BUS 4066

Introduction to Analytics

R Assignment - Group Work

11 June 2023

Submission by

Group 7

Print the structure of the dataset

Load the airquality dataset

```
data(airquality)

str(airquality)
'data.frame':	153 obs. of  6 variables:
 $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
 $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
 $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
 $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
 $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

List the variables in the dataset

```
variables <- names(airquality)
print(variables)
[1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
```

Print the top 15 rows of the dataset

```
head(airquality, 15)
   Ozone Solar.R Wind Temp Month Day
1     41     190  7.4   67     5   1
2     36     118  8.0   72     5   2
3     12     149 12.6   74     5   3
4     18     313 11.5   62     5   4
5     NA      NA 14.3   56     5   5
6     28      NA 14.9   66     5   6
7     23     299  8.6   65     5   7
8     19      99 13.8   59     5   8
9      8      19 20.1   61     5   9
10    NA     194  8.6   69     5  10
11     7      NA  6.9   74     5  11
12    16     256  9.7   69     5  12
13    11     290  9.2   66     5  13
14    14     274 10.9   68     5  14
15    18      65 13.2   58     5  15
```

Write a user defined function using any of the variables from the data set

```
customFunction <- function(temp) {
+    if (temp > 80) {
+        return("Hot")
+    } else if (temp > 60) {
+        return("Moderate")
+    } else {
+        return("Cool")
+    }
+ }

Example usage of the user-defined function
temperature <- airquality$Temp[1]  # Using the "Temp" variable from the dataset
result <- customFunction(temperature)
print(result)
[1] "Moderate"
```

Use data manipulation techniques and filter rows based on any logical criteria that exist in your dataset.

```
library(datasets)
data(airquality)
View(airquality)
# Load the datasets package
library(datasets)

# Load the airquality dataset
data(airquality)

# Filter rows with ozone level above 30
filtered_data <- airquality[airquality$Ozone > 30, ]

# View the filtered dataset
head(filtered_data)
     Ozone Solar.R Wind Temp Month Day
1       41     190  7.4   67     5   1
2       36     118  8.0   72     5   2
NA      NA      NA   NA   NA    NA  NA
NA.1    NA      NA   NA   NA    NA  NA
17      34     307 12.0   66     5  17
24      32      92 12.0   61     5  24

View(airquality)
View(filtered_data)
```

Identify the dependent & independent variables and use reshaping techniques and create a new data frame by joining those variables from your dataset.

```
# Load the datasets package
library(datasets)

# Load the airquality dataset
data(airquality)
```

Select the dependent and independent variables
```
dependent_var <- airquality$Ozone
independent_vars <- airquality[, c("Solar.R", "Wind", "Temp", "Month")]
```

Create a new data frame by joining the variables
```
new_df <- cbind(dependent_var, independent_vars)
```

View the new data frame
```
head(new_df)
  dependent_var Solar.R Wind Temp Month
1            41     190  7.4   67     5
2            36     118  8.0   72     5
3            12     149 12.6   74     5
4            18     313 11.5   62     5
5            NA      NA 14.3   56     5
6            28      NA 14.9   66     5
```

