

**GIPUZKOAKO INGENIARITZA ESKOLA**  
**ESCUELA DE INGENIERÍA DE GIPUZKOA**

**EIBAR**

---

**TFG: INTEGRACIÓN DE TURBINAS EÓLICAS EN EDIFICIOS I**

---

**DOCUMENTO Nº 2: CÁLCULOS**

**Grado: GRADO EN INGENIERÍA DE ENERGÍAS RENOVABLES**

**Curso: 2017 - 2018**

**Autor/a: ÓSCAR GARCÍA HERNÁNDEZ**

**Director/a/s: ALAIN ULAZIA MANTEROLA**

# Integración de turbinas en edificios I

*Óscar García Hernández*

*Septiembre de 2018*

## Introducción

En el presente documento se presentará el código empleado para el tratamiento de los datos obtenidos en las pruebas. Cabe mencionar que el código empleado se ha estructurado básicamente en 2 scripts. Uno en el que definimos todas las funciones creadas y otro script en el que se aplican todas estas funciones para obtener las gráficas y los resultados. Además, se han añadido dos scripts adicionales: un para el tratamiento de valores atípicos en la medición de la velocidad de giro y otro para los cálculos necesarios para el “ejemplo” en el edificio de bilbao.

## Código empleado

### Script de funciones

#### Paquetes usados

```
library(stringr)
library(magrittr)
library(here)
library(dplyr)
library(ggplot2)
library(minpack.lm)
```

### Función de procesamiento principal

Esta función busca las carpetas con los datos de las pruebas y las organiza en una tabla.

```
procesador <- function(path_to_csv){

  path_to_txt <- str_replace(path_to_csv, pattern = 'csv', replacement = 'txt')

  path_to_speed <- str_split(path_to_csv, '/', simplify = TRUE)
  path_to_speed <- path_to_speed[-length(path_to_speed)]
  path_to_speed <- paste(path_to_speed, collapse='/')

  path_to_speed_mean <- paste(path_to_speed, "Vviento_mean.txt", sep = '/')
  path_to_speed <- paste(path_to_speed, "Vviento.txt", sep = '/')

  if(!file.exists(path_to_speed_mean)){

    speed_mean <- mean(read.table(path_to_speed)[,1])
    write.table(speed_mean, file = path_to_speed_mean)
  }else{
    speed_mean <- read.table(path_to_speed_mean)
    if(is.na(speed_mean)){
      speed_mean <- mean(read.table(path_to_speed)[,1])
      write.table(speed_mean, file = path_to_speed_mean)
    }
  }
}
```

```

    } else{
      speed_mean <- speed_mean[1,1]
    }
  }

rpm<-colMeans(as.data.frame(read.table(path_to_txt)))

y<-read.table(path_to_csv, sep =", " )
# El CSV que ofrece el BK es kk esta todo separado por comas
volts_amps<- colMeans(as.data.frame(cbind(as.numeric(paste0(y[,1], ".", y[,2])),
                                          as.numeric(y[,3]+y[,4]*10^(-4))))))

##introduzco las velocidades del experimento piloto: Vel
processed_file <-as.data.frame( cbind(rpm[1],speed_mean,
                                     volts_amps[1], volts_amps[2]))
names(processed_file)<- c("RPM", "m/s", "V", "A")

return(processed_file)
}

```

## Función de creación de variables

Esta función crea las columnas de potencia, Cp y TSR dentro de la tabla principal

```

df_mutate<-function(tabla_cruda){
  df<- tabla_cruda
  df %<>% mutate(experimento= factor(
    ifelse(str_detect(archivos, 'concentrador'), 'concentrador',
            ifelse(str_detect(archivos, 'piloto'), 'piloto', 'pared')
          )
    )
  )
  df %<>% mutate(angulo= str_extract(archivos,
                                     pattern = "concentrador/[0-9]+") )
  %>%
    mutate(angulo = factor(str_remove(angulo,
                                       pattern = "concentrador/")),
           porcentaje=str_extract(archivos,
                                   pattern = "prueba_[0-9]+"))
  %>%
    mutate(porcentaje=factor(str_remove(porcentaje,
                                         pattern = "prueba_")))
  %>%
    mutate(resistencia=str_extract(archivos,
                                    pattern = "[0-9]+.csv"))
  %>%
    mutate(resistencia=factor(str_remove(resistencia,
                                         pattern = ".csv")),
           watts = V*A,

```

```

    wind_power = 0.5*1.2*0.27*0.45*(`m/s`)^3,
    TSR = RPM*2*pi*r/60/`m/s`,
    cp = watts/wind_power)

df%<>% mutate(V_viento_correcion= ifelse(
  df$experimento != "piloto",
                                (df$`m/s`)/0.7, df$`m/s`))
df%<>% mutate(wind_power_correcion =
  0.5*1.2*0.27*0.45*(df$V_viento_correcion)^3,
  TSR_correc = RPM*2*pi*r/60/df$V_viento_correcion,
  cp_correc = watts/wind_power_correcion)

df%<>% mutate(V_viento_alfa= ifelse(df$experimento != "piloto",
                                ((df$`m/s`)/0.7)/1.85,( df$`m/s`)/1.85))
df%<>%
  mutate(wind_power_alfa = 0.5*1.2*0.50*0.45*(df$V_viento_alfa)^3,
  TSR_alfa = RPM*2*pi*r/60/df$V_viento_alfa,
  watts_alfa=watts/1.85,
  cp_alfa = watts_alfa/wind_power_alfa)

return(df)
}

```

Función de tratamiento de valores atípicos de la velocidad de giro y función para añadir coeficientes de la regresión a la tabla principal.

```

ajuste_RPM_Resistencia_so<- function(df,tabla_sinout){
  group_number<-length(attr(group_by(df,experimento,
                                angulo,porcentaje),
                                "group"))

  r<- (-2)

  lista_rpm_resistencia<- list()
  nombres_lista<- vector()
  titulos_grafico<- vector()
  tabladenombres<- matrix(-31,ncol = 3, nrow = group_number)
  for (grupos in 1:group_number) {
    grupos_rpm_resistencia<- df %>% group_by(.,
                                experimento,
                                angulo,
                                porcentaje)

    %>% select_groups(grupos)

    tabla_resistencia_rpm<- as.data.frame(
      cbind(as.numeric(
        grupos_rpm_resistencia$RPM),
        as.data.frame(
          grupos_rpm_resistencia$resistencia)))
  }
}

```

```

colnames(tabla_resistencia_rpm)<- c("RPM", "Omhnios")

nombre_tabla<- unique(paste(grupos_rpm_resistencia$experimento,
                             grupos_rpm_resistencia$angulo,
                             grupos_rpm_resistencia$porcentaje,sep = "_"))

if(is.na(grupos_rpm_resistencia$angulo)){
  titulo_graph<- unique(paste0(grupos_rpm_resistencia$experimento,"-",
                                grupos_rpm_resistencia$porcentaje,"%"))
}else{
  titulo_graph<- unique(paste0(grupos_rpm_resistencia$experimento,"-",
                                grupos_rpm_resistencia$angulo,"°-",
                                grupos_rpm_resistencia$porcentaje,"%"))
}
titulos_grafico[grupos]<- titulo_graph
lista_rpm_resistencia[[grupos]]<- tabla_resistencia_rpm
nombres_lista[grupos]<- nombre_tabla
tabladenombres[grupos,1]<-as.character(grupos_rpm_resistencia$experimento[1])
tabladenombres[grupos,2]<- as.character(grupos_rpm_resistencia$angulo[1])
tabladenombres[grupos,3]<- as.character(grupos_rpm_resistencia$porcentaje[1])
}

ordenando<- function(ordenando){
  return(ordenando[order(as.numeric(as.character(ordenando$Omhnios))),] )
}
lista_rpm_resistencia_ordenada<-lapply(lista_rpm_resistencia, ordenando)

lista_coef<-list()
for (pruebas in 1:length(lista_rpm_resistencia_ordenada)) {
  dir.create(paste0(here(),"/graficos_RPM_Resistencia_sinout/"))

  tiff(paste0(here(),"/graficos_RPM_Resistencia/",
              nombres_lista[pruebas],".tiff"),
       width = 7, height =7,
       units = 'in', res = 300)

  x<-as.numeric(as.character(lista_rpm_resistencia_ordenada[[pruebas]][,2]))
  x_so<-tabla_sinout[[pruebas]][,2]
  y<- as.numeric(as.character(lista_rpm_resistencia_ordenada[[pruebas]][,1]))
  y_so<-tabla_sinout[[pruebas]][,1]

  m<-nls(y~a*x/(b+x))
  m_so_1<-function(y_so,x_so){
    return(nls(y_so~a*x_so/(b+x_so)))
  }
  m_so_2<-function(y_so,x_so){
    return(nlsLM(y_so~a*x_so/(b+x_so)))
  }
  m_so<-tryCatch(m_so_1(y_so,x_so),
                 error=function(e) m_so_2(y_so,x_so))

  coefa_so<-coef(m_so)[1]

