Solar power generation data set

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(corrplot)
```

```
## corrplot 0.95 loaded
```

```
library(klaR)
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(devtools)
```

```
## Loading required package: usethis
```

```
library(patchwork)
```

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

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

Import data and setting graphic settings

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

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

data = na.omit(data)

#We put the date and time variables in one column only
day = as.Date(paste(data$Year, data$Month, data$Day), format = "%Y %m %d")
dates = as.POSIXct(paste(data$Year, data$Month, data$Day, paste0(data$First.Hour.of.P
eriod, ":00:00")), format = "%Y %m %d %H", tz = "US/Pacific")
data = data[,-c(1, 2, 3, 4, 5)]
data = cbind(day, dates, data)

#Converting °F in °C
data$Average.Temperature..Day. = round((data$Average.Temperature..Day. - 32) * (5/9),
2)

colnames(data) = c("day", "dates", "daylight", "dist_sol_noon", "avg_temp", "avg_wind
_dir", "avg_wind_speed", "sky_cov", "vis", "rel_humid", "avg_wind_speed_per", "avg_pr
ess", "pow_gen")

head(data)
```

```
##          day               dates daylight dist_sol_noon avg_temp avg_wind_dir
## 1 2008-09-01 2008-09-01 01:00:00    FALSE     0.8598972    20.56           28
## 2 2008-09-01 2008-09-01 04:00:00    FALSE     0.6285347    20.56           28
## 3 2008-09-01 2008-09-01 07:00:00     TRUE     0.3971722    20.56           28
## 4 2008-09-01 2008-09-01 10:00:00     TRUE     0.1658098    20.56           28
## 5 2008-09-01 2008-09-01 13:00:00     TRUE     0.0655527    20.56           28
## 6 2008-09-01 2008-09-01 16:00:00     TRUE     0.2969152    20.56           28
##   avg_wind_speed sky_cov vis rel_humid avg_wind_speed_per avg_press pow_gen
## 1            7.5       0  10        75                  8     29.82       0
## 2            7.5       0  10        77                  5     29.85       0
## 3            7.5       0  10        70                  0     29.89    5418
## 4            7.5       0  10        33                  0     29.91   25477
## 5            7.5       0  10        21                  3     29.89   30069
## 6            7.5       0  10        20                 23     29.85   16280
```

Data description

The data variables :

Distance to Solar Noon : distance to the sun at noon (supposedly normalized around 1)

Average Temperature (Day) : the average temperature in the day (previously in °F, converted in °C)

Average Wind Direction (Day) : average direction of the wind // discrete “qualitative” variable with 36 modalities (36 directions possible)

Average Wind Speed (Day) : average wind speed (supposedly in km/h) // continuous quantitative

Sky Cover : sky coverage by clouds // qualitative variable with 5 modalities (0 is no coverage, 4 is a lot of coverage)

Relative Humidity : humidity of the day // quantitative variable, on a scale of 0 to 100%

Power Generated : power generated by the power plant in kW in the day // dependent variable, quantitative

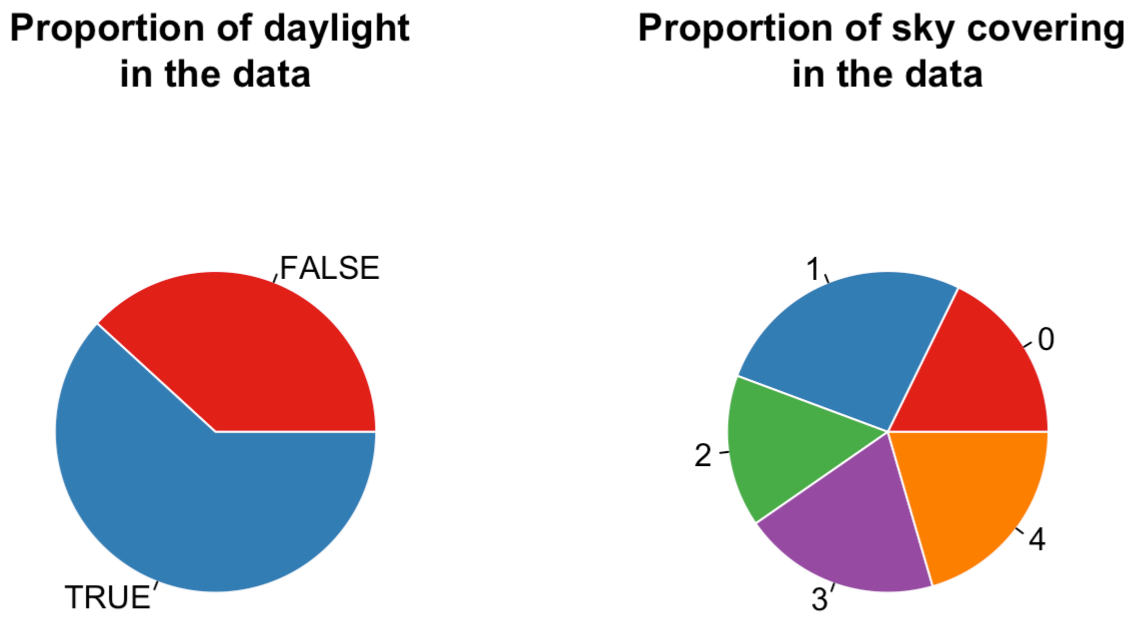
At what time the data is being recorded ?

```
cat(sprintf("The starting date of our data is at %s and it ends at %s\n", min(dates),
max(dates)))
```