```
View(independent_vars)
View(new_df)
View(new_df)
View(independent_vars)
View(new_df)
```
Load the datasets package
```
library(datasets)
```

Load the airquality dataset
```
data(airquality)
```

Create a PDF file
```
pdf("data_output.pdf")
```

Print the dataset or any desired information
```
print(airquality)
    Ozone Solar.R Wind Temp Month Day
1      41     190  7.4   67     5   1
2      36     118  8.0   72     5   2
3      12     149 12.6   74     5   3
4      18     313 11.5   62     5   4
5      NA      NA 14.3   56     5   5
6      28      NA 14.9   66     5   6
7      23     299  8.6   65     5   7
8      19      99 13.8   59     5   8
9       8      19 20.1   61     5   9
10     NA     194  8.6   69     5  10
11      7      NA  6.9   74     5  11
12     16     256  9.7   69     5  12
13     11     290  9.2   66     5  13
14     14     274 10.9   68     5  14
```

15	18	65	13.2	58	5	15
16	14	334	11.5	64	5	16
17	34	307	12.0	66	5	17
18	6	78	18.4	57	5	18
19	30	322	11.5	68	5	19
20	11	44	9.7	62	5	20
21	1	8	9.7	59	5	21
22	11	320	16.6	73	5	22
23	4	25	9.7	61	5	23
24	32	92	12.0	61	5	24
25	NA	66	16.6	57	5	25
26	NA	266	14.9	58	5	26
27	NA	NA	8.0	57	5	27
28	23	13	12.0	67	5	28
29	45	252	14.9	81	5	29
30	115	223	5.7	79	5	30
31	37	279	7.4	76	5	31
32	NA	286	8.6	78	6	1
33	NA	287	9.7	74	6	2
34	NA	242	16.1	67	6	3
35	NA	186	9.2	84	6	4
36	NA	220	8.6	85	6	5
37	NA	264	14.3	79	6	6
38	29	127	9.7	82	6	7
39	NA	273	6.9	87	6	8
40	71	291	13.8	90	6	9
41	39	323	11.5	87	6	10
42	NA	259	10.9	93	6	11
43	NA	250	9.2	92	6	12
44	23	148	8.0	82	6	13
45	NA	332	13.8	80	6	14
46	NA	322	11.5	79	6	15
47	21	191	14.9	77	6	16
48	37	284	20.7	72	6	17
49	20	37	9.2	65	6	18
50	12	120	11.5	73	6	19
51	13	137	10.3	76	6	20
52	NA	150	6.3	77	6	21
53	NA	59	1.7	76	6	22
54	NA	91	4.6	76	6	23
55	NA	250	6.3	76	6	24
56	NA	135	8.0	75	6	25
57	NA	127	8.0	78	6	26
58	NA	47	10.3	73	6	27
59	NA	98	11.5	80	6	28
60	NA	31	14.9	77	6	29
61	NA	138	8.0	83	6	30
62	135	269	4.1	84	7	1
63	49	248	9.2	85	7	2

64	32	236	9.2	81	7	3
65	NA	101	10.9	84	7	4
66	64	175	4.6	83	7	5
67	40	314	10.9	83	7	6
68	77	276	5.1	88	7	7
69	97	267	6.3	92	7	8
70	97	272	5.7	92	7	9
71	85	175	7.4	89	7	10
72	NA	139	8.6	82	7	11
73	10	264	14.3	73	7	12
74	27	175	14.9	81	7	13
75	NA	291	14.9	91	7	14
76	7	48	14.3	80	7	15
77	48	260	6.9	81	7	16
78	35	274	10.3	82	7	17
79	61	285	6.3	84	7	18
80	79	187	5.1	87	7	19
81	63	220	11.5	85	7	20
82	16	7	6.9	74	7	21
83	NA	258	9.7	81	7	22
84	NA	295	11.5	82	7	23
85	80	294	8.6	86	7	24
86	108	223	8.0	85	7	25
87	20	81	8.6	82	7	26
88	52	82	12.0	86	7	27
89	82	213	7.4	88	7	28
90	50	275	7.4	86	7	29
91	64	253	7.4	83	7	30
92	59	254	9.2	81	7	31
93	39	83	6.9	81	8	1
94	9	24	13.8	81	8	2
95	16	77	7.4	82	8	3
96	78	NA	6.9	86	8	4
97	35	NA	7.4	85	8	5
98	66	NA	4.6	87	8	6
99	122	255	4.0	89	8	7
100	89	229	10.3	90	8	8
101	110	207	8.0	90	8	9
102	NA	222	8.6	92	8	10
103	NA	137	11.5	86	8	11
104	44	192	11.5	86	8	12
105	28	273	11.5	82	8	13
106	65	157	9.7	80	8	14
107	NA	64	11.5	79	8	15
108	22	71	10.3	77	8	16
109	59	51	6.3	79	8	17
110	23	115	7.4	76	8	18
111	31	244	10.9	78	8	19
112	44	190	10.3	78	8	20