```

```

coefb_so<- coef(m_so)[2]
x_so_seq<- seq(0,8000,by=1)
y_so_seq<- coefa_so*x_so_seq/(coefb_so+x_so_seq)

coefa <-coef(m)[1]
coefb <- coef(m)[2]
x_seq<- seq(0,8000,by=1)
y_seq<- coefa*x_seq/(coefb+x_seq)

plot(NULL,xlim=c(0,8000),
      ylim = c(0,(max(y_so_seq)+(max(y_so_seq))/5)),
      cex=0.005, yaxt ="n",
      xlab = expression(paste("Resistencia (",Omega,")")),
      ylab = "Velocidad Angular (RPM)", bty='L')
par(new=T)
lines(x_seq,y_seq,lty=2, col="red",lwd=1)
par(new=T)
lines(x_so_seq,y_so_seq,lty=2,col="blue",lwd=1)

points(x_so,y_so, pch= 4, col="blue")

x_dif<-length(setdiff(x, x_so))
y_dif<-length(setdiff(round(y,r),round(y_so,r)))

while(x_dif != y_dif){
  y_dif<-length(setdiff(round(y,r),round(y_so,r)))
  r<-r+1}

if(x_dif==y_dif){
  points(setdiff(x, x_so),setdiff(round(y,r),
                                  round(y_so,r)),
        pch=20, col="red",
        cex=1.5)
}

subtitle_nom<- paste0("Experimento = ",titulos_grafico[pruebas])
maintitle<- paste0("Gráfica Velocidad Angular-Resistencia \n",
                   subtitle_nom)

par(new=T)
axis(2, at=seq(0,
               (max(y_so_seq)+(max(y_so_seq))/5),
               by=round((max(y_so_seq)+(max(y_so_seq))/5)/10,-1)),
     las=2)
title(main = maintitle)

R_sinVA<- paste0("Regresión sin VA",
                 " (Correlación= ",
                 as.character(round(cor(y_so,
                                       predict(m_so)),3)),")")

R_conVA<- paste0("Regresión con VA",
                 " (Correlación= ",

```

```

as.character(round(cor(y,predict(m)),3)),")")

legend("right", inset=c(0,0),
      legend = c(R_sinVA,R_conVA,
                  "Valores comunes",
                  "Valores atípicos "),
      pch = c(NA,NA, 4,20),
      lty = c(2,2,NA,NA),
      col = c("blue","red","black","red"),ncol = 1,cex = 1)

dev.off()

correlacion_sino<-cor(y_so,predict(m_so))
correlacion_cono<-cor(y,predict(m))
if(correlacion_sino>= correlacion_cono){
  aa<- coefa_so
  bb<- coefb_so
}else{
  aa<- coefa
  bb<-coefb
}

coef_tabla<- cbind(as.character(tabladenombres[pruebas,1]),
                  as.character(tabladenombres[pruebas,2]),
                  as.character(tabladenombres[pruebas,3]),
                  aa,bb)
names(coef_tabla)<-c("Experimento","Angulo","Porcentaje","a","b")

lista_coef[[pruebas]]<-coef_tabla
}

return(lista_coef)
}

add_coef<-function(df,coeficientes_RPM){
  df %<>% mutate(coef_a= -31,coef_b=-31)
  vector_logico<- list()
  for (i in 1:length(coeficientes_RPM[,1])) {
    if(coeficientes_RPM[i,1]=="concentrador"){
      vector_logico[[i]]<- c(coeficientes_RPM[i,1]==df$experimento &
                             coeficientes_RPM[i,2]==df$angulo &
                             coeficientes_RPM[i,3]==df$porcentaje)

    }else{
      vector_logico[[i]]<- c(coeficientes_RPM[i,1]==df$experimento &
                             coeficientes_RPM[i,3]==df$porcentaje)

    }
  }
  for (j in 1:length(vector_logico)) {

```

```

df %<>%
  mutate(coef_a=replace(coef_a,
                        vector_logico[[j]],
                        as.numeric(as.character(coeficientes_RPM[j,4]))),
         coef_b=replace(coef_b,
                        vector_logico[[j]],
                        as.numeric(as.character(coeficientes_RPM[j,5]))))
}

df %<>% mutate(RPM_regresion= (as.numeric(as.character(coef_a)) *
                             as.numeric(as.character(resistencia)))/
                  (as.numeric(as.character(coef_b))+
                   as.numeric(as.character(resistencia))))
df %<>% mutate(diff_rpm=RPM-RPM_regresion,
               TSR_regresion_est = RPM_regresion*2*pi*r/60/Vviento_estandar,
               TSR_regresion_lectura = RPM_regresion*2*pi*r/60/`m/s`)
df %<>% mutate(TSR_regresion_corr = RPM_regresion*2*pi*r/60/V_viento_correccion)

}

```

### Función de creación de la curva de potencia del modelo y prototipo

```

grafica_Potencia_V<-function(df,xlimite,ylimite,V){
  select_groups <- function(data, groups) {
    data[sort(unlist(attr(data, "indices")[ groups ])) + 1, ]
  }

  group_number<-length(attr(group_by(df,experimento,angulo), "group"))
  lista_watts_Vviento<- list()
  nombres_expr<-vector()
  for (groups_ind in 1:group_number) {

    xx<- df %>% group_by(.,experimento,angulo) %>% select_groups(groups_ind)
    nombre_1<-as.character(xx$experimento[1])
    nombre_2<-as.character(xx$angulo[1])
    nombre<-paste(nombre_1,nombre_2,sep = "_")

    porcentaje_number<-length(attr(group_by(xx,porcentaje), "group") )
    xx_porcentaje<-list()
    for (per in 1:porcentaje_number) {
      xx_perc<- xx %>% group_by(.,porcentaje) %>% select_groups(per)
      V_select<- as.name(paste0("xx_perc$",V))

      if(V==1){
        xx_perc<-cbind(xx_perc$cp_est,
                      xx_perc$watts,
                      xx_perc$V_viento_correccion)
      }else{
        xx_perc<-cbind(xx_perc$cp_est,
                      xx_perc$watts,
                      xx_perc$Vviento_estandar)
      }
    }
  }
}

```



```

    }
    colnames(xx_perc)<- c("cp","watts", "Vviento")
    xx_porcentaje[[per]]<- xx_perc

  }
  lista_watts_Vviento[[groups_ind]]<- xx_porcentaje
  nombres_expr[groups_ind]<- nombre
}

names(lista_watts_Vviento)<- nombres_expr

lista_watts_Vviento_max<-list()
for (i in 1:length(lista_watts_Vviento)) {

  df_watts_Vviento<- data.frame(matrix(unlist(lapply(lista_watts_Vviento[[i]],
                                                    function(x)
                                                    x[which.max(x[,1]),2:3])),
                                                    nrow=6, byrow=T))

  names(df_watts_Vviento)<- c("Watts","Vviento")
  lista_watts_Vviento_max[[i]]<- df_watts_Vviento
}
names(lista_watts_Vviento_max)<- nombres_expr

dir.create(paste0(here(),"/graficos_Potencia_V/"))

tiff(paste0(here(),"/graficos_Potencia_V/grafica_Potencia_V.tiff"),
     width = 7, height =7,
     units = 'in', res = 300)

colores<- c("orange","red","blue","dodgerblue4","purple","black")
pch_dif<-c(0:5)
correlacion<- vector()
lista_coef<-list()
for(i in 1:length(lista_watts_Vviento_max)){

  x<- lista_watts_Vviento_max[[i]][,2]
  y<- lista_watts_Vviento_max[[i]][,1]
  fit_curva<-nls(y~b+a*x^3,start = list(a=0, b=0))
  xx <- seq(min(x),xlimite[2], by=0.1)

  plot(NULL,xlim=xlimite,
       ylim = ylimite,cex=0.005, yaxt ="n",
       xlab = "Velocidad del viento (m/s)",
       ylab = "Potencia (W)", bty='L')
  title(main= "Curvas de potencia")
  par(new=T)
  lines(xx, predict(fit_curva,
                   data.frame(x=xx)),
        col=colores[i],lwd=1,lty=2)
  points(x,y, pch= pch_dif[i])
  par(new=T)
  correlacion[i]<- cor(y,predict(fit_curva))
}