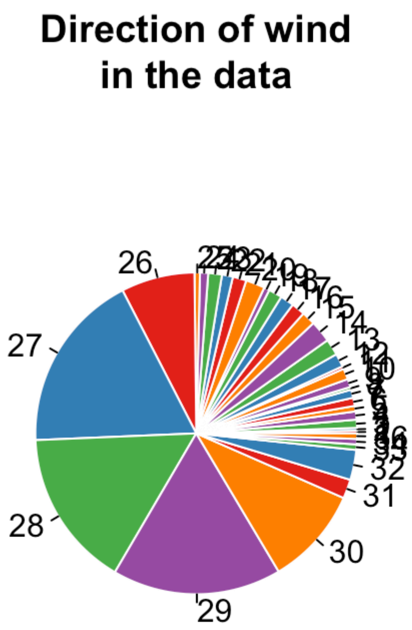
```
## The starting date of our data is at 2008-09-01 01:00:00 and it ends at 2009-08-31
22:00:00
```

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

```
#The pieplots
par(mfrow = c(1,2))
pie(table(data$daylight), border = "white", col = mycol, main = "Proportion of daylig
ht \nin the data")
pie(table(data$sky_cov), border = "white", col = mycol, main = "Proportion of sky cov
ering \nin the data")
```



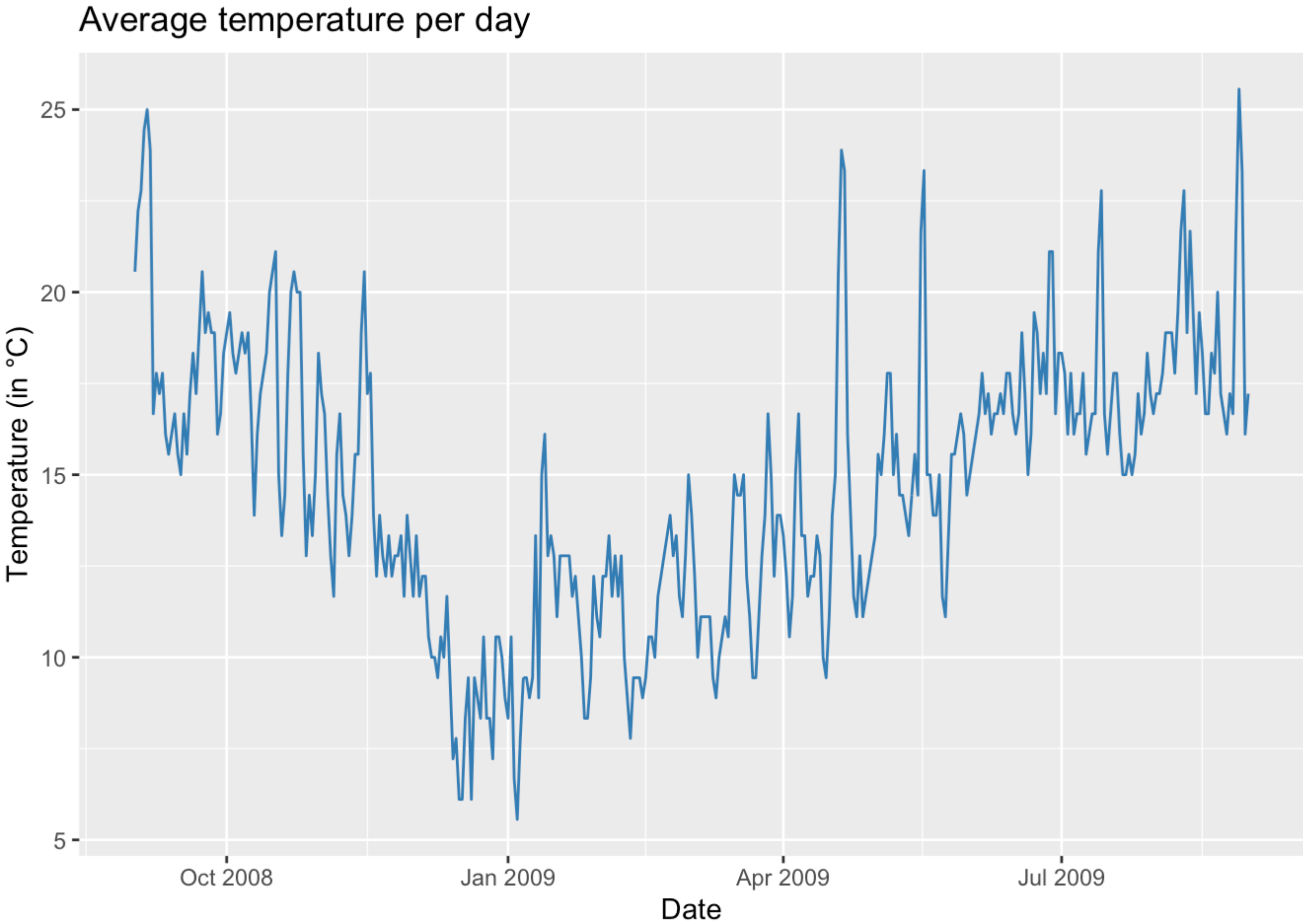

```
pie(table(data$avg_wind_dir), border = "white", col = mycol, main = "Direction of win
d\nin the data")
```



