GGPLOT EDA

June 7, 2024

[1]:
```
##Task 1 - Load the dataset Ensure you read DATE as type character
seoul_bike_sharing <- read.csv("https://cf-courses-data.s3.us.
 ↪cloud-object-storage.appdomain.cloud/
 ↪IBMDeveloperSkillsNetwork-RP0321EN-SkillsNetwork/labs/datasets/
 ↪seoul_bike_sharing.csv", colClasses=c(DATE="character"))
head(seoul_bike_sharing)
```

A data.frame: 6 × 14

	DATE <chr>	RENTED_BIKE_COUNT <int>	HOUR <int>	TEMPERATURE <dbl>	HUMIDITY <int>
1	01/12/2017	254	0	-5.2	37
2	01/12/2017	204	1	-5.5	38
3	01/12/2017	173	2	-6.0	39
4	01/12/2017	107	3	-6.2	40
5	01/12/2017	78	4	-6.0	36
6	01/12/2017	100	5	-6.4	37

[2]:
```
##Task 2 - Recast DATE as a date Use the format of the data, namely "%d/%m/%Y".
seoul_bike_sharing$DATE <- as.Date(seoul_bike_sharing$DATE, "%d/%m/%Y")
```

[3]:
```
seoul_bike_sharing$SEASONS <- as.factor(seoul_bike_sharing$SEASONS)
seoul_bike_sharing$HOLIDAY <- as.factor(seoul_bike_sharing$HOLIDAY)
seoul_bike_sharing$FUNCTIONING_DAY <- as.
 ↪factor(seoul_bike_sharing$FUNCTIONING_DAY)
```

[4]:
```
##Task 3 - Cast HOURS as a categorical variable
seoul_bike_sharing$HOUR <- factor(seoul_bike_sharing$HOUR, ordered = TRUE)
head(seoul_bike_sharing)
```

A data.frame: 6 × 14

	DATE <date>	RENTED_BIKE_COUNT <int>	HOUR <ord>	TEMPERATURE <dbl>	HUMIDITY <int>
1	2017-12-01	254	0	-5.2	37
2	2017-12-01	204	1	-5.5	38
3	2017-12-01	173	2	-6.0	39
4	2017-12-01	107	3	-6.2	40
5	2017-12-01	78	4	-6.0	36
6	2017-12-01	100	5	-6.4	37

[5]:
```
#Check the structure of the dataframe
str(seoul_bike_sharing)
```

```
'data.frame':   8465 obs. of  14 variables:
 $ DATE                 : Date, format: "2017-12-01" "2017-12-01" …
 $ RENTED_BIKE_COUNT    : int  254 204 173 107 78 100 181 460 930 490 …
 $ HOUR                 : Ord.factor w/ 24 levels "0"<"1"<"2"<"3"<..: 1 2 3 4 5
6 7 8 9 10 …
 $ TEMPERATURE          : num  -5.2 -5.5 -6 -6.2 -6 -6.4 -6.6 -7.4 -7.6 -6.5 …
 $ HUMIDITY             : int  37 38 39 40 36 37 35 38 37 27 …
 $ WIND_SPEED           : num  2.2 0.8 1 0.9 2.3 1.5 1.3 0.9 1.1 0.5 …
 $ VISIBILITY           : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 1928
…
 $ DEW_POINT_TEMPERATURE: num  -17.6 -17.6 -17.7 -17.6 -18.6 -18.7 -19.5 -19.3
-19.8 -22.4 …
 $ SOLAR_RADIATION      : num  0 0 0 0 0 0 0 0 0.01 0.23 …
 $ RAINFALL             : num  0 0 0 0 0 0 0 0 0 0 …
 $ SNOWFALL             : num  0 0 0 0 0 0 0 0 0 0 …
 $ SEASONS              : Factor w/ 4 levels "Autumn","Spring",..: 4 4 4 4 4 4 4
4 4 4 …
 $ HOLIDAY              : Factor w/ 2 levels "Holiday","No Holiday": 2 2 2 2 2 2
2 2 2 2 …
 $ FUNCTIONING_DAY      : Factor w/ 1 level "Yes": 1 1 1 1 1 1 1 1 1 1 …
```

[6]:
```
#Finally, ensure there are no missing values
sum(is.na(seoul_bike_sharing))
```

```
0
```

[7]:
```
###Descriptive Statistics
#Task 4 - Dataset Summary
summary(seoul_bike_sharing)
```

```
      DATE             RENTED_BIKE_COUNT      HOUR       TEMPERATURE
 Min.   :2017-12-01   Min.   :   2.0    7      : 353   Min.   :-17.80
 1st Qu.:2018-02-27   1st Qu.: 214.0    8      : 353   1st Qu.:  3.00
 Median :2018-05-28   Median : 542.0    9      : 353   Median : 13.50
 Mean   :2018-05-28   Mean   : 729.2    10     : 353   Mean   : 12.77
 3rd Qu.:2018-08-24   3rd Qu.:1084.0    11     : 353   3rd Qu.: 22.70
 Max.   :2018-11-30   Max.   :3556.0    12     : 353   Max.   : 39.40
                                        (Other):6347
    HUMIDITY       WIND_SPEED      VISIBILITY   DEW_POINT_TEMPERATURE
 Min.   : 0.00   Min.   :0.000   Min.   :  27   Min.   :-30.600
 1st Qu.:42.00   1st Qu.:0.900   1st Qu.: 935   1st Qu.: -5.100
 Median :57.00   Median :1.500   Median :1690   Median :  4.700
 Mean   :58.15   Mean   :1.726   Mean   :1434   Mean   :  3.945
 3rd Qu.:74.00   3rd Qu.:2.300   3rd Qu.:2000   3rd Qu.: 15.200
 Max.   :98.00   Max.   :7.400   Max.   :2000   Max.   : 27.200
```

```
 SOLAR_RADIATION      RAINFALL          SNOWFALL          SEASONS
 Min.   :0.0000   Min.   : 0.0000   Min.   :0.00000   Autumn:1937
 1st Qu.:0.0000   1st Qu.: 0.0000   1st Qu.:0.00000   Spring:2160
 Median :0.0100   Median : 0.0000   Median :0.00000   Summer:2208
 Mean   :0.5679   Mean   : 0.1491   Mean   :0.07769   Winter:2160
 3rd Qu.:0.9300   3rd Qu.: 0.0000   3rd Qu.:0.00000
 Max.   :3.5200   Max.   :35.0000   Max.   :8.80000

       HOLIDAY     FUNCTIONING_DAY
 Holiday    : 408   Yes:8465
 No Holiday:8057
```

[8]:
```
#Task 5 - Based on the above stats, calculate how many Holidays there are
sum(seoul_bike_sharing$HOLIDAY == 'Holiday')
```