```

```

    lista_coef[[i]]<- coef(fit_curva)
  }

axis(2, at=seq(0,ylimite[2], by=round(ylimite[2]/7,0)),las=2)

titulos_graficos<-function(df){
  group_number<-length(attr(group_by(df,
                                experimento,angulo,
                                porcentaje), "group"))

  lista_rpm_resistencia<- list()
  nombres_lista<- vector()
  titulos_grafico<- vector()
  tabladenombres<- matrix(-31,ncol = 3, nrow = group_number)
  for (grupos in 1:group_number) {
    grupos_rpm_resistencia<- df %>% group_by(.,
                                experimento,
                                angulo,
                                porcentaje) %>%

    select_groups(grupos)

    tabla_resistencia_rpm<- as.data.frame(cbind(as.numeric(grupos_rpm_resistencia$RPM),
                                                  as.data.frame(grupos_rpm_resistencia$resistencia)))
    colnames(tabla_resistencia_rpm)<- c("RPM", "Omhnios")

    nombre_tabla<- unique(paste(grupos_rpm_resistencia$experimento,
                                grupos_rpm_resistencia$angulo,
                                grupos_rpm_resistencia$porcentaje,sep = "_"))

    if(is.na(grupos_rpm_resistencia$angulo)){
      titulo_graph<- unique(paste0(grupos_rpm_resistencia$experimento))
    }else{
      titulo_graph<- unique(paste0(grupos_rpm_resistencia$experimento,"-",
                                    grupos_rpm_resistencia$angulo,"°"))
    }
    titulos_grafico[grupos]<- titulo_graph
    lista_rpm_resistencia[[grupos]]<- tabla_resistencia_rpm
    nombres_lista[grupos]<- nombre_tabla
    tabladenombres[grupos,1]<-as.character(grupos_rpm_resistencia$experimento[1])
    tabladenombres[grupos,2]<- as.character(grupos_rpm_resistencia$angulo[1])
    tabladenombres[grupos,3]<- as.character(grupos_rpm_resistencia$porcentaje[1])
  }
  return(unique(titulos_grafico))
}

nombres_df<- titulos_graficos(df)

leyenda<- paste0(nombres_df," (Cor= ",round(correlacion,4),")")

legend("topleft",y.intersp = 0.75,seg.len = 0.9,
      bty="n", bg="transparent",inset=c(0,0),
      legend = leyenda,lty = c(2,2,2,2,2),
      lwd = c(2,2,2,2,2),col = colores[1:5],

```

```

        ncol = 1,cex = 1)
dev.off()

tabladenombress<-function(df){
  group_number<-length(attr(group_by(df,experimento,angulo), "group"))
  tabladenombres<- matrix(-31,ncol = 2, nrow = group_number)
  for (grupos in 1:group_number) {
    grupos_rpm_resistencia<- df %>% group_by(.,
                                              experimento,angulo)

    %>% select_groups(grupos)
    tabladenombres[grupos,1]<-as.character(grupos_rpm_resistencia$experimento[1])
    tabladenombres[grupos,2]<- as.character(grupos_rpm_resistencia$angulo[1])
  }
  return(as.data.frame(tabladenombres))
}

tablanombres<-tabladenombress(df)

coeficientes_P_V<-data.frame(matrix(unlist(lista_coef),
                                     nrow=5, byrow=T))

coef_tabla<- cbind(tablanombres,coeficientes_P_V)
names(coef_tabla)<-c("Experimento","Angulo","a","b")

return(coef_tabla)
}

grafica_Potencia_V_alfa<-function(df,xlimite,ylmite){
  select_groups <- function(data, groups) {
    data[sort(unlist(attr(data, "indices")[ groups ])) + 1, ]
  }

  group_number<-length(attr(group_by(df,experimento,angulo), "group"))
  lista_watts_Vviento_modelo<- list()
  lista_watts_Vviento_prototipo<- list()

  nombres_expr<-vector()
  for (groups_ind in 1:group_number) {

    xx<- df %>% group_by(.,experimento,angulo) %>%
      select_groups(groups_ind)
    nombre_1<-as.character(xx$experimento[1])
    nombre_2<-as.character(xx$angulo[1])
    nombre<-paste(nombre_1,nombre_2,sep = "_")

    porcentaje_number<-length(attr(group_by(xx,porcentaje), "group") )
    xx_porcentaje_modelo<-list()
    xx_porcentaje_prototipo<-list()

    for (per in 1:porcentaje_number) {
      xx_perc<- xx %>% group_by(.,porcentaje) %>% select_groups(per)
    }
  }
}

```

```

xx_perc_modelo<-cbind(xx_perc$cp_est,
                      xx_perc$watts,
                      xx_perc$V_viento_correcion)
colnames(xx_perc_modelo)<- c("cp","watts", "Vviento")
xx_porcentaje_modelo[[per]]<- xx_perc_modelo

xx_perc_prototipo<-cbind(xx_perc$cp_alfa,
                        xx_perc$watts_alfa,
                        xx_perc$V_viento_alfa)
colnames(xx_perc_prototipo)<- c("cp","watts", "Vviento")
xx_porcentaje_prototipo[[per]]<- xx_perc_prototipo

}
lista_watts_Vviento_modelo[[groups_ind]]<- xx_porcentaje_modelo
lista_watts_Vviento_prototipo[[groups_ind]]<- xx_porcentaje_prototipo

nombres_expr[groups_ind]<- nombre
}

names(lista_watts_Vviento_modelo)<- nombres_expr

lista_watts_Vviento_max_modelo<-list()
lista_watts_Vviento_max_prototipo<-list()

for (i in 1:length(lista_watts_Vviento_modelo)) {

  df_watts_Vviento_modelo<- data.frame(matrix(unlist(lapply(lista_watts_Vviento_modelo[[i]],
                                                            function(x)
                                                                x[which.max(x[,1]),2:3])),
                                                                nrow=6, byrow=T))
  names(df_watts_Vviento_modelo)<- c("Watss","Vviento")
  lista_watts_Vviento_max_modelo[[i]]<- df_watts_Vviento_modelo

  df_watts_Vviento_prototipo<- data.frame(matrix(unlist(lapply(lista_watts_Vviento_prototipo[[i]],
                                                            function(x)
                                                                x[which.max(x[,1]),2:3])),
                                                                nrow=6, byrow=T))
  names(df_watts_Vviento_prototipo)<- c("Watss","Vviento")
  lista_watts_Vviento_max_prototipo[[i]]<- df_watts_Vviento_prototipo

}

names(lista_watts_Vviento_max_modelo)<- nombres_expr

dir.create(paste0(here()),"/graficos_Potencia_V/"))

tiff(paste0(here()),"/graficos_Potencia_V/grafica_Potencia_V_comparativa.tiff"),
    width = 7, height =7,
    units = 'in', res = 300)