```
113    21     259 15.5   77      8  21
114     9      36 14.3   72      8  22
115    NA     255 12.6   75      8  23
116    45     212  9.7   79      8  24
117   168     238  3.4   81      8  25
118    73     215  8.0   86      8  26
119    NA     153  5.7   88      8  27
120    76     203  9.7   97      8  28
121   118     225  2.3   94      8  29
122    84     237  6.3   96      8  30
123    85     188  6.3   94      8  31
124    96     167  6.9   91      9   1
125    78     197  5.1   92      9   2
126    73     183  2.8   93      9   3
127    91     189  4.6   93      9   4
128    47      95  7.4   87      9   5
129    32      92 15.5   84      9   6
130    20     252 10.9   80      9   7
131    23     220 10.3   78      9   8
132    21     230 10.9   75      9   9
133    24     259  9.7   73      9  10
134    44     236 14.9   81      9  11
135    21     259 15.5   76      9  12
136    28     238  6.3   77      9  13
137     9      24 10.9   71      9  14
138    13     112 11.5   71      9  15
139    46     237  6.9   78      9  16
140    18     224 13.8   67      9  17
141    13      27 10.3   76      9  18
142    24     238 10.3   68      9  19
143    16     201  8.0   82      9  20
144    13     238 12.6   64      9  21
145    23      14  9.2   71      9  22
146    36     139 10.3   81      9  23
147     7      49 10.3   69      9  24
148    14      20 16.6   63      9  25
149    30     193  6.9   70      9  26
150    NA     145 13.2   77      9  27
151    14     191 14.3   75      9  28
152    18     131  8.0   76      9  29
153    20     223 11.5   68      9  30
```

Save additional information

Remove missing values from the airquality dataset

```
clean_airquality <- na.omit(airquality)
```

Identify and remove duplicated data in your dataset

Identify duplicate rows
```
duplicated_rows <- duplicated(airquality)
```

Print the duplicate rows
```
duplicate_data <- airquality[duplicated_rows, ]
print(duplicate_data)
```

Remove duplicate rows
```
clean_airquality <- unique(airquality)
print(clean_airquality)
```

Load the required package
```
library(dplyr)
```

Reorder rows in descending order based on the Ozone column

```
reordered_airquality <- airquality %>% arrange(desc(Ozone))
```

Print the reordered dataset
```
print(reordered_airquality)
```

Rename some of the column names in your dataset

```
names(airquality)[names(airquality)=="Temp"]<-"Temperature"
names(airquality)[names(airquality)=="Wind"]<-"Wind Level"
```

Check airquality data set column names
```
colnames(airquality)
```

Add new variables in your data frame by using a mathematical function (for e.g. – multiply an existing column by 2 and add it as a new variable to your data frame)

```
# Add a new variable by multiplying an existing column by 5
airquality$Temp_Double <- airquality$Temp * 5

# Print the updated data frame
print(airquality)
```

Create a training set using random number generator engine

```
# Set a seed for reproducibility
set.seed(123)

# Create a training set using a random number generator
train_indices <- sample(1:nrow(airquality), size = 100, replace = FALSE)
training_set <- airquality[train_indices, ]

# Print the training set
print(training_set)
```

Print summary of the airquality dataset

```
summary(airquality)
     Ozone           Solar.R           Wind             Temp      
 Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00  
 1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00  
 Median : 31.50   Median :205.0   Median : 9.700   Median :79.00  
 Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88  
 3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00  
 Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00  
 NA's   :37       NA's   :7                                       
     Month            Day      
 Min.   :5.000   Min.   : 1.0  
 1st Qu.:6.000   1st Qu.: 8.0  
 Median :7.000   Median :16.0  
 Mean   :6.993   Mean   :15.8  
 3rd Qu.:8.000   3rd Qu.:23.0  
 Max.   :9.000   Max.   :31.0  
```

Use any of the numerical variables from the dataset and perform the following statistical functions • Mean • Median • Mode • Range

Extract the "Ozone" variable from the airquality dataset

