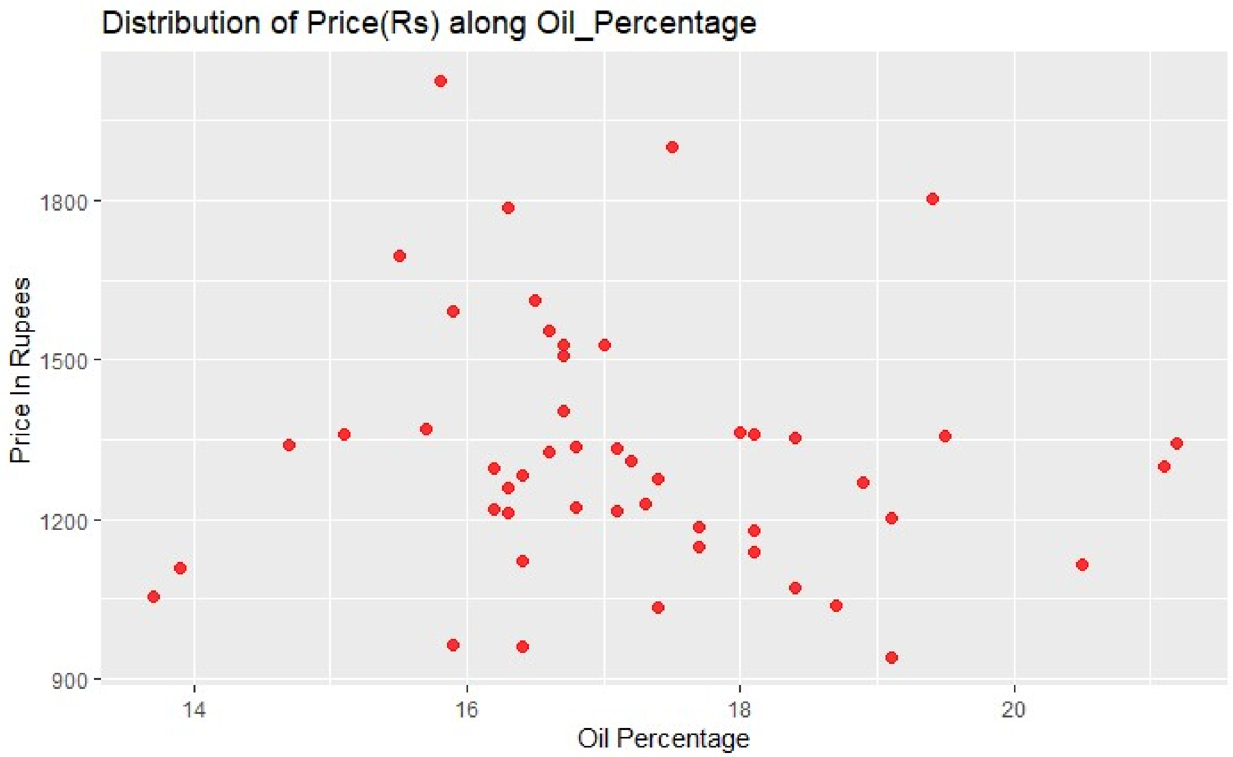
Hide

```
# Dot plot of Density vs Price_in_Rs
p=ggplot(data=food_data_03)+
  geom_point(mapping = aes(x=Density,y=Price_in_Rs),color='red',alpha=0.8,size=2)+
  labs(title='Distribution of Price(Rs) along Density',x='Density',y='Price In Rupees')
p
```