```

```

colores<- c("orange","red","blue","dodgerblue4","purple","black")
pch_dif<-c(0:5)
correlacion<- vector()
lista_coef_modelo<-list()
lista_coef_prototipo<-list()
for(i in 1:length(lista_watts_Vviento_max_modelo)){
  x_modelo<- lista_watts_Vviento_max_modelo[[i]][,2]
  y_modelo<- lista_watts_Vviento_max_modelo[[i]][,1]
  fit_curva_modelo<-nls(y_modelo~b+a*x_modelo^3,
                        start = list(a=0, b=0))
  xx_modelo <- seq(min(x_modelo),xlimite[2], by=0.1)

  x_prototipo<- lista_watts_Vviento_max_prototipo[[i]][,2]
  y_prototipo<- lista_watts_Vviento_max_prototipo[[i]][,1]
  fit_curva_prototipo<-nls(y_prototipo~b+a*x_prototipo^3,
                           start = list(a=0, b=0))
  xx_prototipo <- seq(min(x_prototipo),xlimite[2], by=0.1)

  plot(NULL,xlim=xlimite,
        ylim = ylimite,cex=0.005, yaxt ="n",
        xlab = "Velocidad del viento (m/s)",
        ylab = "Potencia (W)", bty='L')
  title(main= "Curvas de potencia")
  par(new=T)

  lines(xx_modelo,
        predict(fit_curva_modelo,
                data.frame(x_modelo=xx_modelo)),
        col=colores[i],
        lwd=1,lty=2)

  lines(xx_prototipo, predict(fit_curva_prototipo, data.frame(x_prototipo=xx_prototipo)),
        col=colores[i],lwd=1,lty=1)

  par(new=T)
  lista_coef_modelo[[i]]<- coef(fit_curva_modelo)
  lista_coef_prototipo[[i]]<- coef(fit_curva_prototipo)
}

axis(2, at=seq(0,ylimite[2],
               by=round(ylimite[2]/7,0)),las=2)

titulos_graficos<-function(df){
  group_number<-length(attr(group_by(df,experimento,
                                     angulo,
                                     porcentaje),
                          "group"))
  lista_rpm_resistencia<- list()
  nombres_lista<- vector()
  titulos_grafico<- vector()
  tabladenombres<- matrix(-31,ncol = 3,

```

```

        nrow = group_number)
for (grupos in 1:group_number) {
  grupos_rpm_resistencia<- df %>% group_by(.,
                                          experimento,
                                          angulo,
                                          porcentaje) %>% select_groups(grupos)

  tabla_resistencia_rpm<- as.data.frame(cbind(as.numeric(
    grupos_rpm_resistencia$RPM),
    as.data.frame(
      grupos_rpm_resistencia$resistencia)))
  colnames(tabla_resistencia_rpm)<- c("RPM", "Omhnios")

  nombre_tabla<- unique(paste(grupos_rpm_resistencia$experimento,
                              grupos_rpm_resistencia$angulo,
                              grupos_rpm_resistencia$porcentaje,sep = "_"))

  if(is.na(grupos_rpm_resistencia$angulo)){
    titulo_graph<- unique(paste0(grupos_rpm_resistencia$experimento))
  }else{
    titulo_graph<- unique(paste0(grupos_rpm_resistencia$experimento,"-",
                                  grupos_rpm_resistencia$angulo,"°"))
  }
  titulos_grafico[grupos]<- titulo_graph
  lista_rpm_resistencia[[grupos]]<- tabla_resistencia_rpm
  nombres_lista[grupos]<- nombre_tabla
  tabladenombres[grupos,1]<-as.character(grupos_rpm_resistencia$experimento[1])
  tabladenombres[grupos,2]<- as.character(grupos_rpm_resistencia$angulo[1])
  tabladenombres[grupos,3]<- as.character(grupos_rpm_resistencia$porcentaje[1])
}
return(unique(titulos_grafico))
}
nombres_df<- titulos_graficos(df)

leyenda<- paste0(nombres_df)

legend("topleft",y.intersp = 0.75,seg.len = 1.2,
      bty="n", bg="transparent",inset=c(0,0),
      legend = leyenda,lty = c(1,1,1,1,1),
      lwd = c(2,2,2,2,2),col = colores[1:5],
      ncol = 1,cex = 1)
dev.off()

tabladenombress<-function(df){
  group_number<-length(attr(group_by(df,
                                     experimento,angulo),
                           "group"))
  tabladenombres<- matrix(-31,ncol = 2, nrow = group_number)
  for (grupos in 1:group_number) {
    grupos_rpm_resistencia<- df %>% group_by(.,
                                              experimento,

```

```

                                angulo) %>% select_groups(grupos)
    tabladenombres[grupos,1]<-as.character(grupos_rpm_resistencia$experimento[1])
    tabladenombres[grupos,2]<- as.character(grupos_rpm_resistencia$angulo[1])
  }
  return(as.data.frame(tabladenombres))
}

tablanombres<-tabladenombress(df)

coeficientes_P_V_modelo<-data.frame(matrix(unlist(lista_coef_modelo),
                                             nrow=5, byrow=T))
coeficientes_P_V_prototipo<-data.frame(matrix(unlist(lista_coef_prototipo),
                                              nrow=5, byrow=T))

coef_tabla<- cbind(tablanombres,
                   coeficientes_P_V_prototipo)
names(coef_tabla)<-c("Experimento", "Angulo", "a", "b")

return(coef_tabla)
}

```

### Función de creación de la curva Cp-lambda, individual y comparativa

```

ploteo_experimento_estandar_RPM_regresion_CPmax<- function(datos,grados){
  df<-datos
  select_groups <- function(data, groups) {
    data[sort(unlist(attr(data, "indices")[ groups ])) + 1, ]
  }

  lista_Cpmax_total<-list()
  group_number<-length(attr(group_by(df,experimento,angulo), "group"))
  for (groups_ind in 1:group_number) {

    xx<- df %>% group_by(.,experimento,angulo) %>%
      select_groups(groups_ind)
    nombre_1<-as.character(xx$experimento[1])
    nombre_2<-as.character(xx$angulo[1])
    nombre<-paste(nombre_1,nombre_2,sep = "_")
    if(is.na(nombre_2)){
      nombre_grafica<- nombre_1
    }else{
      nombre_grafica<-paste(nombre_1," ",nombre_2,"°",sep = "")
    }

    porcentaje_number<-length(attr(group_by(xx,porcentaje),
                                       "group") )

    xx_porcentaje<-list()
    for (per in 1:porcentaje_number) {
      xx_perc<- xx %>% group_by(.,porcentaje) %>%
        select_groups(per)
      xx_perc<-cbind(xx_perc$cp_correc,

```

```

        xx_perc$TSR_regresion_corr,
        xx_perc$V_viento_correcion)
colnames(xx_perc)<- c("cp", "TSR", "Vviento")
xx_porcentaje[[per]]<- xx_perc
}

##lambda_cp es una tabla de dos columnas (cp,lambda)
lambda_Cp<- xx_porcentaje
lambda_Cp_clean<-list()
for(j in 1:length(lambda_Cp)){
  cp_lmb<- lambda_Cp[[j]]
  cp_lmb<-cp_lmb[order(cp_lmb[,2]),]

  TSR_1<-cp_lmb[,2]
  Cp_1<-cp_lmb[,1]

  V_tsr<- seq(0.05,max(TSR_1),by=max(TSR_1)/8)
  TSR_2<- vector()
  Cp_2<- vector()

  for(i in 1:length(V_tsr)){

    Cp_2[i]<- Cp_1[which.min(abs(TSR_1-V_tsr[i]))]
    TSR_2[i]<-TSR_1[which.min(abs(TSR_1-V_tsr[i]))]

  }
  validacion<-cbind(unique(Cp_2),unique(TSR_2))
  validacion_1<- validacion[1:which.max(validacion[,1]),]
  validacion_2<- validacion[(which.max(validacion[,1])+1):
                           length(validacion[,1]),]

  indeeex<- vector()
  rr<- 1

  if(length(validacion_2)==2){
    validacion_2<- validacion_2
  }else{
    for (i in 1:length(validacion_2[,1])) {

      if(i==length(validacion_2[,1])){break}else{

        if(validacion_2[i,1] < validacion_2[(i+1),1]){
          indeeex[rr]<- as.numeric(i)
          rr<-rr+1
        }

      }
    }
  }
}