408

[9]:
```
#Task 6 - Calculate the percentage of records that fall on a holiday
(sum(seoul_bike_sharing$HOLIDAY == 'Holiday') / nrow(seoul_bike_sharing)) * 100
```

4.8198464264619

[10]:
```
#Task 7 - Given there is exactly a full year of data, determine how many␣
 ↪records we expect to have.
expect_year <- length(seq(from = min(seoul_bike_sharing$DATE), to =␣
 ↪max(seoul_bike_sharing$DATE), by = 'day'))-1
(nrow(seoul_bike_sharing) / expect_year) * 365
```

8488.25549450549

[11]:
```
#task 8 - Given the observations for the 'FUNCTIONING_DAY' how many records␣
 ↪must there be?
sum(seoul_bike_sharing$FUNCTIONING_DAY == 'Yes')
```

8465

[12]:
```
##Drilling Down
#Task 9 - Load the dplyr package, group the data by SEASONS, and use the␣
 ↪summarize() function to
#calculate:the seasonal total rainfall and snowfal

library(tidyverse)
```

```
 Attaching packages                     tidyverse 1.3.0
ggplot2 3.3.0      purrr   0.3.4
tibble  3.0.1      dplyr   0.8.5
tidyr   1.0.2      stringr 1.4.0
readr   1.3.1      forcats 0.5.0
 Conflicts                      tidyverse_conflicts()
dplyr::filter() masks stats::filter()
dplyr::lag()    masks stats::lag()
```

[13]:
```
seoul_bike_sharing %>%
group_by(SEASONS) %>%
summarize(TOTAL_RAINFALL = sum(RAINFALL), TOTAL_SNOWFALL = sum(SNOWFALL))
```

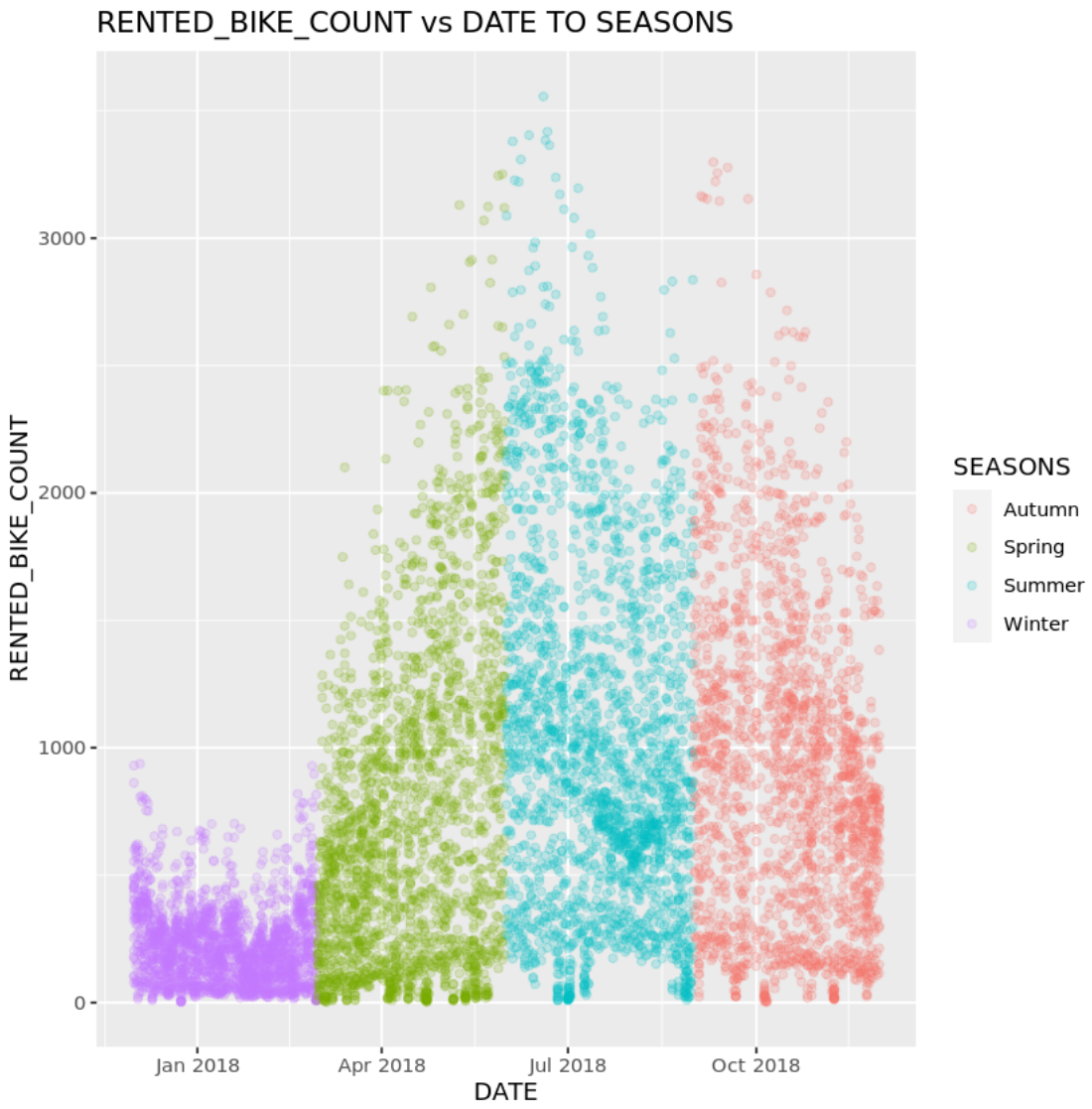
A tibble: 4 × 3

SEASONS <fct>	TOTAL_RAINFALL <dbl>	TOTAL_SNOWFALL <dbl>
Autumn	227.9	123.0
Spring	403.8	0.0
Summer	559.7	0.0
Winter	70.9	534.6