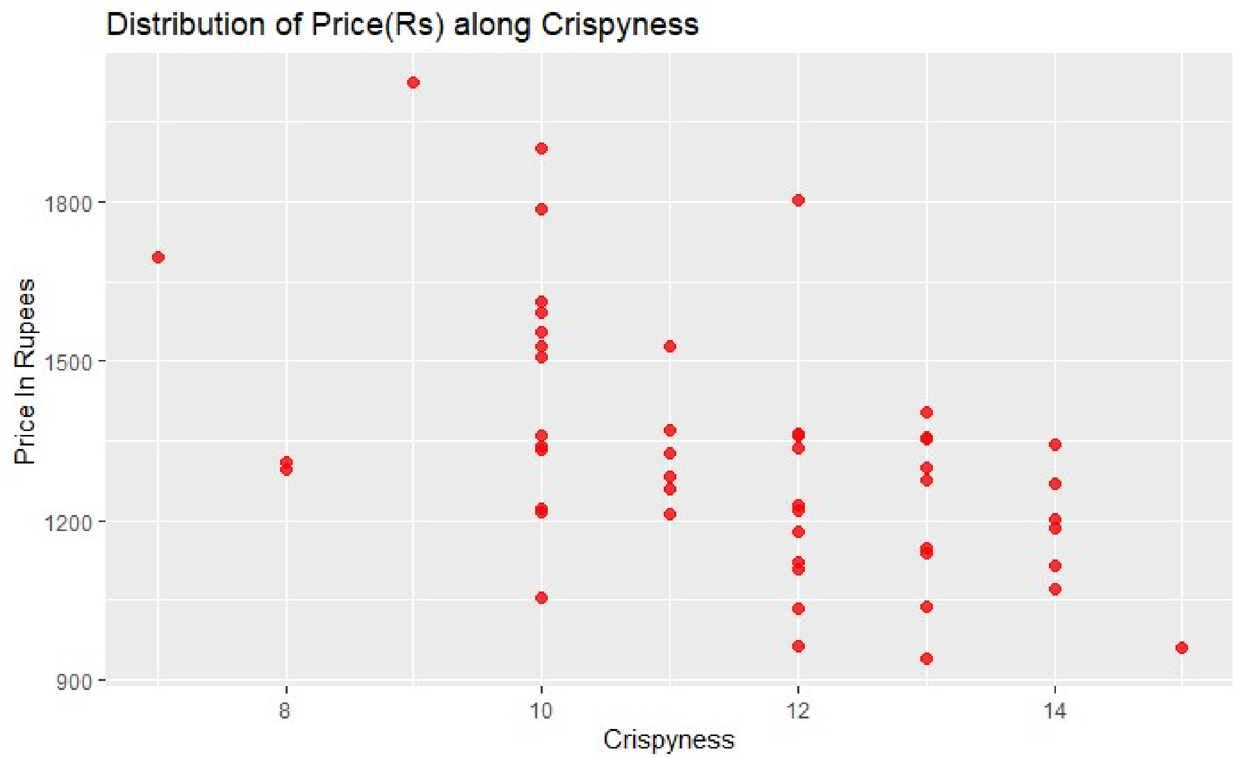
Hide

```
# Dot plot of Oil_Percentage vs Price_in_Rs
p1=ggplot(data=food_data_03)+
  geom_point(mapping = aes(x=Oil_Percentage,y=Price_in_Rs),color='red',alpha=0.8,size=2)+
  labs(title='Distribution of Price(Rs) along Oil_Percentage',x='Oil Percentage',y='Price In Rupees')
p1
```



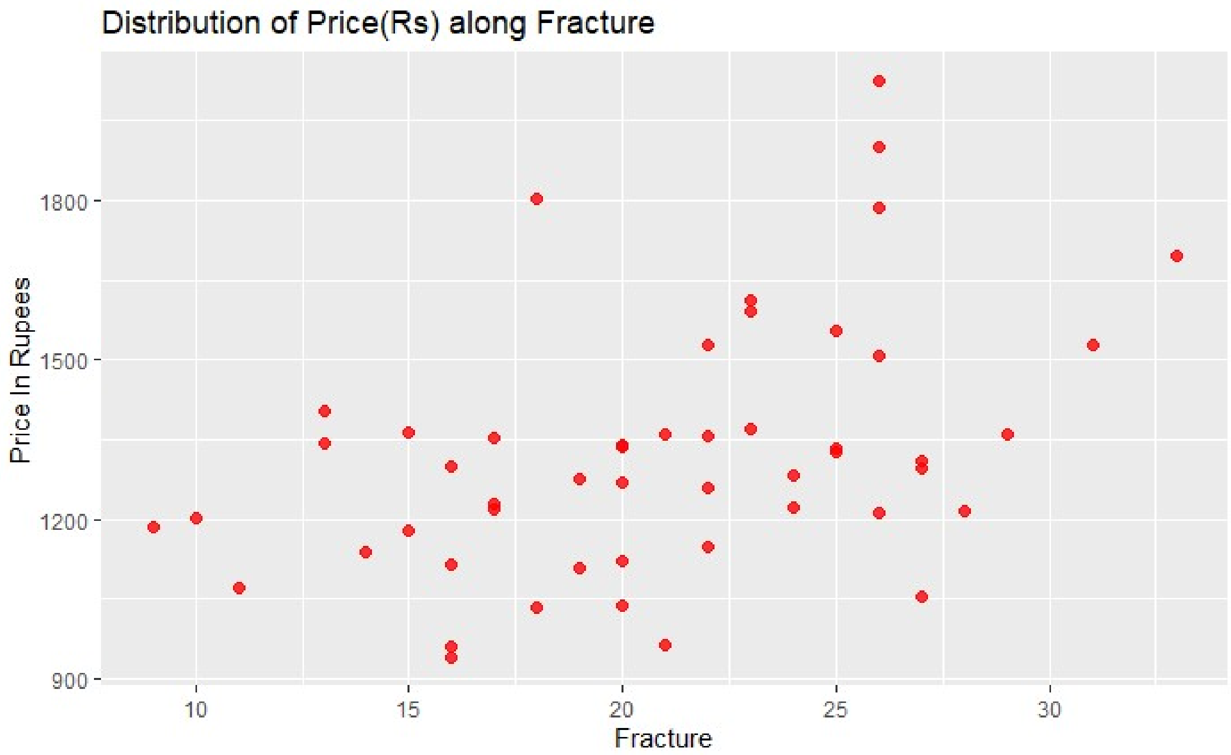
Hide

```
# Dot plot of Crispness vs Price_in_Rs
p2=ggplot(data=food_data_03)+
  geom_point(mapping = aes(x=Crispyness,y=Price_in_Rs),color='red',alpha=0.8,
size=2)+
  labs(title='Distribution of Price(Rs) along Crispyness',x='Crispyness',y='P
rice In Rupees')
p2
```