```



```

    }
  }

}

if(length(indeex)==0){
  validacion_2<- validacion_2
}else{
  validacion_2<-validacion_2[-indeex,]
}

clean_table<-rbind(validacion_1,validacion_2)

lambda_Cp_clean[[j]]<- clean_table
}

dir.create(paste0(here(),"/graficos_Cpmax_RPMreg_estandar_fit",grados,"/"))

tiff(paste0(here(),"/graficos_Cpmax_RPMreg_estandar_fit",
            grados,"/",nombre,".tiff"),
     width = 7,
     height =7,
     units = 'in',
     res = 300)

lambda_Cp<- lambda_Cp_clean
colores<- c("orange","red","blue","dodgerblue4","purple","black")
pch_dif<-c(0:5)
lista_Cpmax<-list()
for(i in 1:length(lambda_Cp)){
  x<- lambda_Cp[[i]][,2]
  y<- lambda_Cp[[i]][,1]
  fit5<-lm(y~poly(x,grados,raw=TRUE))
  xx <- seq(min(x),max(x), by=0.01)

  plot(NULL,xlim=c(0,2),
       ylim = c(0,0.20),cex=0.005, yaxt ="n",
       xlab = "TSR", ylab = "Cp", bty='L')
  par(new=T)
  lines(xx, predict(fit5, data.frame(x=xx)), col=colores[i],lwd=2)
  tabla_maxcp<-cbind(xx,predict(fit5, data.frame(x=xx)))
  Cp_max_point<-tabla_maxcp[which.max(tabla_maxcp[,2]),]
  lista_Cpmax[[i]]<- Cp_max_point
  par(new=T)

```

```

}

axis(2, at=seq(0,0.2, by=0.02),las=2)
V_viento<-sapply(xx_porcentaje,"[,1,3)

leyenda_veintos<-paste0(round(V_viento,digits = 2), " m/s")
orden_leyenda<-cbind(V_viento,leyenda_veintos,pch_dif,colores)
orden_leyenda<-orden_leyenda[order(as.numeric(orden_leyenda[,1]),
                                   decreasing = TRUE),]

legend("topright", inset=c(0,0),orden_leyenda[,2],
      pch = as.numeric(orden_leyenda[,3]),
      text.col = orden_leyenda[,4],ncol = 1,cex = 1)

tabla_CPmax<-data.frame(matrix(unlist(lista_CPmax),
                               nrow=length(lista_CPmax),
                               byrow=T),stringsAsFactors=FALSE)
tabla_CPmax<-rbind(tabla_CPmax)
names(tabla_CPmax)<- c("TSR","Cp")
tabla_CPmax<- tabla_CPmax[order(tabla_CPmax[,1]),]

points(tabla_CPmax$TSR,tabla_CPmax$Cp, pch= 20,cex=2)
y<- tabla_CPmax$Cp
x<- tabla_CPmax$TSR

fit_cp<-lm(y~poly(x,grados,raw=TRUE))

m_so_1<-function(y_so,x_so){
  return(nls(y_so~a*x_so/(b+x_so),
            start = list(a=-100, b=-100)))
}
m_so_2<-function(y_so,x_so){
  return(nlsLM(y_so~a*x_so/(b+x_so),
            start = list(a=-100, b=-100)))
}
m_so<-tryCatch(m_so_1(y,x), error=function(e) m_so_2(y,x))
a_m<- as.numeric(as.character(coef(m_so)[1]))
b_m<-as.numeric(as.character(coef(m_so)[2]))
y_m_so<- (a_m*xx)/(b_m+xx)

subtitulo_nom<- paste0("Experimento = ",nombre_grafica)
maintitle<- paste0("Gráfica CP-TSR \n",subtitulo_nom)
title(main = maintitle)

dev.off()

lista_CPmax_total[[groups_ind]]<- tabla_CPmax

```

```

}

return(lista_Cpmax_total)
}

ploteo_CPmax10<- function(datos,grados){
  df<-datos
  select_groups <- function(data, groups) {
    data[sort(unlist(attr(data, "indices")[ groups ])) + 1, ]
  }

  lista_Cpmax_total<-list()
  lista_Cps_nueva<-list()
  nombres_gra<- vector()
  group_number<-length(attr(group_by(df,
                                   experimento,
                                   angulo), "group"))
  for (groups_ind in 1:group_number) {

    xx<- df %>% group_by(.,experimento,angulo)
    %>% select_groups(groups_ind)
    porcentaje_number<-length(attr(group_by(xx,porcentaje),
                                       "group") )

    nombre_1<-as.character(xx$experimento[1])
    nombre_2<-as.character(xx$angulo[1])
    nombre<-paste(nombre_1,nombre_2,sep = "_")

    if(is.na(nombre_2)){
      nombre_grafica<- nombre_1
    }else{
      nombre_grafica<-paste(nombre_1," ",
                           nombre_2,"°",
                           sep = "")
    }

    xx_porcentaje<-list()
    for (per in 1:porcentaje_number) {
      xx_perc<- xx %>% group_by(.,porcentaje)
      %>% select_groups(per)
      xx_perc<-cbind(xx_perc$cp_alfa,
                    xx_perc$TSR_alfa,
                    xx_perc$V_viento_alfa)
      colnames(xx_perc)<- c("cp","TSR", "Vviento")
      xx_porcentaje[[per]]<- xx_perc
    }

    ##lambda_cp es una tabla de dos columnas (cp,lambda)
    lambda_Cp<- xx_porcentaje
  }
}

```

```

lambda_Cp_clean<-list()
for(j in 1:length(lambda_Cp)){
  cp_lmb<- lambda_Cp[[j]]
  cp_lmb<-cp_lmb[order(cp_lmb[,2]),]

  TSR_1<-cp_lmb[,2]
  Cp_1<-cp_lmb[,1]

  V_tsr<- seq(0.05,max(TSR_1),by=max(TSR_1)/8)
  TSR_2<- vector()
  Cp_2<- vector()

  for(i in 1:length(V_tsr)){

    Cp_2[i]<- Cp_1[which.min(abs(TSR_1-V_tsr[i]))]
    TSR_2[i]<-TSR_1[which.min(abs(TSR_1-V_tsr[i]))]

  }
  validacion<-cbind(unique(Cp_2),unique(TSR_2))
  validacion_1<- validacion[1:which.max(validacion[,1]),]
  validacion_2<- validacion[(which.max(validacion[,1])+1):
                           length(validacion[,1]),]

  indeeex<- vector()
  rr<- 1

  if(length(validacion_2)==2){
    validacion_2<- validacion_2
  }else{
    for (i in 1:length(validacion_2[,1])) {

      if(i==length(validacion_2[,1])){break}else{

        if(validacion_2[i,1] < validacion_2[(i+1),1]){
          indeeex[rr]<- as.numeric(i)
          rr<-rr+1

        }

      }
    }
  }

  if(length(indeeex)==0){
    validacion_2<- validacion_2
  }else{
    validacion_2<-validacion_2[-indeeex,]
  }
}

```

```

clean_table<-rbind(validacion_1,validacion_2)

lambda_Cp_clean[[j]]<- clean_table
}

lista_Cps_nueva[[groups_ind]]<- lambda_Cp_clean
nombres_gra[groups_ind]<- nombre_grafica
}

vector_nombre_vel<- c(10.34,5.47,6.04,7.29,8.33,9.36)

for (porcentaje in 1:length(lista_Cps_nueva[[1]])) {

  lista_cpmsVmax<- lapply(lista_Cps_nueva,"[",porcentaje)

  x_max<-max(sapply(lista_cpmsVmax, function(x) max(x[,2])))
  y_max<-max(sapply(lista_cpmsVmax, function(x) max(x[,1])))

  dir.create(paste0(here()),"/graficos_Cp_juntos/")

  #jpeg(paste0(here()),"/graficos_Cp_juntos/",porcentaje,".jpeg")
  tiff(paste0(here()),"/graficos_Cp_juntos/",porcentaje,".tiff"),
      width = 7, height =7, units = 'in', res = 300)
  colores<- c("orange","red","blue","dodgerblue4","purple")
  pch_dif<-c(0:5)
  for(i in 1:(length(lista_cpmsVmax))){
    x<- lista_cpmsVmax[[i]][,2]
    y<- lista_cpmsVmax[[i]][,1]
    fit5<-lm(y~poly(x,grados,raw=TRUE))
    xx <- seq(min(x),max(x), by=0.01)

    plot(NULL,xlim=c(0,(x_max+0.2)),
         ylim = c(0,(y_max+0.01)),cex=0.005, yaxt ="n",
         xlab = "TSR", ylab = "Cp", bty='L')
    par(new=T)
    lines(xx, predict(fit5, data.frame(x=xx)), col=colores[i],lwd=2)
    tabla_maxcp<-cbind(xx,predict(fit5, data.frame(x=xx)))
    par(new=T)
  }

  axis(2, at=unique(round(seq(0,(y_max+0.02),
                           by=round((y_max/10),
                                   digits = 4))),
                           digits = 2)),las=2)

  leyenda<-paste0(nombres_gra)