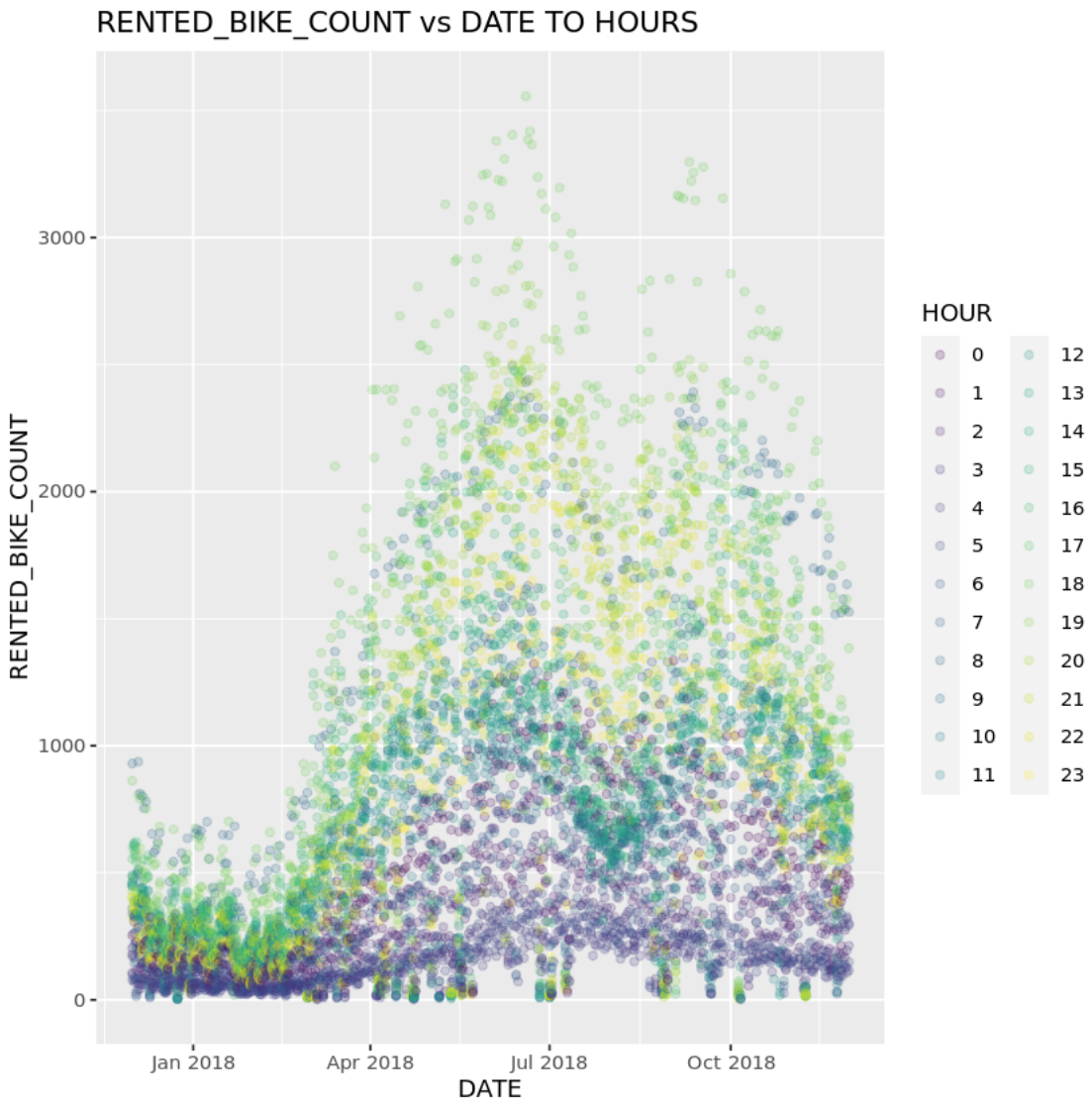
[14]:
```
##Data Visualization
#Load the ggplot2 package so we can generate some data visualizations.
install.packages('ggthemes')
library(ggthemes)
library(ggplot2)
```

```
Updating HTML index of packages in '.Library'
Making 'packages.html' … done
```

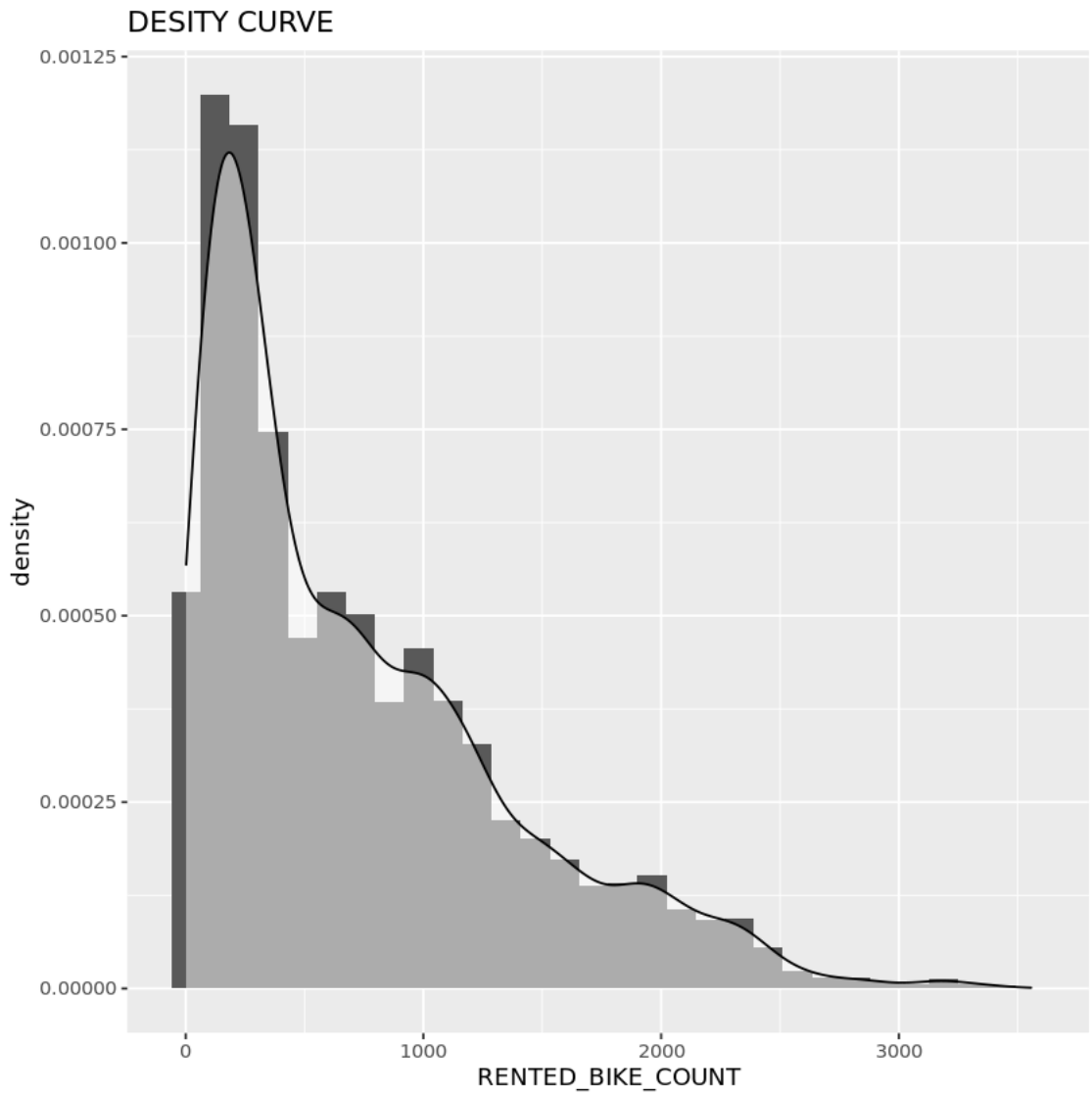
[15]:
```
#Task 10 - Create a scatter plot of RENTED_BIKE_COUNT vs DATE using the alpha
ggplot(seoul_bike_sharing,aes(x=DATE, y=RENTED_BIKE_COUNT, color = SEASONS)) +
geom_point(alpha = 0.2) +
labs(title= "RENTED_BIKE_COUNT vs DATE TO SEASONS",␣
 ↪x="DATE",y="RENTED_BIKE_COUNT")
```