```
ozone <- airquality$Ozone
```

Calculate the mean

```
mean_value <- mean(ozone, na.rm = TRUE)
```

Calculate the median

```
median_value <- median(ozone, na.rm = TRUE)
```

Calculate the mode

```
mode_value <- as.numeric(names(which.max(table(ozone))))
```

Calculate the range

```
range_value <- range(ozone, na.rm = TRUE)
```

Print the mean, median, mode, and range for Ozone

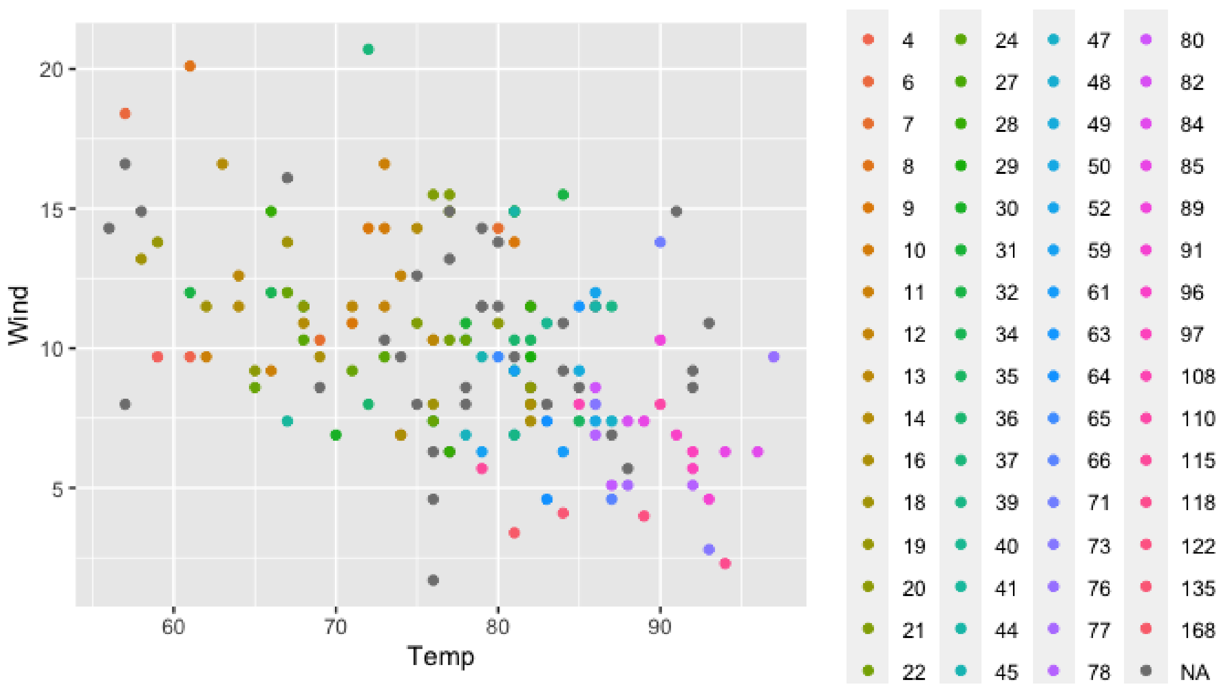
```
cat("Mean:", mean_value, "\n")
Mean: 42.12931
cat("Median:", median_value, "\n")
Median: 31.5
cat("Mode:", mode_value, "\n")
Mode: 23
cat("Range:", range_value[2] - range_value[1], "\n")
Range: 167
```

Plot a scatter plot for any 2 variables in your dataset