```

```

    legend("topright", inset=c(0,0),leyenda,
           text.col = colores,ncol = 1,cex = 1)

    titulo<- paste0("Comparación de gráficas Cp-TSR \n
                    con el túnel de viento al 100 %")
    title(main = titulo)

    dev.off()

}

}

```

## Script de aplicación

El siguiente script se encarga de aplicar las funciones anteriormente definidas para obtener las gráficas y hacer los cálculos necesarios.

```

library(here)
source(here('funciones_necesarias.R'),encoding = "UTF-8")

data_path<- here('data/')
archivos <- list.files(data_path, pattern = 'csv',
                      recursive = TRUE, full.names = TRUE)

r<-0.27/2 #radio del cacharro

#Aplicamos función "procesador" y creamos la tabla principal "df"
lista_procesada<-lapply(archivos, procesador)
column_names <- names(lista_procesada[[1]])
df <- data.frame(matrix(unlist(lista_procesada),
                       nrow=length(lista_procesada),
                       byrow=T),stringsAsFactors=FALSE)
colnames(df) <- column_names
df<- cbind(archivos,df)

#Realizamos modificaciones y calculos en tabla "df".
#Añadiendo columnas de potencia, Cp y TSR

df<- df_mutate(df)

#Se ejecuta el script "tratamiento_outliers" para obtener una corrección de los valores atípicos
source(here('tratamiento_outliers.R'))

#Corrección de valores atípicos
coeficientes_RPM<-ajuste_RPM_Resistencia_so(df,
                                             tablas_sin_outliers_ni_decreasing)
coeficientes_RPM<-data.frame(matrix(unlist(coeficientes_RPM),

```

```

                                nrow=30, byrow=T))
names(coeficientes_RPM)<- c("Experimento","Angulo","Porcentaje","a","b")

#Se añaden coeficientes de la regresión a la tabla principal "df"
df<-add_coef(df,coeficientes_RPM)

#Ploteamos curvas Cp-lambda
ploteo_experimento_corr_RPM_regresion(df,3)

#Obtención de los Cp máximos
tabla_cpmax_tsr<-ploteo_experimento_estandar_RPM_regresion_CPmax(df,3)

#Ploteos de Cp-lambda comparativos
ploteo_CPmax10(df,2)

#Ploteos de curva de potencia añadiendo los limites de las gráficas
limitex<- c(0,11)
limitey<- c(0,20)
coeficientes_Curva_P_V<- grafica_Potencia_V(df,limitex,limitey,2)
grafica_Potencia_V_alfa(df,limitex,limitey)

```

## Script para el tratamiento de valores atípicos

```

library(outliers)
outliers_prueba<-ajuste_RPM_Resistencia(df)
RPM_outliers<-sapply(outliers_prueba, "[",,1)
Resistencia_outliers<-sapply(outliers_prueba, "[",,2)
Resistencia_outliers<-sapply(Resistencia_outliers,
                             function(x)
                               as.numeric(as.character(x)))
chisq_outlier<-t(sapply(RPM_outliers, chisq.out.test))
alternative_outlier<- chisq_outlier[,2]
alternative_outlier<-str_extract_all(alternative_outlier,"\\d+")
make_outlier<- function(x){

  return( paste(as.numeric(x[1]),as.numeric(x[2]), sep = "."))
}
alternative_outlier<- sapply(alternative_outlier,make_outlier)
Eliminar_outlier<- function(lista_rpms, vector_outliers){
  difff<- abs(length(lista_rpms)-as.numeric(vector_outliers))
  if(difff< 1){
    ret<-lista_rpms[-length(lista_rpms)]
  }else{ret<- lista_rpms}
  return(ret)
}
RPM_sin_outliers<- lapply(RPM_outliers, Eliminar_outlier,
                          vector_outliers=alternative_outlier)

tabla_sin_outliers<- function(resistencia,rpms){

```

```

if(length(resistencia)==length(rpms)){
  tabla<-cbind(rpms,resistencia)
}else{
  tabla<- cbind(rpms, resistencia[-length(resistencia)])
}
return(tabla)
}
tablas_sin_ot<-mapply(tabla_sin_outliers,
                      rpms=RPM_sin_outliers,
                      resistencia=Resistencia_outliers)

eliminar_decreasing<-function(validacion){
  indeeex<- vector()
  rr<- 1

  for (i in 1:length(validacion[,1])) {
    if(i==length(validacion[,1])){break}else{

      if(validacion[i+1,1] < validacion[(i),1]){
        indeeex[rr]<- as.numeric(i+1)
        rr<-rr+1

      }

    }
  }

  if(length(indeeex)==0){
    validacion<- validacion
  }else{
    validacion<-validacion[-indeeex,]
  }

  return(validacion)
}

tablas_sin_outliers_ni_decreasing<-lapply(tablas_sin_ot,
                                           eliminar_decreasing)

```

## Script para el tratamiento de los datos aportados por ERA-Interim

Este script es para analizar los datos aportados por el ERA-interim y además obtener la energía anual esperada de la instalación. Además, se incluyen las funciones generadoras de rosas de los vientos y diagramas de barras.

```

#archivo binario .nc.
library(RNetCDF)
library(dplyr)
library(here)
library(openair)
library(ggplot2)
#Creo un objeto string con el nombre completo del archivo

```



```

#PATH+nombre
inputpath<- here()

inputfile<-paste(inputpath,"/bilbo1979_2017.nc",sep="" )

#Acceder al contenido del archivo
# 2 pasos: 1.Abro el archivo; 2. Leo el archivo
SATALT_ini<-open.nc(inputfile)

#Vemos un resumen del contenido
print.nc(SATALT_ini)

#2.1 Leo el tiempo
time_1<-var.get.nc(SATALT_ini,"time")

#2.2 Latitud
latitude_1<-var.get.nc(SATALT_ini,"latitude")

#2.3 Longitud
longitude_1<-var.get.nc(SATALT_ini,"longitude")

#localización del edificio
lonref0<- 357.054; latref0<- 43.258

idlon<-which.min(abs(longitude_1 - lonref0))
longitude_1[idlon]

idlat<-which.min(abs(latitude_1-latref0))
latitude_1[idlat]

u10<-var.get.nc(SATALT_ini,"u10", unpack = T)
v10<-var.get.nc(SATALT_ini,"v10",unpack = T)

wind_abs = sqrt(u10^2 + v10^2)
wind_dir_trig_to = atan2(u10/wind_abs, v10/wind_abs)
wind_dir_trig_to_degrees = wind_dir_trig_to * 180/pi ## -111.6 degrees
ind_dir_trig_from_degrees = wind_dir_trig_to_degrees + 180 ## 68.38 degrees
summary(wind_abs)

tabla_loc<-expand.grid(longitude_1,latitude_1)
tabla_loc_time<- expand.grid(tabla_loc[,1],time_1)
tabla<- as.data.frame(cbind(tabla_loc_time[,1],tabla_loc[,2],
                           tabla_loc_time[,2],
                           wind_abs,ind_dir_trig_from_degrees))

names(tabla)<- c("longitud","latitud","time","ws","wd")

lon<- unique(tabla$longitud)
lat<-unique(tabla$latitud)