For the average temperature in the day

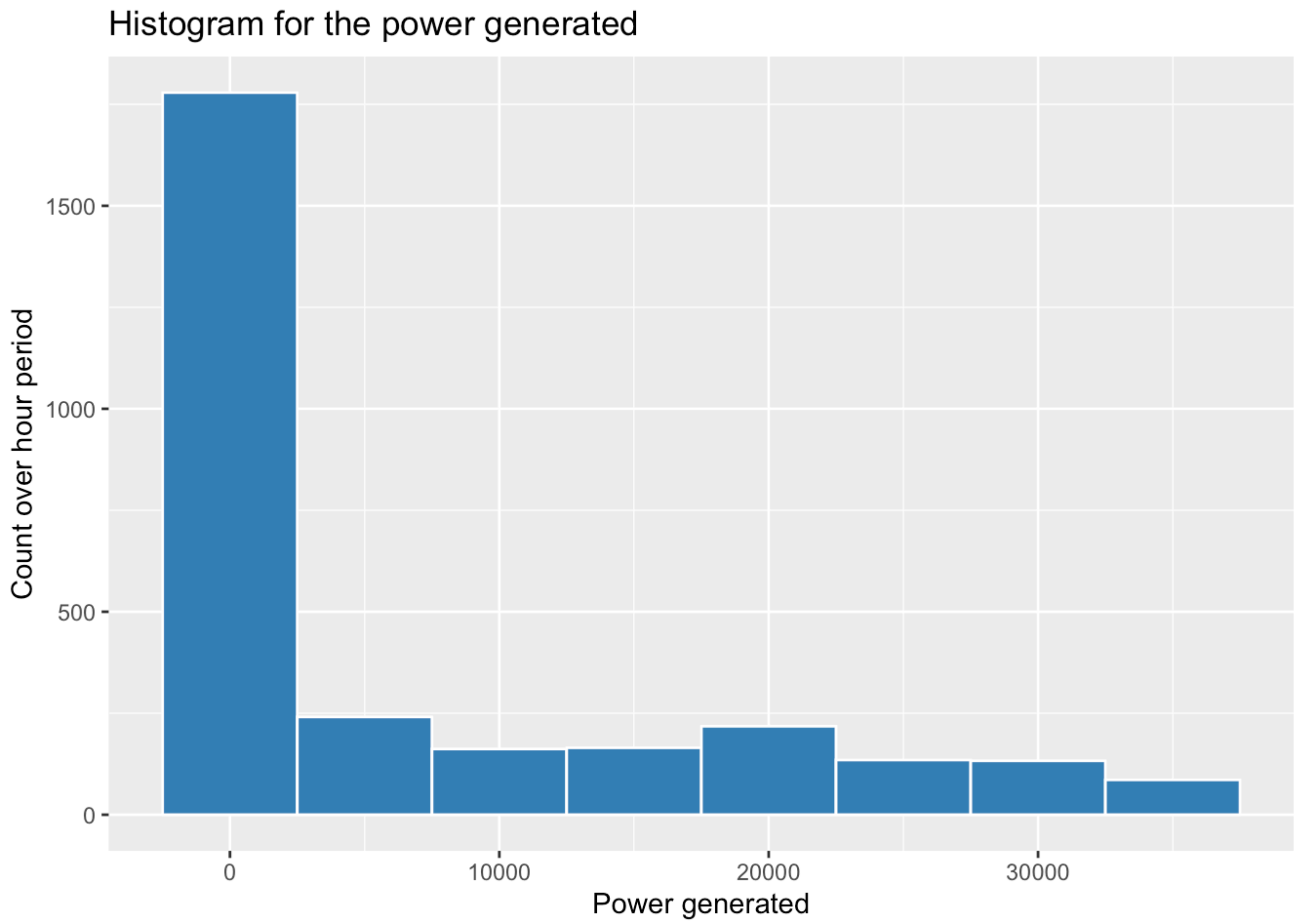
```
#We regroup the avg temperature for each day as it is just repeated in the dataset (f
or the time period)
data_avg_temp = data %>%
  group_by(day) %>%
  summarize(avg_temp = mean(avg_temp))

ggplot(data = data_avg_temp, aes(x = day, y = avg_temp)) +
  geom_line(col = mycol[2]) +
  labs(title = "Average temperature per day",
       x = "Date",
       y = "Temperature (in °C)")
```