```
install.packages("airquality")
View(airquality)

install.packages("ggplot2")

```{r}
library(ggplot2)
library(ggpubr)
#Plot a scatter plot for any 2 variables in your dataset
ScatterPlot<-ggplot(data = airquality,aes(x = Temp,y = Wind,col =
factor(Ozone)))+geom_point()
```
```{r show_figure, fig.width = 9, fig.height = 3}
ScatterPlot
```
```

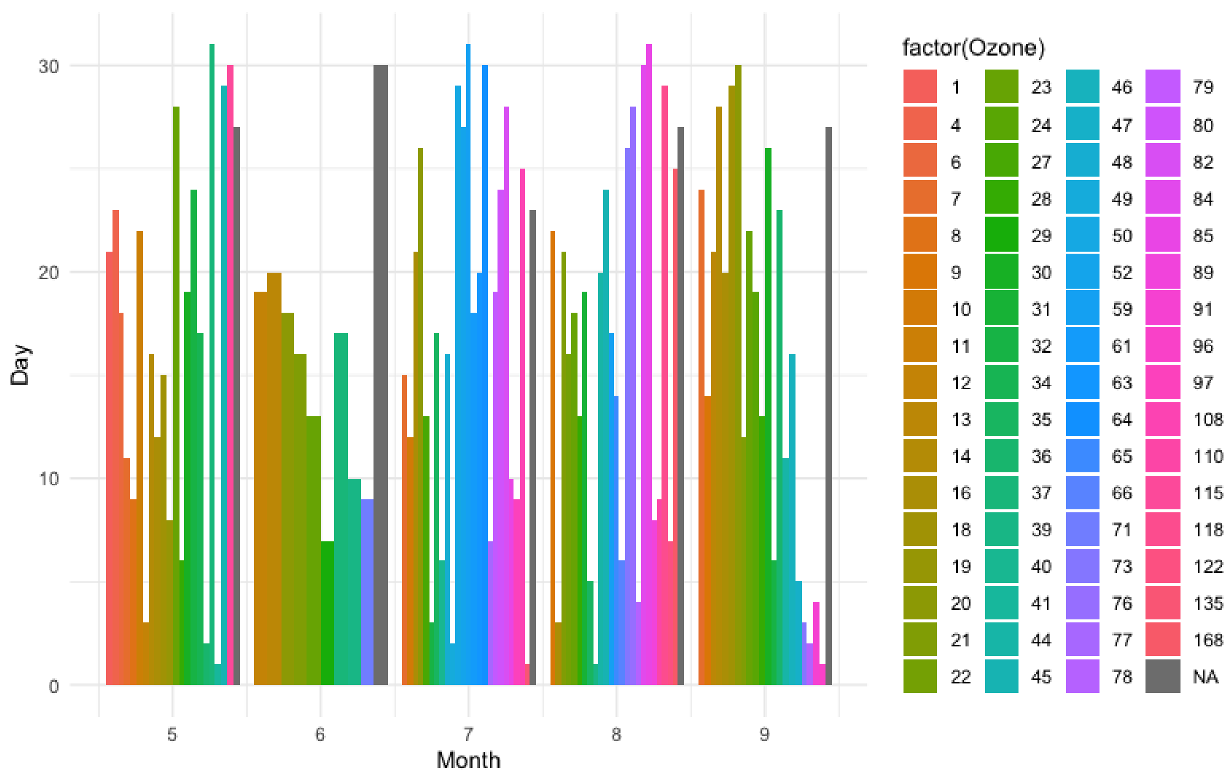


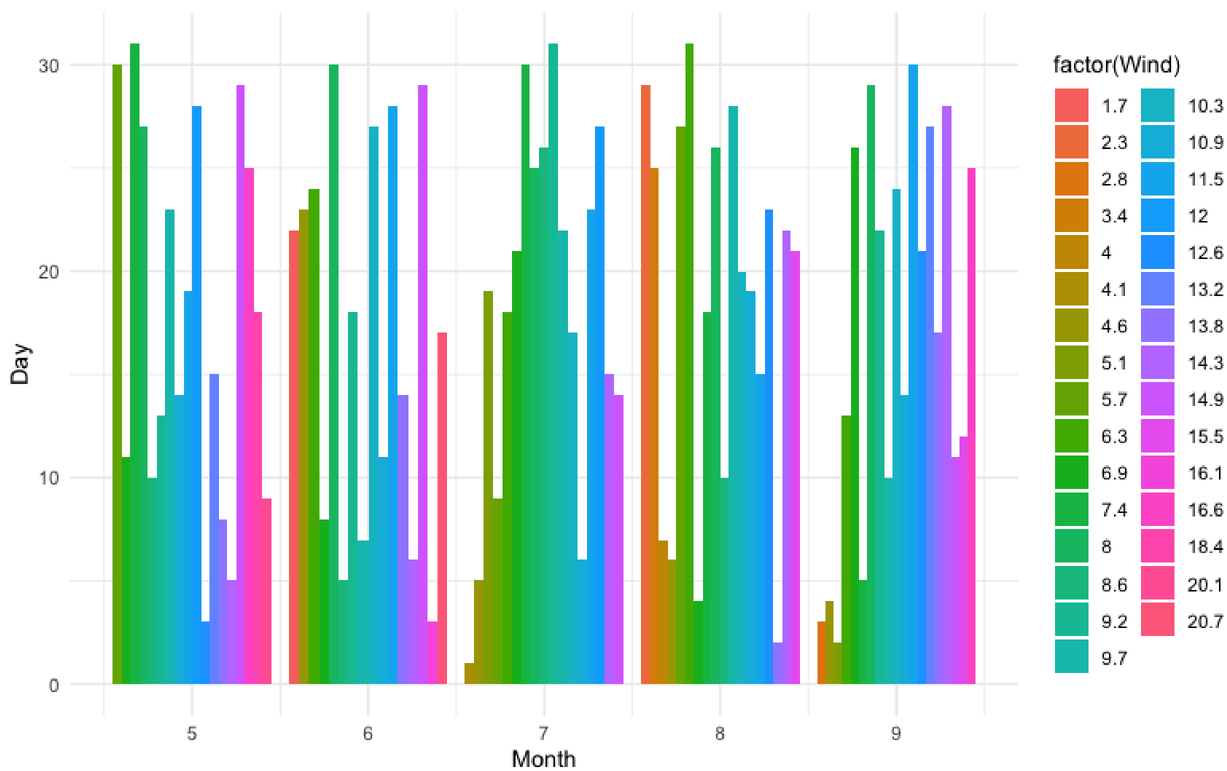
Plot a bar plot for any 2 variables in your dataset