```

```

tabla_localizacion<-tabla[tabla$longitud==lon[2] & tabla$latitud==lat[2],]
tabla_localizacion<-tabla_localizacion %>%
  mutate(grup_vel=cut(
    tabla_localizacion$ws,
    seq(0,max(tabla_localizacion$ws),by=0.5),
    labels = seq(0.5,
      max(tabla_localizacion$ws),
      by=0.5),
    include.lowest = T,right = T))

tabla_localizacion_altura<- as.data.frame(tabla_localizacion %>%
  mutate(ws=ws*log(50/1) / log(10/1)))
tabla_localizacion_altura<-tabla_localizacion_altura %>%
  mutate(grup_vel=cut(tabla_localizacion_altura$ws,
    seq(0,max(tabla_localizacion_altura$ws),
      by=0.5),
    labels = seq(0.5,
      max(tabla_localizacion_altura$ws),
      by=0.5),
    include.lowest = T,right = T))

### representamos windrose a la altura del edificio
path_here<-paste0(here(),"/graficas_rosas/")
breaks_rose<-length(seq(0,max(tabla_localizacion_altura$ws),by=2))

tiff(paste0(path_here,"rosa_elegida_he.tiff"),
  width = 7, height =7,
  units = 'in', res = 300)

windRose(tabla_localizacion_altura, angle = 22.5,
  breaks = breaks_rose,paddle = F,
  annotate = F,
  key.position = "right")
dev.off()

## Realizamos barplot comparativo
distribuciones_velocidad<- table(tabla_localizacion_altura$grup_vel)
dist_total<-sum(distribuciones_velocidad)
distribuciones_velocidad_porcentaje<- distribuciones_velocidad/dist_total
distribuciones_velocidad_anual<- distribuciones_velocidad_porcentaje*8600

distribuciones_velocidad10<- table(tabla_localizacion$grup_vel)
dist_total10<-sum(distribuciones_velocidad10)
distribuciones_velocidad_porcentaje10<- distribuciones_velocidad10/dist_total10

```

```

distribuciones_velocidad_anual10<- distribuciones_velocidad_porcentaje10*8600

add.col<-function(df, new.col) {n.row<-dim(df)[1]
length(new.col)<-n.row
cbind(df, new.col)
}

frame_barplot<- as.data.frame(add.col(distribuciones_velocidad_anual,
                                     distribuciones_velocidad_anual10))
frame_barplot<- as.data.frame(cbind(frame_barplot$df,
                                     ifelse(is.na(frame_barplot$new.col),
                                             0,frame_barplot$new.col)))
names(frame_barplot)<- c("Dist_50","Dist_10")
row.names(frame_barplot)<- names(distribuciones_velocidad_anual)

ggplot(frame_barplot)+
  geom_bar(aes(x=as.numeric(row.names(frame_barplot)),
              y=Dist_50),stat = "identity",
          alpha=.95,fill='lightblue',
          color='lightblue4',
          show.legend = T)+
  geom_bar(aes(x=as.numeric(row.names(frame_barplot)),
              y=Dist_10),stat = "identity",
          alpha=.3,fill='pink',color='red',
          show.legend = T)+
  xlab("Velocidad del viento")+
  ylab("Horas anuales") +
  geom_point(x=18, y =500, shape=22, size=5, alpha=.95,
            fill='lightblue',color='lightblue4')+
  geom_point(x=18, y =400, shape=22,
            size=5, alpha=.3,fill='pink',
            color='red')+
  annotate("text",
         label="Distribución de la velocidad del viento a 50 metros",
         x = 30, y = 500)+
  annotate("text",label="Distribución de la velocidad del viento a 10 metros",
         x = 30, y = 400)+
  theme_bw()

path_here<-paste0(here(),"/barplot/")
dir.create(path_here)

ggplot(frame_barplot)+
  geom_bar(aes(x=as.numeric(row.names(frame_barplot)),
              y=Dist_50),stat = "identity",alpha=.95,
          fill='lightblue',
          color='lightblue4',
          show.legend = T)+
  geom_bar(aes(x=as.numeric(row.names(frame_barplot)),y=Dist_10),
          stat = "identity", alpha=.3,
          fill='pink',color='red',

```

```

        show.legend = T)+
xlab("Velocidad del viento (m/s)")+
ylab("Horas anuales") +
geom_point(x=18, y =500,
            shape=22, size=5,
            alpha=.95,fill='lightblue',
            color='lightblue4')+
geom_point(x=18, y =400, shape=22, size=5,
            alpha=.3,fill='pink',
            color='red')+
annotate("text",label="Distribución de la velocidad del viento a 50 metros",
         x = 30, y = 500)+
annotate("text",label="Distribución de la velocidad del viento a 10 metros",
         x = 30, y = 400)+
theme_bw()

ggsave(paste0(path_here,"barplotcomparativo.tiff"),
        device = "tiff", dpi=1200,width =8,
        height =7, units = 'in')

tabla_localizacion_NO<-tabla_localizacion_altura[tabla_localizacion_altura$wd
                                                < 350 &
                                                tabla_localizacion_altura$wd
                                                > 314,]

distribuciones_velocidad_NO<- table(tabla_localizacion_NO$grup_vel)
dist_total_NO<-sum(distribuciones_velocidad_NO)
distribuciones_velocidad_porcentaje_NO<- distribuciones_velocidad_NO/dist_total
distribuciones_velocidad_anual_NO<- distribuciones_velocidad_porcentaje_NO*8600
tabla_NO<-as.data.frame( cbind(distribuciones_velocidad_anual_NO))
names(tabla_NO)<- "distribucion"

ggplot(tabla_NO)+
  geom_bar(aes(x=as.numeric(row.names(tabla_NO)),y=distribucion),
           stat = "identity",alpha=.95,fill='lightblue',color='lightblue4')+
  xlab("Velocidad del viento (m/s)")+
  ylab("Horas anuales") +
  ggtitle("Distribución de la velocidad del viento \n dirección Noroeste")+
  theme_bw()+
  theme(plot.title = element_text(hjust = 0.5))

ggsave(paste0(path_here,"barplotN0.tiff"), device = "tiff",
        dpi=1200,width =8, height =7,
        units = 'in')

```

```

###Creamos un barplot comparativo entre todas las direcciones y la direccion noroeste

frame_barplot_NO<- as.data.frame(add.col(distribuciones_velocidad_anual,
                                         distribuciones_velocidad_anual_NO))
frame_barplot_NO<- as.data.frame(cbind(frame_barplot_NO$df,
                                       ifelse(is.na(frame_barplot_NO$new.col),
                                              0,frame_barplot_NO$new.col)))
names(frame_barplot_NO)<- c("Dist_total","Dist_NO")
row.names(frame_barplot_NO)<- names(distribuciones_velocidad_anual)


ggplot(frame_barplot_NO)+
  geom_bar(aes(x=as.numeric(row.names(frame_barplot_NO)),
              y=Dist_total),stat = "identity",
          alpha=.95,fill='lightblue',
          color='lightblue4',
          show.legend = T)+
  geom_bar(aes(x=as.numeric(row.names(frame_barplot_NO)),
              y=Dist_NO),stat = "identity",
          alpha=.4,fill='pink',
          color='red',show.legend = T)+
  xlab("Velocidad del viento (m/s)")+
  ylab("Horas anuales") +
  geom_point(x=22, y =300, shape=22,
            size=5, alpha=.95,
            fill='lightblue',
            color='lightblue4')+
  geom_point(x=22, y =200, shape=22,
            size=5, alpha=.3,
            fill='pink',color='red')+
  annotate("text",label="Distribución de la velocidad del \n
                    viento en todas las direcciones",
          x = 30, y = 300)+
  annotate("text",label="Distribución de la velocidad del \n
                    viento en la dirección Noroeste",
          x = 30, y = 200)+
  ggtitle("Comparativa de las distribuciones del viento \n
          en todas las direcciones y en la dirección aprovechable")+
  theme_bw()+
  theme(plot.title = element_text(hjust = 0.5))

ggsave(paste0(path_here,"barplotNO_comparativa.tiff"),
       device = "tiff", dpi=1200,width =8,
       height =7, units = 'in')