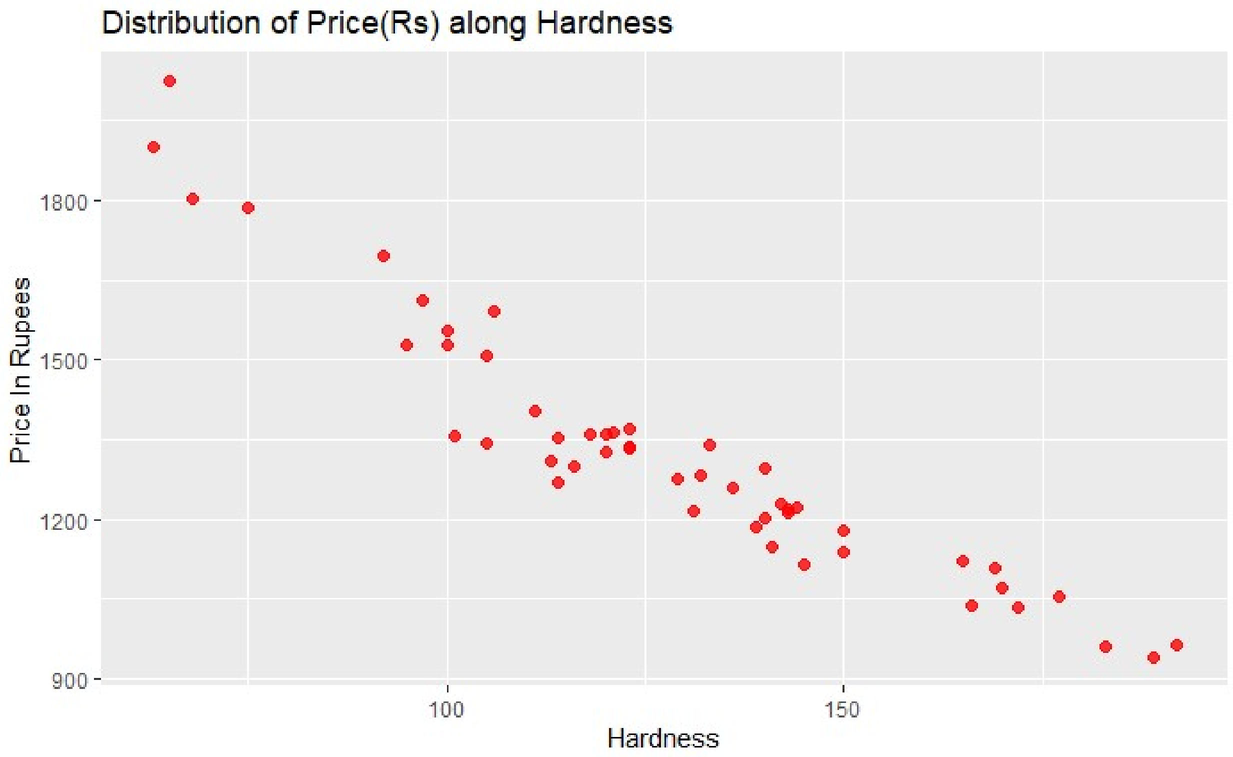
Hide

```
# Dot plot of Fracture vs Price_in_Rs

p3=ggplot(data=food_data_03)+

  geom_point(mapping = aes(x=Fracture,y=Price_in_Rs),color='red',alpha=0.8,size=2)+

  labs(title='Distribution of Price(Rs) along Fracture',x='Fracture',y='Price In Rupees')

p3
```



Hide

```
# Dot plot of Hardness vs Price_in_Rs

p4=ggplot(data=food_data_03)+

  geom_point(mapping = aes(x=Hardness,y=Price_in_Rs),color='red',alpha=0.8,si
ze=2)+

  labs(title='Distribution of Price(Rs) along Hardness',x='Hardness',y='Price
In Rupees')

p4
```


Distribution of Price(Rs) along Hardness
Price In Rupees
1800
1500
1200
900
100
150
Hardness