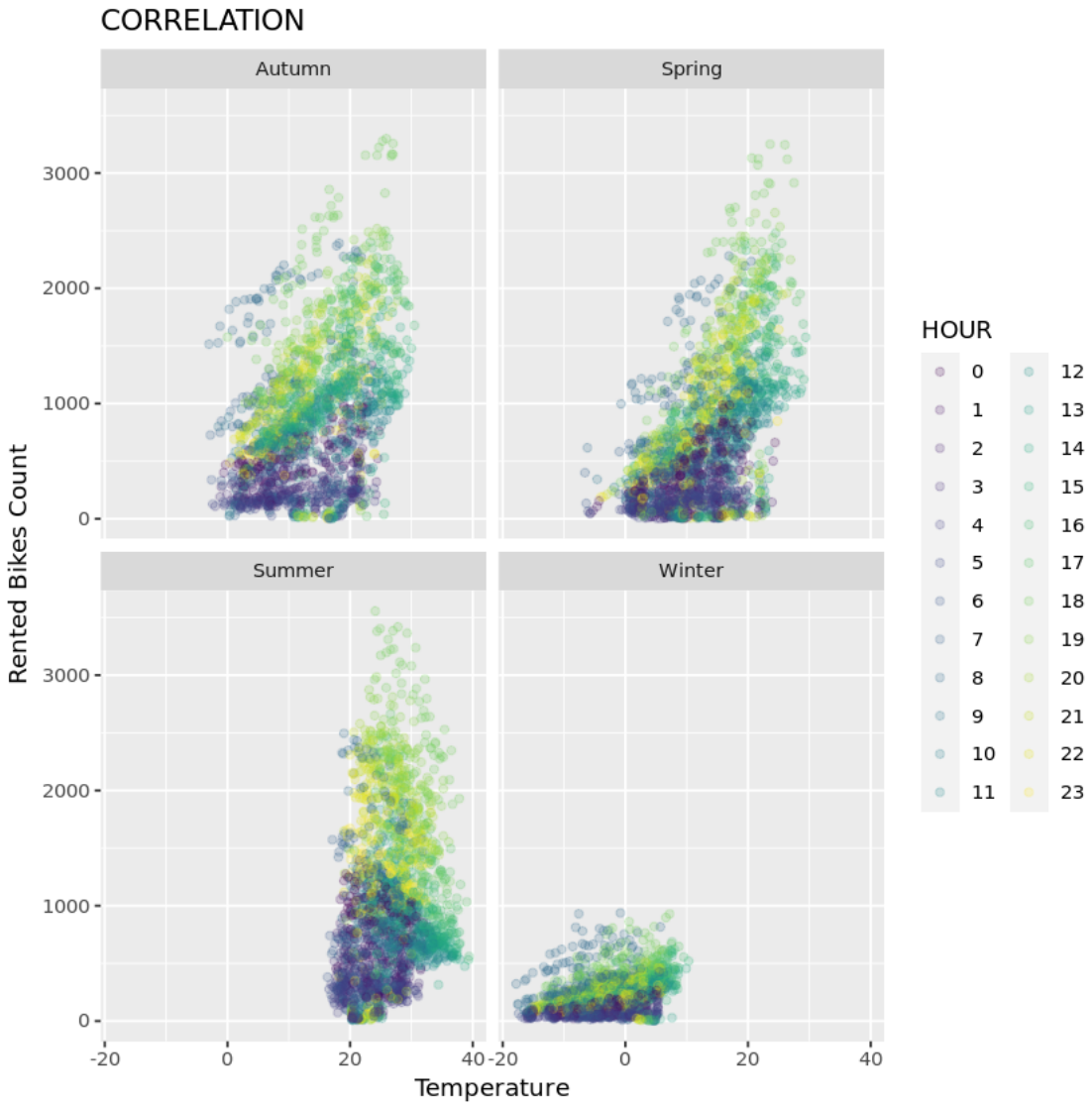
[16]:
```
#Task 11 - Create the same plot of the RENTED_BIKE_COUNT time series, but now␣
 ↪add HOURS as the colour
ggplot(seoul_bike_sharing,aes(x=DATE, y=RENTED_BIKE_COUNT, color = HOUR)) +
geom_point(alpha = 0.2) +
labs(title= "RENTED_BIKE_COUNT vs DATE TO HOURS",␣
 ↪x="DATE",y="RENTED_BIKE_COUNT")
```