```

```

#### ya nos ponemos al calculo de la energia anual producida. Empleando los coeficien
## la formula de ajuste es  $y \sim b + a \cdot x^3$ 
V_viento<-seq(0,max(tabla_localizacion_altura$ws),
              by=0.5)
select_groups <- function(data, groups) {
  data[sort(unlist(attr(data, "indices")[ groups ])) + 1, ]
}
group_number<-length(attr(group_by(df,experimento,angulo), "group"))

coeficientes_Curva_P_V_medida<- grafica_Potencia_V(df,limitex,limitey,1)
coeficientes_Curva_P_V_estandar<- grafica_Potencia_V(df,limitex,limitey,2)
coeficientes_Curva_P_V_alfa<- grafica_Potencia_V_alfa(df,limitex,limitey)

lista_energias<- list()

for (i in 1:group_number) {
  coefs_medido<- coeficientes_Curva_P_V_alfa %>%
    group_by(.,Experimento,Angulo) %>%
    select_groups(i)
  coefs_estandar<- coeficientes_Curva_P_V_estandar %>%
    group_by(.,Experimento,Angulo) %>%
    select_groups(i)

  Curva_de_potencia_medida<- coefs_medido$b+coefs_medido$a*V_viento^3
  Curva_de_potencia_medida<-replace(Curva_de_potencia_medida,
                                   which(Curva_de_potencia_medida<0),0)
  Curva_de_potencia_medida<- Curva_de_potencia_medida[2:80]

  Curva_de_potencia_estandar<- coefs_estandar$b+coefs_estandar$a*V_viento^3
  Curva_de_potencia_estandar<-replace(Curva_de_potencia_estandar,
                                       which(Curva_de_potencia_estandar<0),0)
  Curva_de_potencia_estandar<- Curva_de_potencia_estandar[2:80]

  tabla_dist_pot<- as.data.frame(cbind(tabla_NO,
                                       Curva_de_potencia_estandar,
                                       Curva_de_potencia_medida))
  names(tabla_dist_pot)<- c("horas", "est", "med")
  energias<- tabla_dist_pot %>% mutate(energia_est= est*horas,
                                       energia_med=med*horas)

  lista_energias[[i]]<- energias
}

names(lista_energias)<- paste0(coeficientes_Curva_P_V_medida[,1],
                              coeficientes_Curva_P_V_medida[,2])

```

```

#barplot_energias_estandar<-function(){
  tabla_energia_est<-as.data.frame(matrix(unlist(lapply(lista_energias,
                                                    "[",,4), use.names = F),
                                          byrow = F,ncol = 5))
  names(tabla_energia_est)<- names(lista_energias)
  row.names(tabla_energia_est)<- names(distribuciones_velocidad_anual)

  path_here<-paste0(here(),"/barplot/")

  ggplot(tabla_energia_est)+
    geom_bar(aes(x=as.numeric(row.names(tabla_energia_est)),
                y=concentrador30),
            stat = "identity",alpha=.9,
            fill='lightblue',
            color='lightblue4',
            show.legend = T)+
    geom_bar(aes(x=as.numeric(row.names(tabla_energia_est)),
                y=concentrador45),
            stat = "identity",
            alpha=.8,fill='pink',
            color='red',
            show.legend = T)+
    geom_bar(aes(x=as.numeric(row.names(tabla_energia_est)),
                y=concentrador70),
            stat = "identity",alpha=.7,
            fill='green2',color='green4',
            show.legend = T)+
    geom_bar(aes(x=as.numeric(row.names(tabla_energia_est)),
                y=paredNA),
            stat = "identity", alpha=.6,fill='mediumorchid1',
            color='mediumorchid4',
            show.legend = T)+
    geom_bar(aes(x=as.numeric(row.names(tabla_energia_est)),
                y=pilotoNA),stat = "identity",
            alpha=.5,fill='orange',color='orange4',
            show.legend = T)+
    xlab("Velocidad del viento (m/s)")+
    ylab("Energía (W/h)") +
    geom_point(x=25, y =200,
              shape=22, size=5,
              alpha=.5,fill='lightblue',
              color='lightblue4')+
    geom_point(x=25, y =180, shape=22,
              size=5, alpha=.4,fill='pink',
              color='red')+
    geom_point(x=25, y =160, shape=22,
              size=5, alpha=.3,fill='green2',
              color='green4')+
    geom_point(x=25, y =140,

```

```

        shape=22, size=5, alpha=.2, fill='mediumorchid1',
        color='mediumorchid4')+
geom_point(x=25, y =120, shape=22, size=5,
        alpha=.1, fill='orange', color='orange4')+
annotate("text", label="Concentrador 30°", x = 30, y = 200)+
annotate("text", label="Concentrador 45°", x = 30, y = 180)+
annotate("text", label="Concentrador 70°", x = 30, y = 160)+
annotate("text", label="Pared", x = 27.5, y = 140)+
annotate("text", label="Piloto", x = 27.5, y = 120)+
ggtitle("Comparativa de las energías anuales producidas empleando las 5 configuraciones \n
        usando las curvas de potencia generadas con la velocidad estándar")+
theme_bw()+
theme(plot.title = element_text(hjust = 0.5))

ggsave(paste0(path_here, "barplotenergíacomparativa_estandar.tiff"),
        device = "tiff", dpi=1200, width =8,
        height =7, units = 'in')

return(as.data.frame(colSums(tabla_energia_est)))

}

barplot_energias_medida<-function(){
  tabla_energia_est<-
    as.data.frame(matrix(unlist(lapply(lista_energias,
                                     "[",,5),
                               use.names = F),
                        byrow = F, ncol = 5))
  names(tabla_energia_est)<- names(lista_energias)
  row.names(tabla_energia_est)<- names(distribuciones_velocidad_anual)

  path_here<-paste0(here(), "/barplot/")

  ggplot(tabla_energia_est)+
    geom_bar(aes(x=as.numeric(row.names(tabla_energia_est)),
                y=concentrador45),
            stat = "identity", alpha=.8,
            fill='pink', color='red', show.legend = T)+
    geom_bar(aes(x=as.numeric(row.names(tabla_energia_est)),
                y=concentrador30), stat = "identity", alpha=.9, fill='lightblue', color='lightblue4',
            show.legend = T)+
    geom_bar(aes(x=as.numeric(row.names(tabla_energia_est)),
                y=concentrador70),
            stat = "identity",
            alpha=.7, fill='green2',
            color='green4',
            show.legend = T)+
    geom_bar(aes(x=as.numeric(row.names(tabla_energia_est)),

```



```

        y=paredNA),stat = "identity", alpha=.6,fill='mediumorchid1',color='mediumorchid4',
        show.legend = T)+
geom_bar(aes(x=as.numeric(row.names(tabla_energia_est)),
        y=pilotoNA),stat = "identity",alpha=.5,
        fill='orange',color='orange4', show.legend = T)+
xlab("Velocidad del viento (m/s)")+
ylab("Energía (W/h)") +
geom_point(x=30, y =1000, shape=22, size=5,
        alpha=.1,fill='pink',color='red')+
geom_point(x=30, y =850,
        shape=22, size=5, alpha=.9,
        fill='lightblue',
        color='lightblue4')+
geom_point(x=30, y =700, shape=22,
        size=5, alpha=.3,fill='green2',
        color='green4')+
geom_point(x=30, y =550, shape=22,
        size=5, alpha=.2,fill='mediumorchid1',
        color='mediumorchid4')+
geom_point(x=30, y =400, shape=22,
        size=5,alpha=.1,fill='orange',
        color='orange4')+
annotate("text",label="Concentrador 30°", x = 35, y = 850)+
annotate("text",label="Concentrador 45°", x = 35, y = 1000)+
annotate("text",label="Concentrador 70°", x = 35, y = 700)+
annotate("text",label="Pared", x = 32.5, y = 550)+
annotate("text",label="Piloto", x = 32.5, y = 400)+
ggtitle("Comparativa de las energías
        anuales producidas empleando las 5 configuraciones")+
theme_bw()+
theme(plot.title = element_text(hjust = 0.5))

ggsave(paste0(path_here,"barplotenergíacomparativa_medida.tiff"),
        device = "tiff", dpi=1200,width =8,
        height =7, units = 'in')

return(as.data.frame(colSums(tabla_energia_est)))

}

#Obtención de la energía anual esperada
energia_anual_med<-barplot_energias_medida()

```