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

```
ggplot(data, aes(x = pow_gen)) +
  geom_histogram(binwidth = 5000, color = "white", fill = mycol[2]) +
  labs(title = "Histogram for the power generated", x = "Power generated", y = "Count
over hour period")
```



We did most of the visualization in Python, so let's dive in the data analysis for now

Data analysis

Based on our findings in Python, we saw that the distance to solar noon mean was a lot lower when there was power generated than when there was not. Let's investigate this

```
#The data of interest
dist_sol_no_pow = data[data$pow_gen==0,]$dist_sol_noon
dist_sol_pow = data[data$pow_gen!=0,]$dist_sol_noon

sprintf("The distance to solar noon mean for the subgroup of 0 power generated is of %s", round(mean(dist_sol_no_pow), 3))
```

```
## [1] "The distance to solar noon mean for the subgroup of 0 power generated is of 0.771"
```

```
sprintf("The distance to solar noon mean for the subgroup of above 0 power generated is of %s", round(mean(dist_sol_pow), 3))
```

```
## [1] "The distance to solar noon mean for the subgroup of above 0 power generated i
s of 0.282"
```

Let's perform a t-test IV : Power generated, qualitative, 2 modalities : 0 or >0 DV : Distance to solar noon, quantitative, continuous

H0 : there is no difference between the mean of the distance for no generated power and the distance for >0 generated power H1 : there is a difference