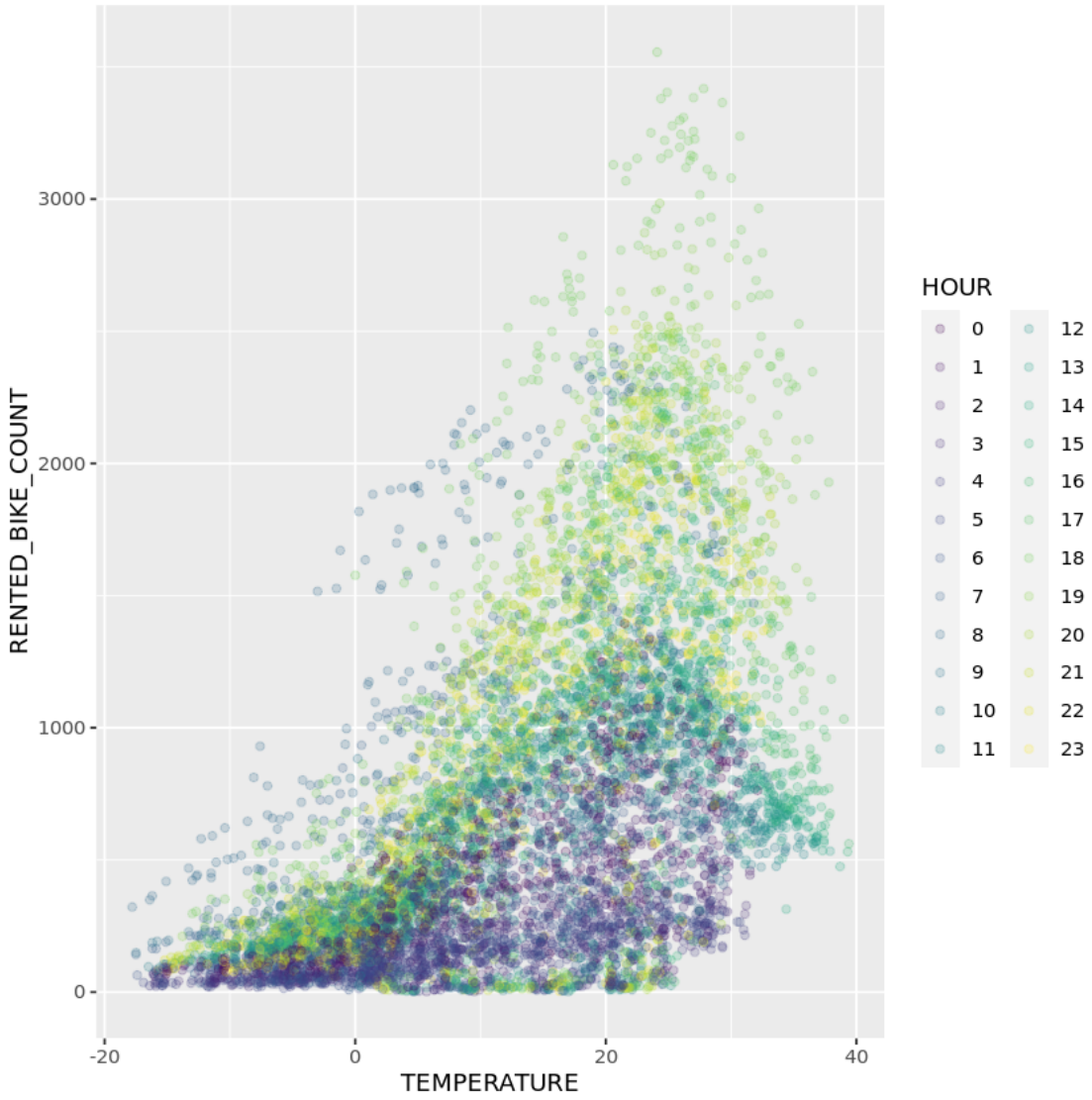
[17]:
```
#Task 12 - Create a histogram overlaid with a kernel density curve
ggplot(data= seoul_bike_sharing, aes(x = RENTED_BIKE_COUNT )) +
geom_histogram(bins=30, aes(y =..density..)) +
geom_density(colour = "black", fill = "white", alpha = 0.5) +
labs(title="DESITY CURVE")
```



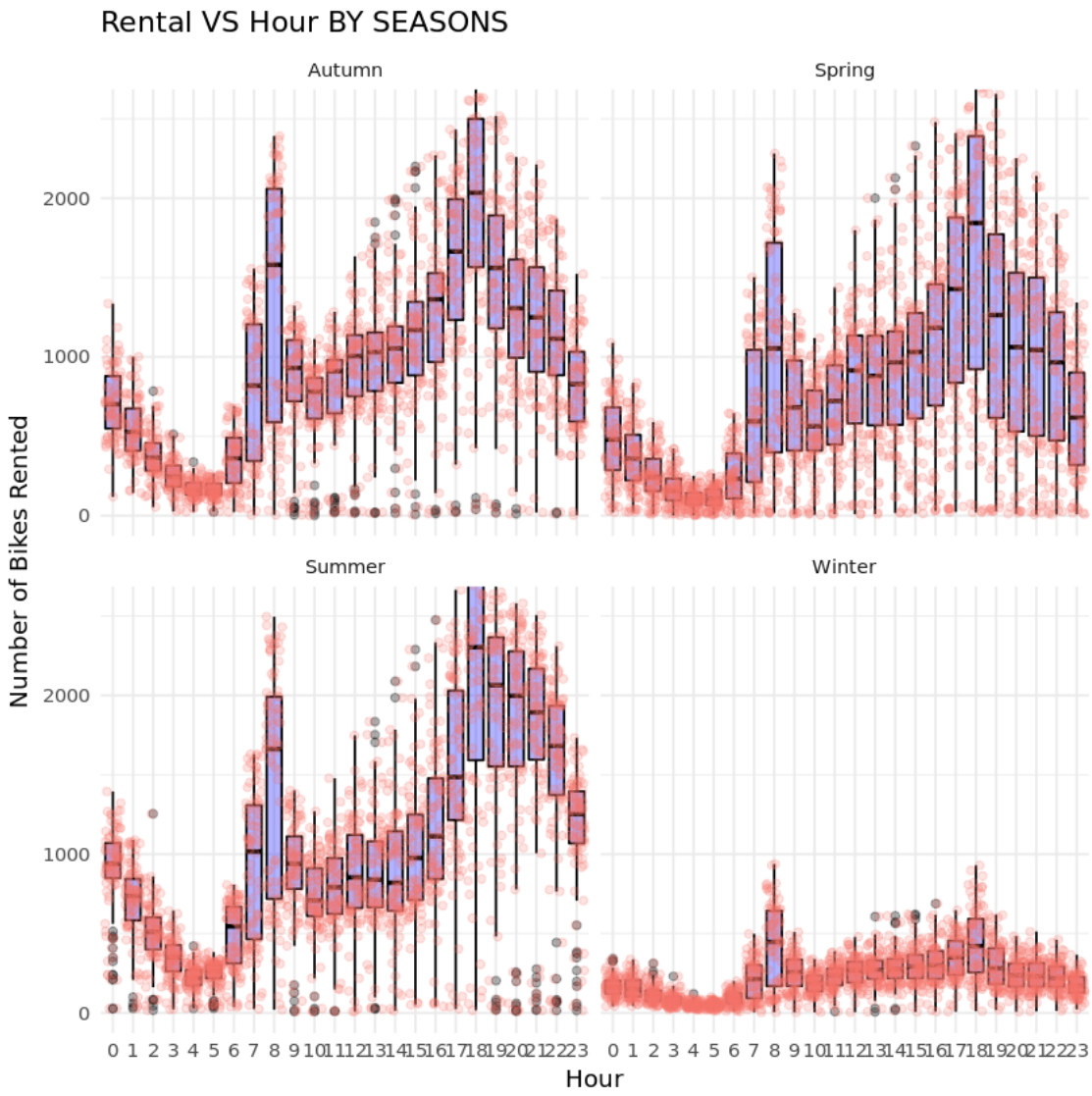
[18]:
```
#Task 13 - Use a scatter plot to visualize the correlation between␣
 ↪RENTED_BIKE_COUNT and TEMPERATURE
#by SEASONS
ggplot(data= seoul_bike_sharing, mapping= aes(x = TEMPERATURE, y =␣
 ↪RENTED_BIKE_COUNT, color = HOUR)) +
geom_point(alpha = 0.2) +
labs(title = "CORRELATION", x = "Temperature", y = "Rented Bikes Count") +
facet_wrap(~SEASONS)
```