```{r}
# Plot a bar plot for any 2 variables in your dataset
## Barplot Version 1 Factor Ozone
BarplotV1<-ggplot(data = airquality,aes(x = Month,y=Day, fill =
factor(Ozone)))+geom_bar(stat="identity",position=position_dodge())+theme_minim
al()

## Barplot Version 1 Factor Wind
BarplotV2<-ggplot(data = airquality,aes(x = Month,y=Day, fill = factor(Wind)))+
  geom_bar(stat="identity",
  position=position_dodge())+theme_minimal()
```

```{r show_figure1, fig.width = 9, fig.height = 3}
BarplotV1
BarplotV2
```





Day
30
20
10
0
5
6
7
8
9
Month
factor(Wind)
1.7
2.3
2.8
3.4
4
4.1
4.6
5.1
5.7
6.3
6.9
7.4
8
8.6
9.2
9.7
10.3
10.9
11.5
12
12.6
13.2
13.8
14.3
14.9
15.5
16.1
16.6
18.4
20.1
20.7

Find the correlation between any 2 variables by applying least square linear regression model

```{r}
# Find the correlation between any 2 variables by applying least square linear
regression model
ScatterModel<-ggscatter(airquality, x = "Wind", y = "Temp",
  add = "reg.line", conf.int = TRUE,cor.coef = TRUE,
  cor.method = "pearson", xlab = "Wind", ylab = "Temperature")
y<-airquality[,"Temp"]
x<-airquality[,"Wind"]
xycorr<- cor(y,x, method="pearson")
head(xycorr)
```
```{r show_figure2, fig.width = 6, fig.height = 3}
ScatterModel
```