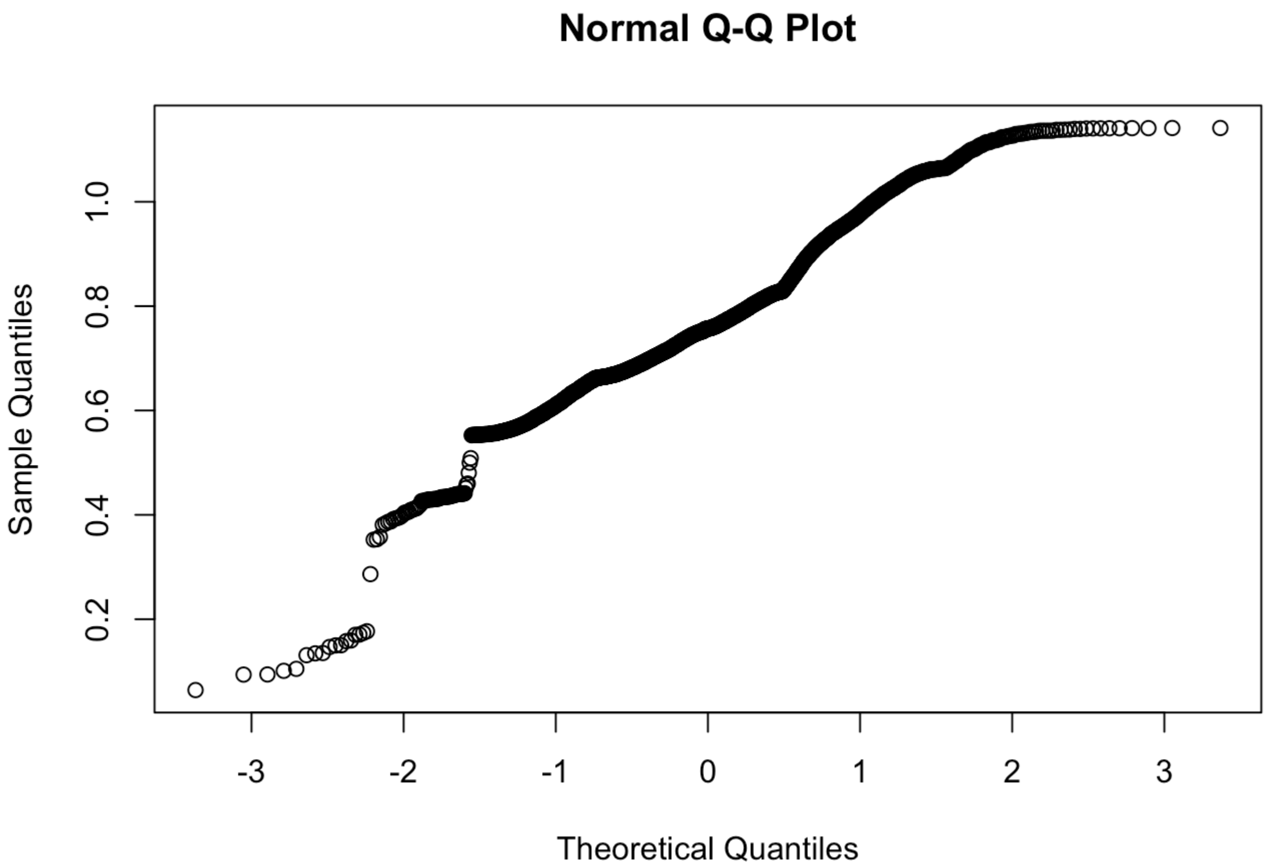
```
#Let's check assumptions first ; is the data normally distributed
shapiro.test(dist_sol_no_pow)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dist_sol_no_pow
## W = 0.97179, p-value = 2.209e-15
```

```
shapiro.test(dist_sol_pow)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dist_sol_pow
## W = 0.93941, p-value < 2.2e-16
```

```
qqnorm(dist_sol_no_pow) #It is not normally distributed at all (p-value well below 0.
001, qqplot not following the diagonal)
```




```
#We can still go about the test, but we know that we cannot take the result into acco
unt
```

```
#The t-test
t.test(dist_sol_no_pow, dist_sol_pow, alternative = c("greater"), conf.level = 0.95)
#of course we get a very small p-value. If the test was valid, we could reject H0 and accept H1
```

```
##
##  Welch Two Sample t-test
##
## data:  dist_sol_no_pow and dist_sol_pow
## t = 75.175, df = 2623.9, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.4778685       Inf
## sample estimates:
## mean of x mean of y
## 0.7709570 0.2823948
```

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

```
#The correlation coefficient between the 2 variables
cor(data$dist_sol_noon, data$pow_gen)
```