[19]:
```
ggplot(seoul_bike_sharing) +
   geom_point(aes(x=TEMPERATURE,y=RENTED_BIKE_COUNT,colour=HOUR),alpha=1/5)
```



[20]:
```
##task 14 - Create a display of four boxplots of RENTED_BIKE_COUNT vs. HOUR␣
↪grouped by SEASONS.
ggplot(data= seoul_bike_sharing, mapping = aes(x=HOUR, y=RENTED_BIKE_COUNT)) +
geom_boxplot(fill = "blue",color = "black", alpha = 0.3) +
geom_jitter(aes(color = 'blue'), alpha=0.2) +
labs(x = "Hour", y = "Number of Bikes Rented") +
ggtitle("Rental VS Hour BY SEASONS")+
facet_wrap(~SEASONS)+
guides(color = FALSE) +
theme_minimal() +
coord_cartesian(ylim = quantile(seoul_bike_sharing$RENTED_BIKE_COUNT, c(0, 0.
↪99)))
```



[21]:
```
#Task 15 - Group the data by DATE, and use the summarize() function
#to calculate the daily total rainfall and snowfall
seoul_bike_sharing %>%
group_by(DATE) %>%
summarize(TOTAL_RAINFALL = sum(RAINFALL), TOTAL_SNOWFALL = sum(SNOWFALL)) %>%
slice(0:10)
```

A tibble: 10 × 3

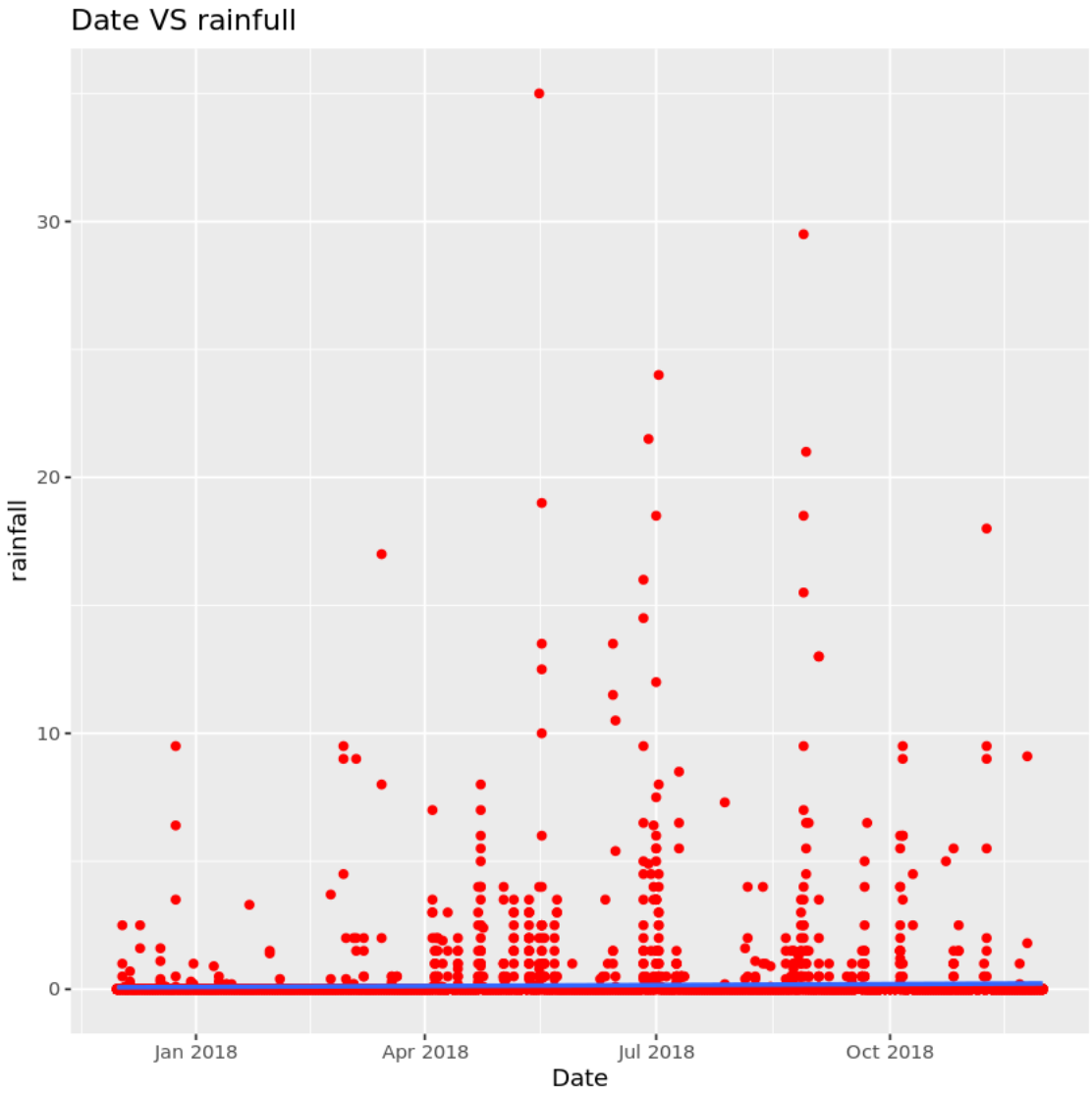
DATE <date>	TOTAL_RAINFALL <dbl>	TOTAL_SNOWFALL <dbl>
2017-12-01	0.0	0.0
2017-12-02	0.0	0.0
2017-12-03	4.0	0.0
2017-12-04	0.1	0.0
2017-12-05	0.0	0.0
2017-12-06	1.3	8.6
2017-12-07	0.0	10.4
2017-12-08	0.0	0.0
2017-12-09	0.0	0.0
2017-12-10	4.1	32.5

