Install packages

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflic
ts to become errors
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(ggplot2)
library(RColorBrewer)
library(patchwork)
```

```
##
## Attaching package: 'patchwork'
##
## The following object is masked from 'package:MASS':
##
##     area
```

Import data and setting graphic settings

```
sol_pow_gen = read.csv("sol_pow_gen.csv", sep = ",")

mycol = brewer.pal(5, "Set1")

head(sol_pow_gen)
```

```
##   Day.of.Year Year Month Day First.Hour.of.Period Is.Daylight
## 1         245 2008     9   1                    1       FALSE
## 2         245 2008     9   1                    4       FALSE
## 3         245 2008     9   1                    7        TRUE
## 4         245 2008     9   1                   10        TRUE
## 5         245 2008     9   1                   13        TRUE
## 6         245 2008     9   1                   16        TRUE
##   Distance.to.Solar.Noon Average.Temperature..Day. Average.Wind.Direction..Day.
## 1              0.8598972                        69                           28
## 2              0.6285347                        69                           28
## 3              0.3971722                        69                           28
## 4              0.1658098                        69                           28
## 5              0.0655527                        69                           28
## 6              0.2969152                        69                           28
##   Average.Wind.Speed..Day. Sky.Cover Visibility Relative.Humidity
## 1                      7.5         0         10                75
## 2                      7.5         0         10                77
## 3                      7.5         0         10                70
## 4                      7.5         0         10                33
## 5                      7.5         0         10                21
## 6                      7.5         0         10                20
##   Average.Wind.Speed..Period. Average.Barometric.Pressure..Period.
## 1                           8                                29.82
## 2                           5                                29.85
## 3                           0                                29.89
## 4                           0                                29.91
## 5                           3                                29.89
## 6                          23                                29.85
##   Power.Generated
## 1               0
## 2               0
## 3            5418
## 4           25477
## 5           30069
## 6           16280
```

The goal is to introduce the nature of the data, including key descriptive statistics, and to then apply analytical tools. Our goal would be to predict the amount of power generated using all the other variables.

Data description

At what time the data is being recorded ?

```
table(sol_pow_gen$First.Hour.of.Period)
```

```
##
##   1   4   7  10  13  16  19  22
## 365 365 365 365 365 365 365 365
```

```
table(sol_pow_gen$Year)
```

```
##
## 2008 2009
##  976 1944
```

```
sol_pow_gen$Month[1]
```

```
## [1] 9
```

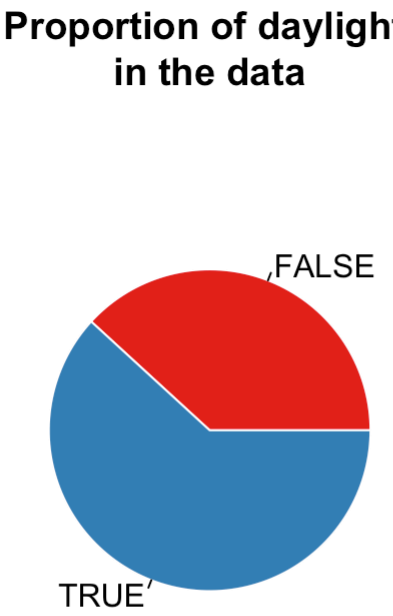
```
sol_pow_gen$Month[length(sol_pow_gen$Month)]
```

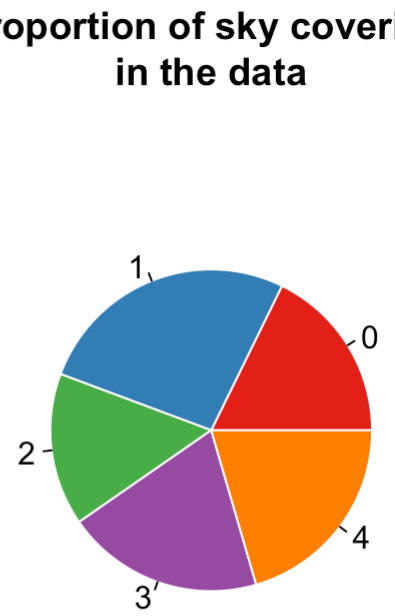
```
## [1] 8
```

We have data at 1, 4, 7, 10, 13, 16, 19 and 22 for all the days of the year. The time period is between september 2008 and october 2009

Proportion of daylight in the data along with the proportion of sky covering in the data