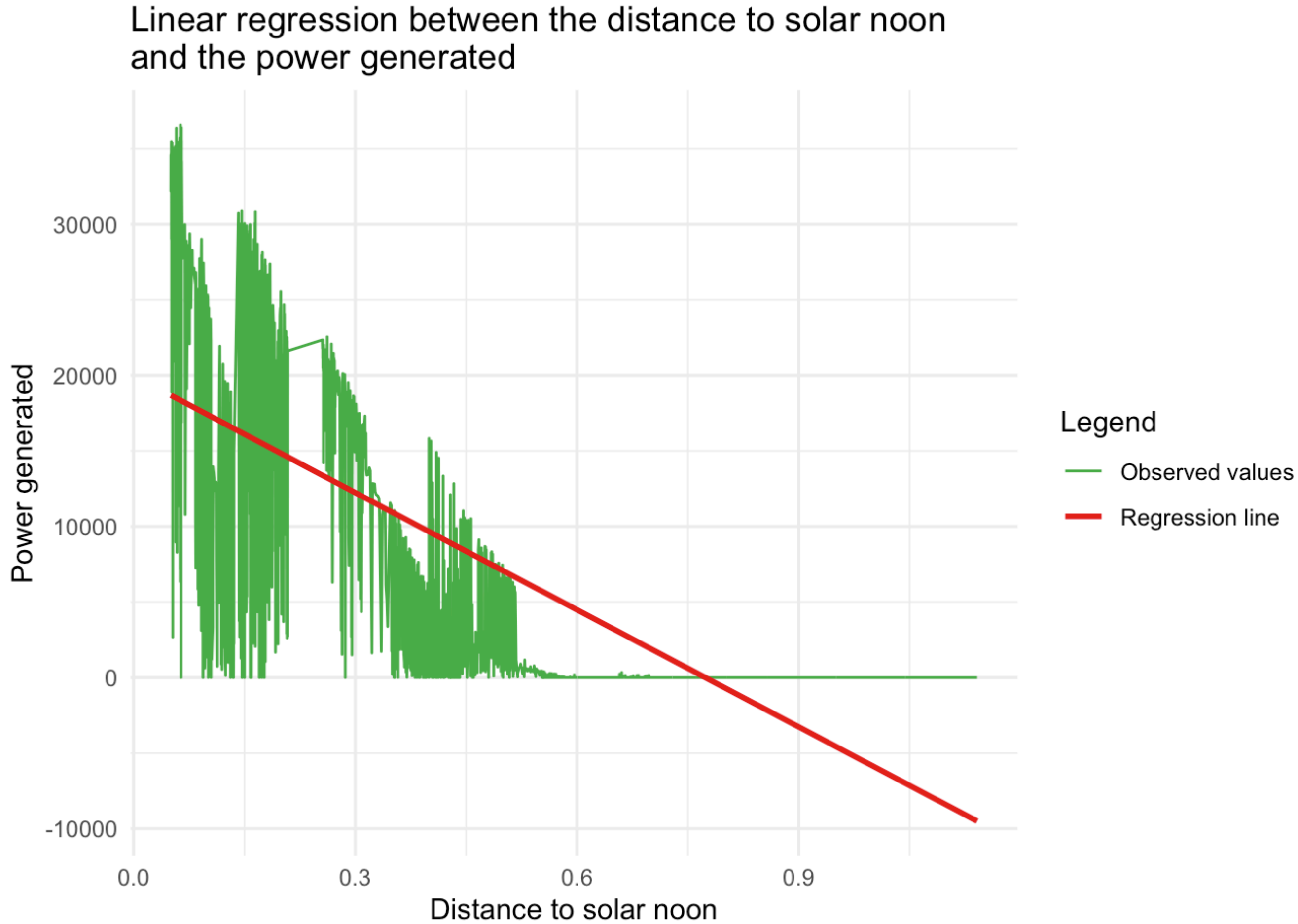
```
## [1] -0.7468249
```

Let's see if the linear regression has interesting coefficients

```
#The linear regression
data_ord_dist = data[order(data$dist_sol_noon),]

ggplot(data = data, aes(x = dist_sol_noon, y = pow_gen)) +
  geom_line(aes(color = "Observed values")) +
  geom_smooth(method = "lm", aes(color = "Regression line"), se = FALSE) +
  scale_color_manual(values = c("Observed values" = mycol[3], "Regression line" = mycol[1])) +
  labs(title = "Linear regression between the distance to solar noon\nand the power generated",
       x = "Distance to solar noon",
       y = "Power generated",
       color = "Legend") +
  theme_minimal()
```

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




```
linear_model = lm(data_ord_dist$dist_sol_noon~data_ord_dist$pow_gen)
summary(linear_model) #We observe an R2 of 0.5576, which is not that good
```

```
##
## Call:
## lm(formula = data_ord_dist$dist_sol_noon ~ data_ord_dist$pow_gen)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58983 -0.13835  0.01188  0.11336  0.48733
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            6.540e-01  4.432e-03  147.58   <2e-16 ***
## data_ord_dist$pow_gen -2.158e-05  3.559e-07  -60.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1983 on 2917 degrees of freedom
## Multiple R-squared:  0.5577, Adjusted R-squared:  0.5576
## F-statistic:  3679 on 1 and 2917 DF,  p-value: < 2.2e-16
```