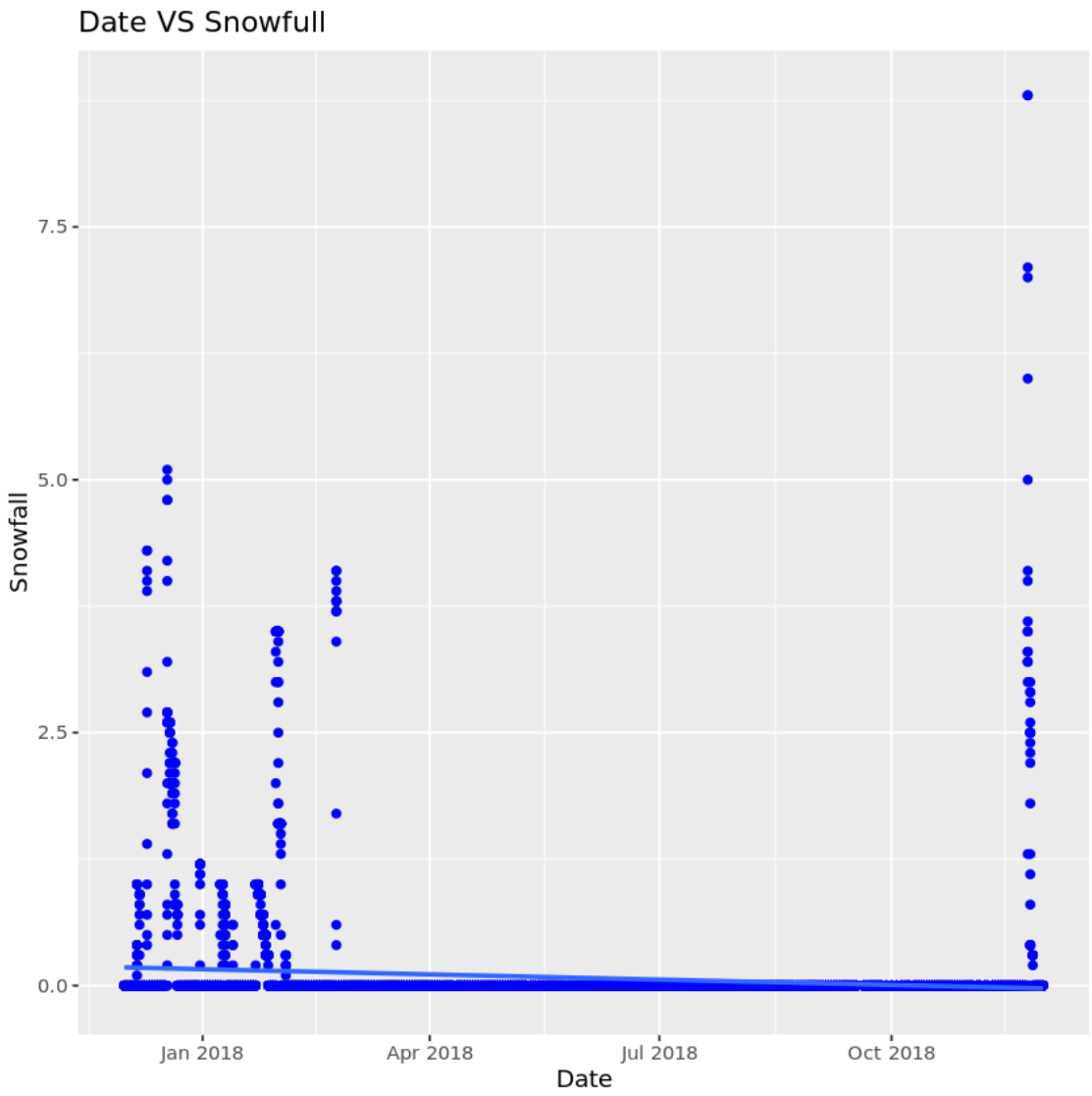
[22]:
```
ggplot(data = seoul_bike_sharing, mapping = aes(x = DATE, y = RAINFALL)) +
 geom_point(color="red") +
labs(title = "Date VS rainfull", x = "Date", y = "rainfall")+
 geom_smooth(method = "lm", na.rm = TRUE)

ggplot(data = seoul_bike_sharing, mapping = aes(x = DATE, y = SNOWFALL)) +
 geom_point(color="blue") +
labs(title = "Date VS Snowfull", x = "Date", y = "Snowfall")+
 geom_smooth(method = "lm", na.rm = TRUE)
```

```
`geom_smooth()` using formula 'y ~ x'
```

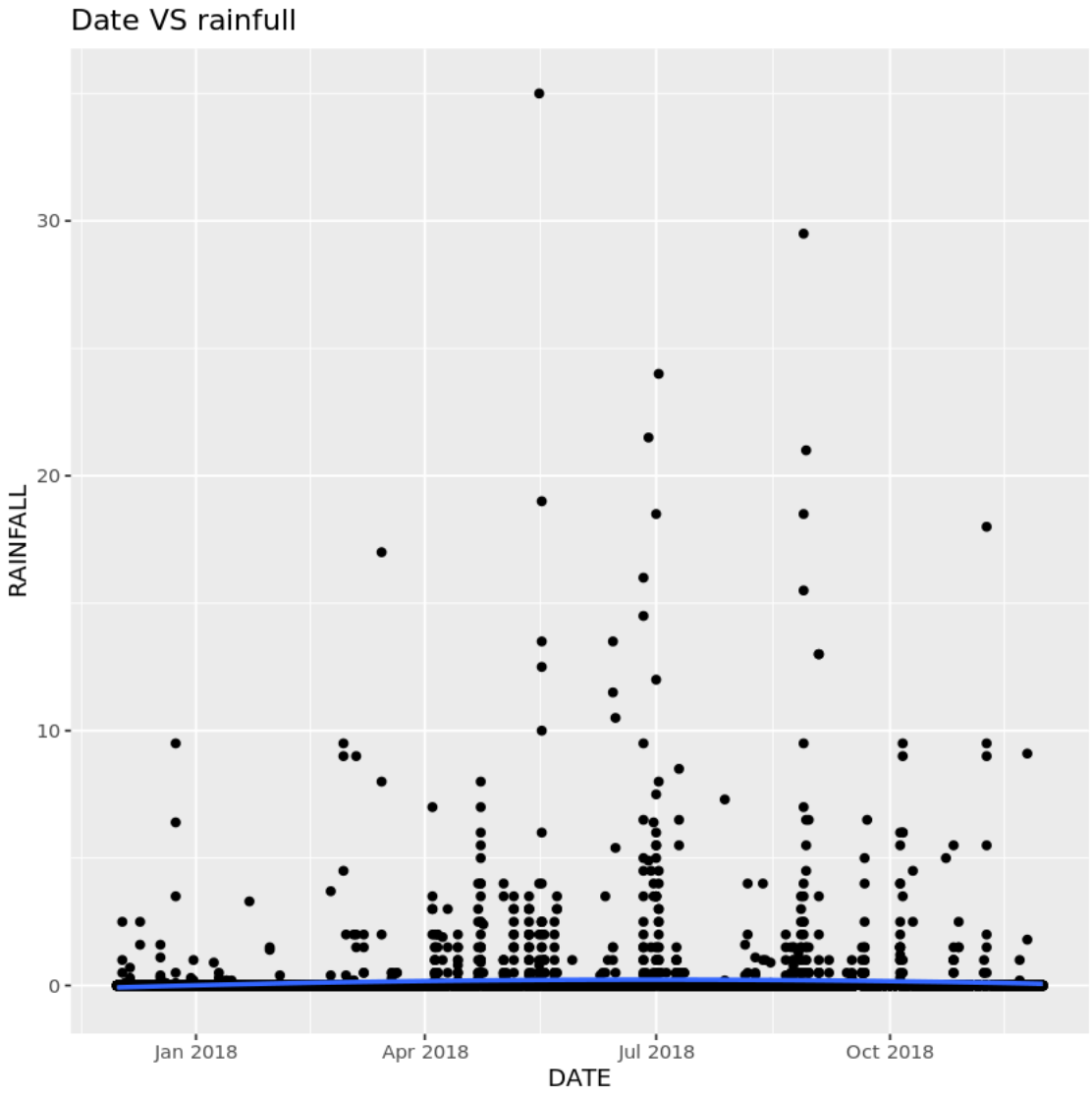
```
`geom_smooth()` using formula 'y ~ x'
```