```
par(mfrow = c(1,2))
pie(table(sol_pow_gen$Is.Daylight), border = "white", col = mycol, main = "Proportion
of daylight \nin the data")
pie(table(sol_pow_gen$Sky.Cover), border = "white", col = mycol, main = "Proportion o
f sky covering \nin the data")
```





For the average temperature in the day As it is in F°, we want to convert it to Celsius

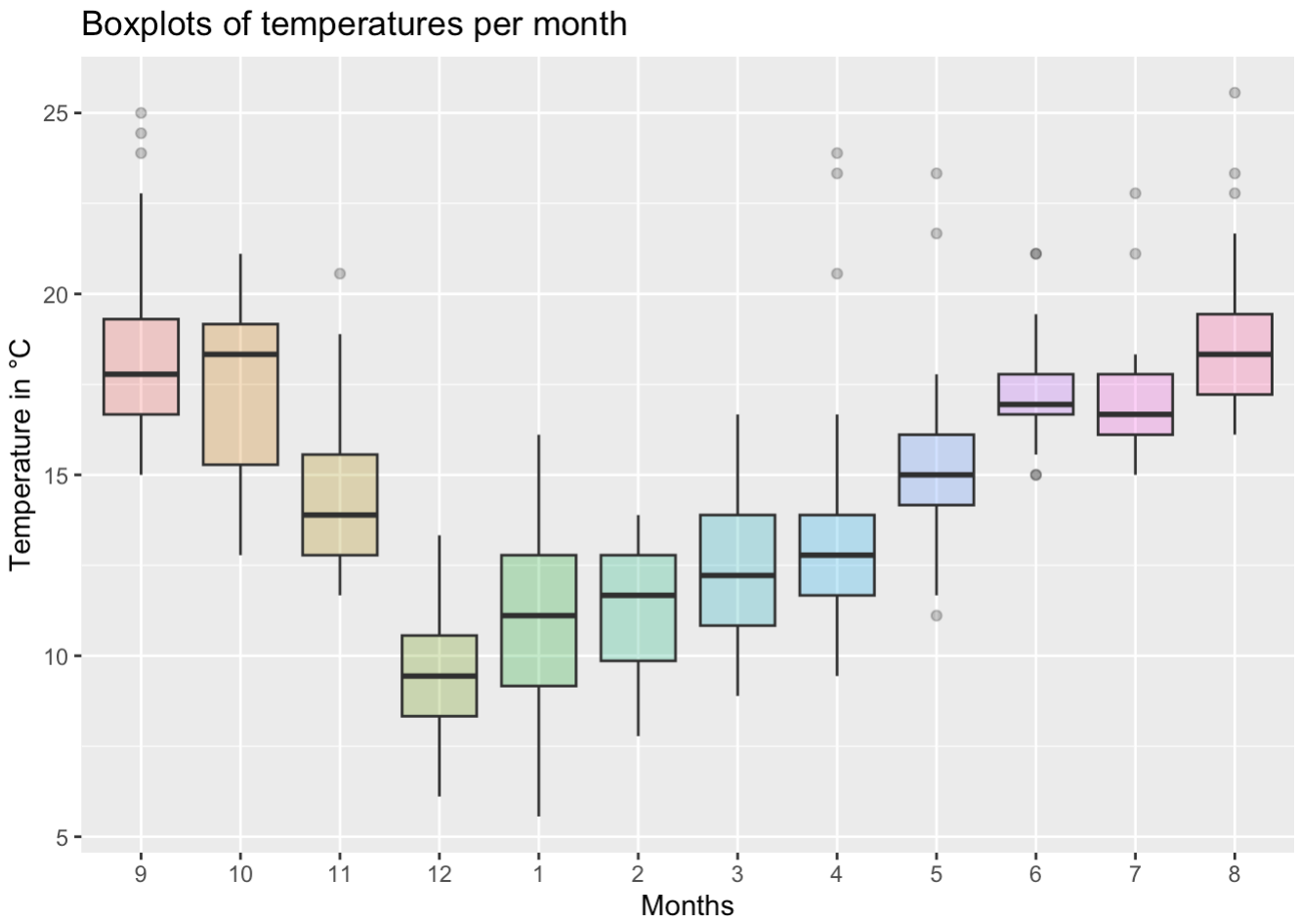
```
sol_pow_gen = read.csv("sol_pow_gen.csv", sep = ",") #to avoid modifying the column a
gain and again
sol_pow_gen$Average.Temperature..Day. = round((sol_pow_gen$Average.Temperature..Day.
- 32) * (5/9), 2)
```

We then want to visualise it

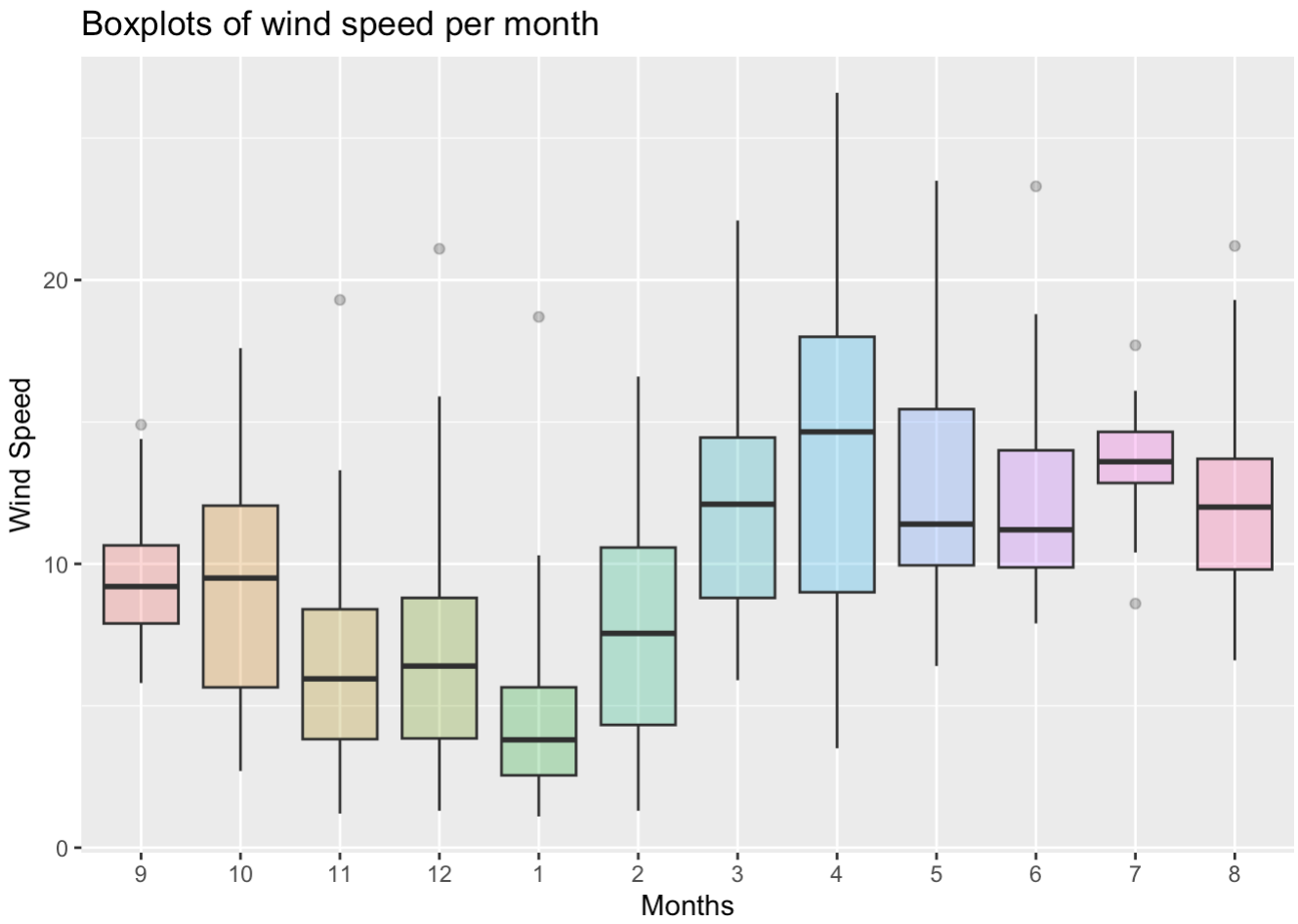
```
data_day = sol_pow_gen %>%
  group_by(Year, Month, Day) %>%
  summarise(avg_temp_day = first(Average.Temperature..Day.), avg_wind_day = first(Ave
rage.Wind.Speed..Day.), avg_humidity = mean(Relative.Humidity))
```

```
## `summarise()` has grouped output by 'Year', 'Month'. You can override using the
## `.groups` argument.
```

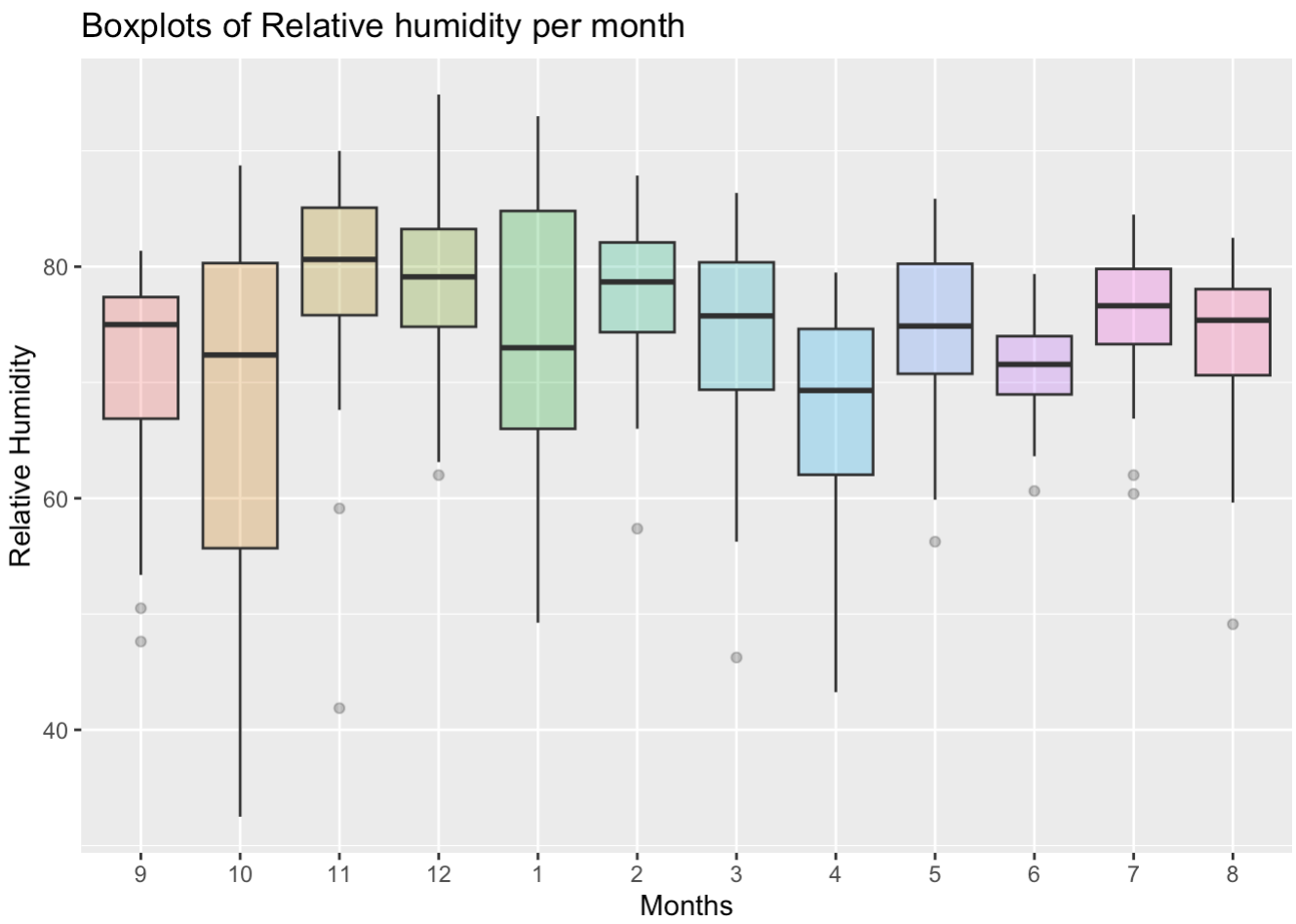
```
data_day$Month = factor(data_day$Month, levels = c(9, 10, 11, 12, 1, 2, 3, 4, 5, 6,
7, 8))

ggplot(data = data_day, aes(x= Month, y = avg_temp_day, fill = Month)) +
  geom_boxplot(alpha=0.3) +
  labs(title = "Boxplots of temperatures per month", x = "Months", y = "Temperature i
n °C") +
  theme(legend.position = "none")
```




```
ggplot(data = data_day, aes(x= Month, y = avg_wind_day, fill = Month)) +
  geom_boxplot(alpha=0.3) +
  xlab("Months") +
  ylab("Wind Speed") +
  ggtitle("Boxplots of wind speed per month") +
  theme(legend.position = "none")
```

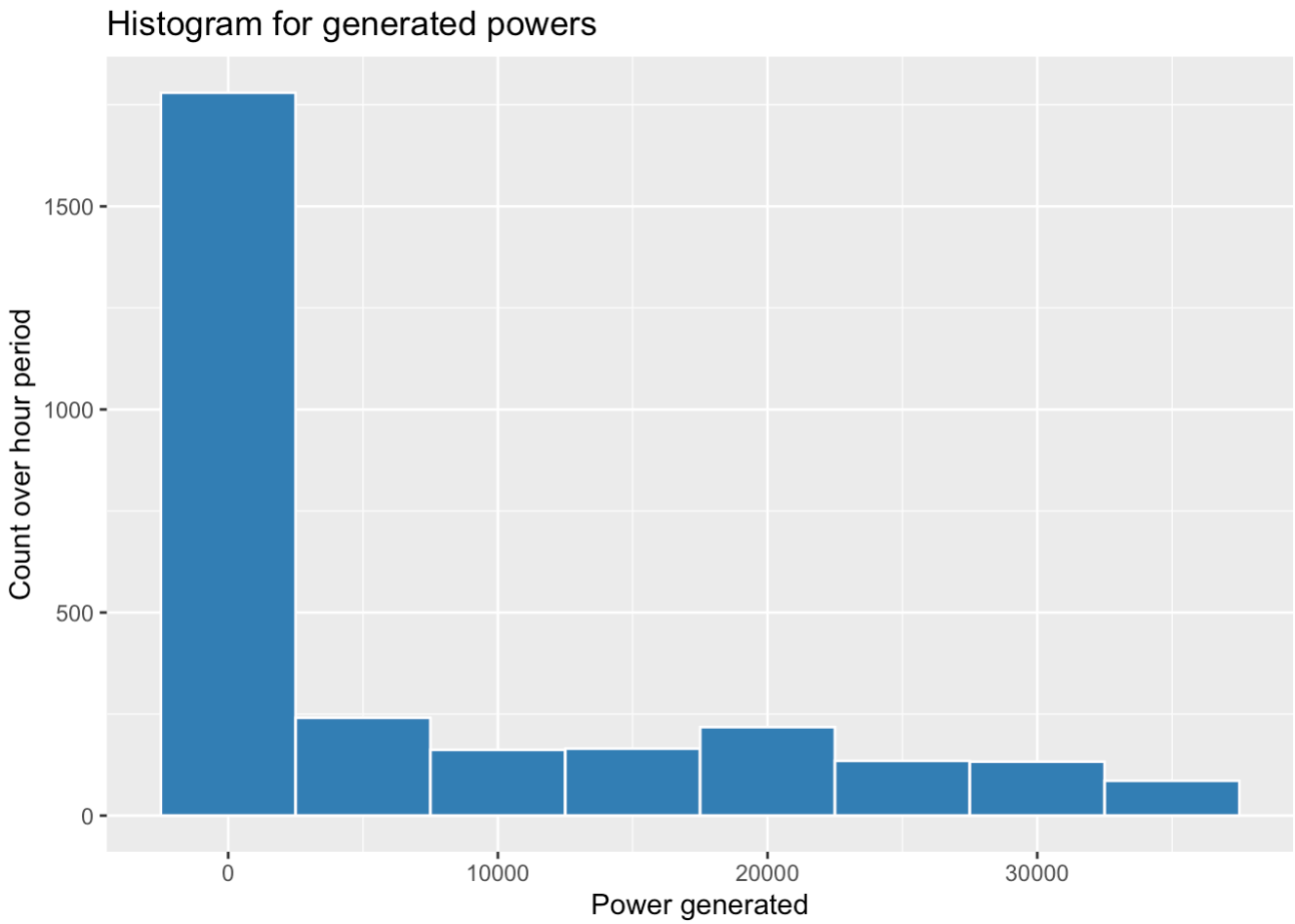



```
ggplot(data = data_day, aes(x= Month, y = avg_humidity, fill = Month)) +
  geom_boxplot(alpha=0.3) +
  xlab("Months") +
  ylab("Relative Humidity") +
  ggtitle("Boxplots of Relative humidity per month") +
  theme(legend.position = "none")
```



Let’s take a look at the variable of interest : the generated power

```
ggplot(sol_pow_gen, aes(x = Power.Generated)) +
  geom_histogram(binwidth = 5000, color = "white", fill = mycol[2]) +
  labs(title = "Histogram for generated powers", x = "Power generated", y = "Count ov
er hour period")
```



Using our results in Python, we know distance to solar and power generated are negatively correlated

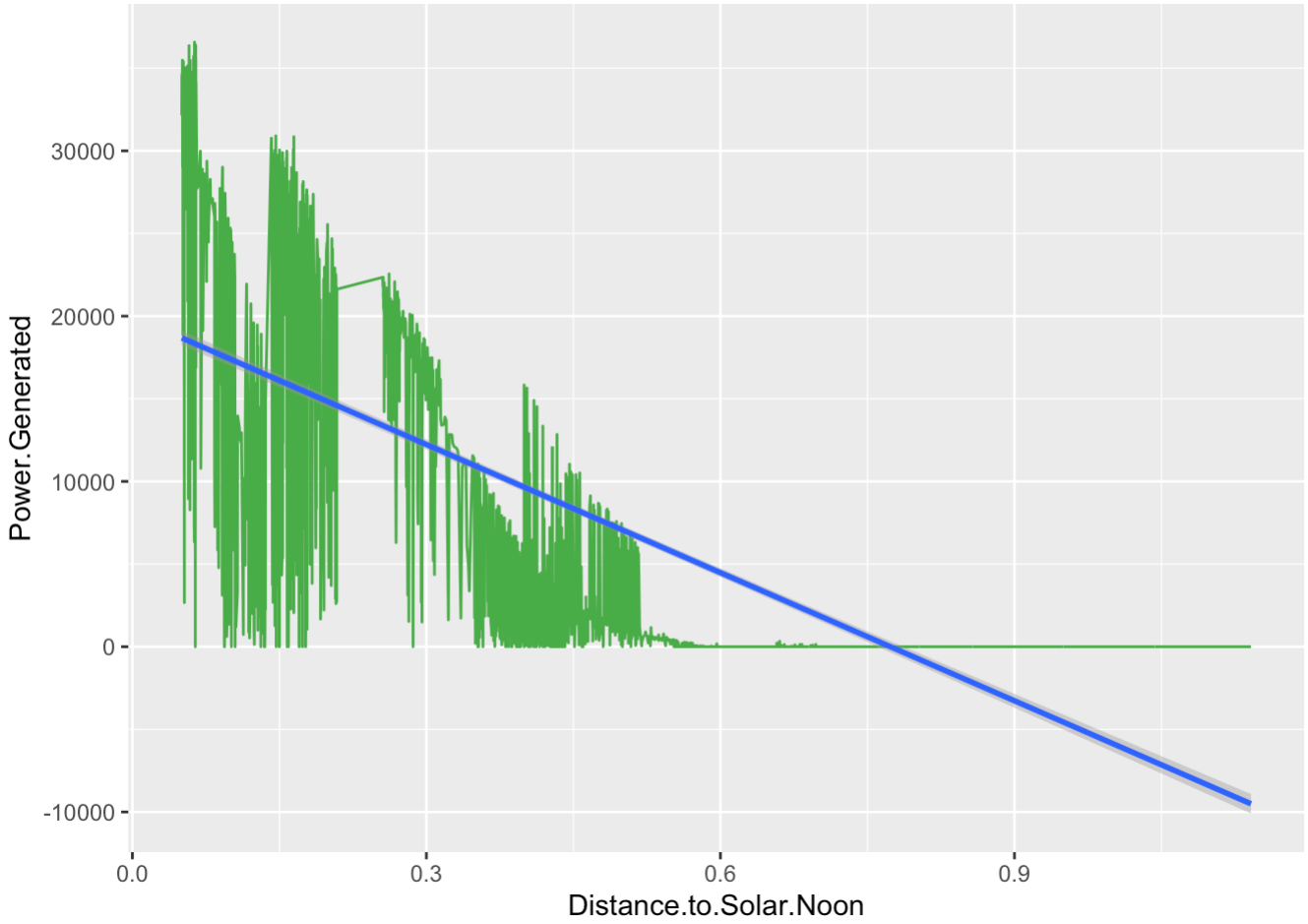
```
cor(sol_pow_gen$Distance.to.Solar.Noon, sol_pow_gen$Power.Generated)
```

```
## [1] -0.7466814
```

We could try and make a linear regression, but as seen in Python, its not going to be relevant

```
data_ord_dist = sol_pow_gen[order(sol_pow_gen$Distance.to.Solar.Noon),c(7,16)]

ggplot(data = data_ord_dist, aes(Distance.to.Solar.Noon, Power.Generated)) +
  geom_line(col = mycol[3]) +
  geom_smooth(method = "lm")
```

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




```
linear_model = lm(data_ord_dist$Distance.to.Solar.Noon~data_ord_dist$Power.Generated)
summary(linear_model)
```

```
##
## Call:
## lm(formula = data_ord_dist$Distance.to.Solar.Noon ~ data_ord_dist$Power.Generated)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58971 -0.13840  0.01188  0.11329  0.48745
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    6.539e-01  4.431e-03  147.58   <2e-16 ***
## data_ord_dist$Power.Generated -2.158e-05  3.559e-07  -60.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1983 on 2918 degrees of freedom
## Multiple R-squared:  0.5575, Adjusted R-squared:  0.5574
## F-statistic:  3677 on 1 and 2918 DF,  p-value: < 2.2e-16
```

Time series

Trials and errors