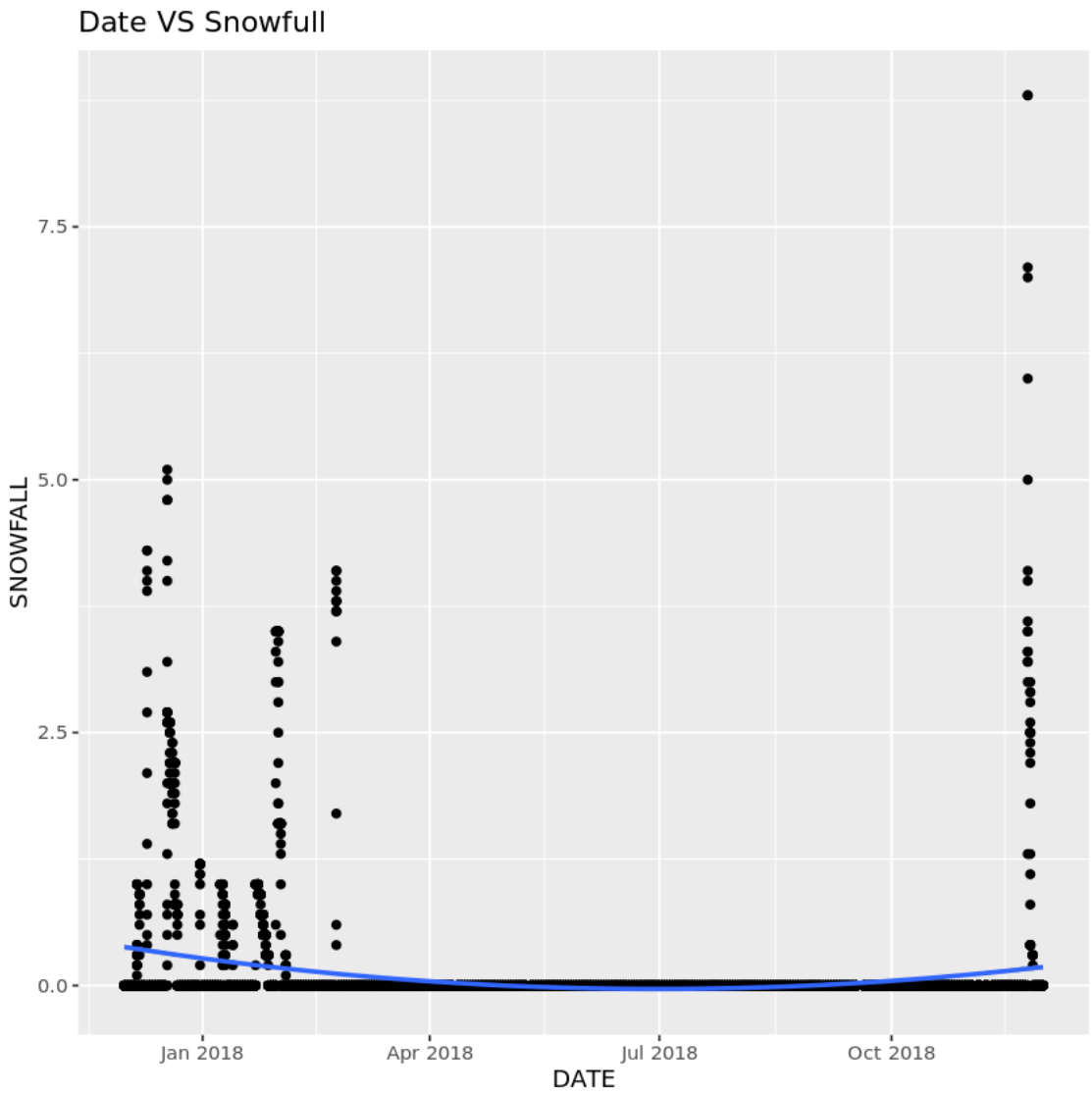

Date VS rainfull
rainfall
30
20
10
0
Jan 2018
Apr 2018
Jul 2018
Oct 2018
Date



[23]:
```
ggplot(data = seoul_bike_sharing,aes(DATE,RAINFALL)) +
 geom_point() +
labs(title = "Date VS rainfull")+
 geom_smooth(method = "lm", formula = y ~ poly(x, 2))

ggplot(data = seoul_bike_sharing,aes(DATE, SNOWFALL)) +
 geom_point() +
labs(title = "Date VS Snowfull")+
geom_smooth(method = "lm", formula = y ~ poly(x, 2))
```


Date VS rainfull
RAINFALL
30
20
10
0
Jan 2018
Apr 2018
Jul 2018
Oct 2018
DATE



[24]:
```
#Task 16 - Determine how many days had snowfall
snowfall_days <- seoul_bike_sharing %>%
group_by(DATE) %>%
filter(SNOWFALL != 0) %>%
summarize(TOTAL_SNOWFALL = sum(SNOWFALL))
(snowfall_days)
```

A tibble: 27 × 2

DATE <date>	TOTAL_SNOWFALL <dbl>
2017-12-06	8.6
2017-12-07	10.4
2017-12-10	32.5
2017-12-18	59.7
2017-12-19	55.6
2017-12-20	48.3
2017-12-21	38.9
2017-12-22	7.7
2017-12-31	14.3
2018-01-08	4.5
2018-01-09	10.8
2018-01-10	10.2
2018-01-13	2.2
2018-01-22	6.2
2018-01-23	23.3
2018-01-24	19.7
2018-01-25	14.2
2018-01-26	10.3
2018-01-27	4.4
2018-01-30	19.4
2018-01-31	64.8
2018-02-01	21.7
2018-02-03	2.2
2018-02-23	44.7
2018-11-24	78.7
2018-11-25	41.4
2018-11-26	2.9

[ ]:

[ ]: