CT474: Smart Grid

Lecture 4: Weather and Grid Data

Dr. Jim Duggan,
School of Engineering & Informatics
National University of Ireland Galway.
https://github.com/JimDuggan/PDAR
https://twitter.com/_jimduggan

Relational Data with dplyr

- Typically, data analysis involves many tables of data that must be combined to answer questions
- Collectively, multiple tables of data are called *relational data*
- Relations are always defined between a pair of tables

key	val_x
1	x1
2	x2
3	x3

key	val_y
1	y1
2	y2
4	y3

Keys

- The variables used to connect each pair of tables are called keys
- **A key is a variable (or set of variables) that uniquely identifies an observation**
- There are two types of keys:
  - A ***primary key*** uniquely identifies an observation in its own table
  - A ***foreign key*** uniquely identifies an observation in another table.

Simple Example

name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

```
x <- data.frame(
  name = c("John", "Paul", "George", "Ringo", "Stuart", "Pete"),
  instrument = c("guitar","bass","guitar","drums","bass","drums"),
  stringsAsFactors = F
)
```

```
y <- data.frame(
  name = c("John", "Paul", "George", "Ringo", "Brian"),
  band = c(T,T,T,T,F),
  stringsAsFactors = F
)
```

Joining Tables x and y in dplyr

Type	Action
inner	Include only rows in **both** x and y
left	Include all of x, and matching rows of y
semi	Include rows of x that match y
anti	Include rows of x that **don't** match y

Type	Action
inner	Include only rows in **both** x and y

name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

```
> inner_join(x,y)
Joining, by = "name"
    name instrument band
1   John      guitar TRUE
2   Paul        bass TRUE
3 George      guitar TRUE
4  Ringo       drums TRUE
```

Type	Action
left	Include all of x, and matching rows of y

name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

```
> left_join(x,y)
Joining, by = "name"
    name instrument band
1   John     guitar TRUE
2   Paul       bass TRUE
3 George     guitar TRUE
4  Ringo      drums TRUE
5 Stuart       bass   NA
6   Pete      drums   NA
```

Type	Action
semi	Include rows of x that match y

name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

```
>
> semi_join(x,y)
Joining, by = "name"
    name instrument
1   John     guitar
2   Paul       bass
3 George     guitar
4  Ringo      drums
```

Type	Action
anti	Include rows of x that **don't** match y

name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

```
> anti_join(x,y)
Joining, by = "name"
    name instrument
1   Pete      drums
2 Stuart       bass
```

Type	Action
left	Include all of x, and matching rows of y

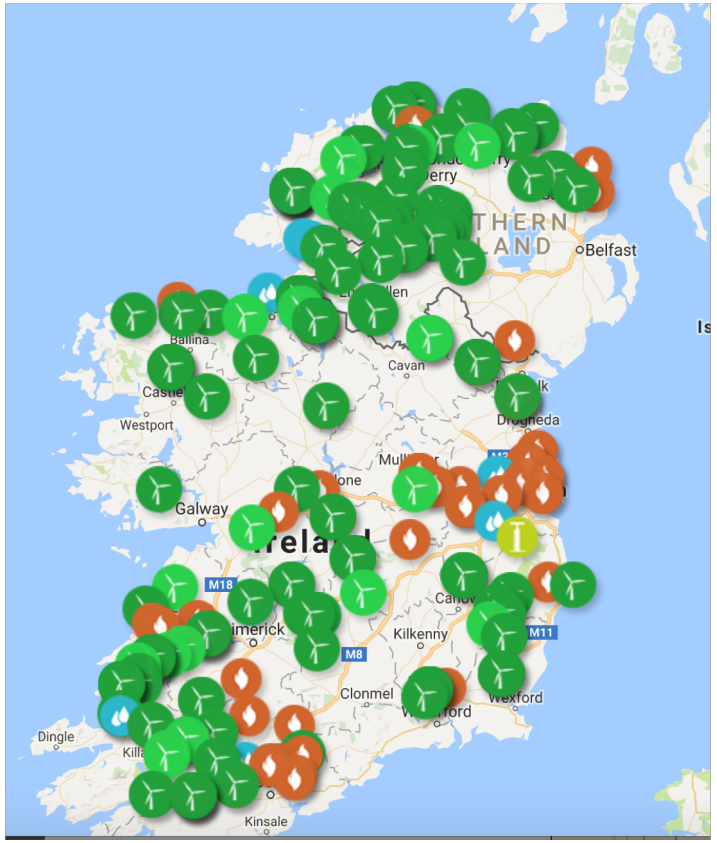
name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

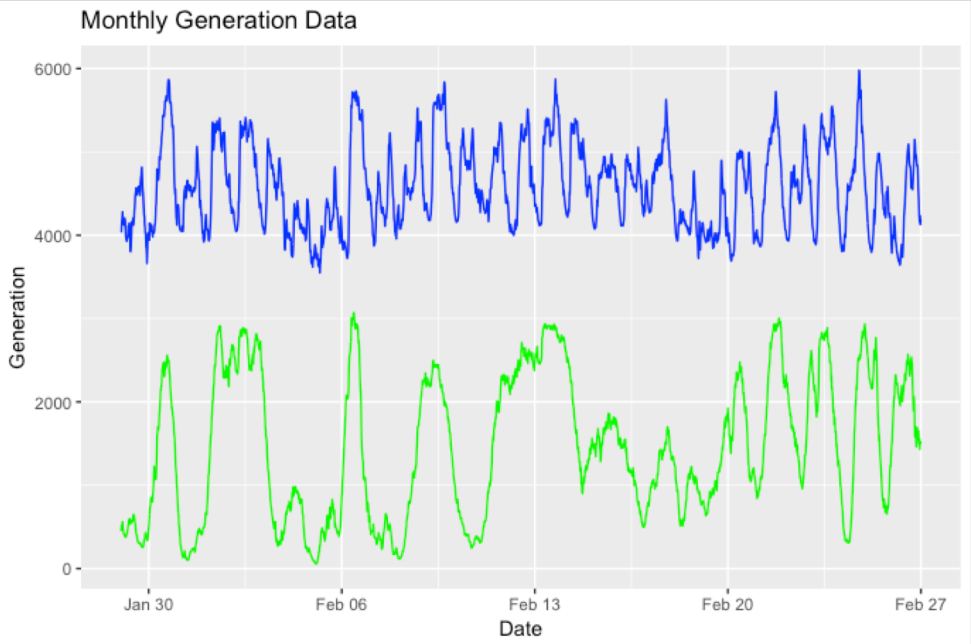
```
> left_join(x,y)
Joining, by = "name"
    name instrument band
1   John     guitar TRUE
2   Paul       bass TRUE
3 George     guitar TRUE
4  Ringo      drums TRUE
5 Stuart       bass   NA
6   Pete      drums   NA
```

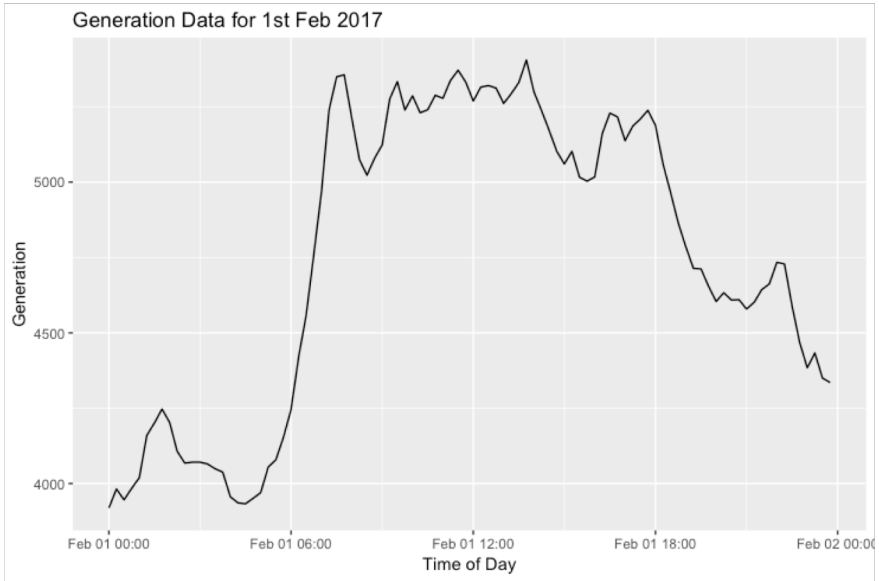
Generation Information

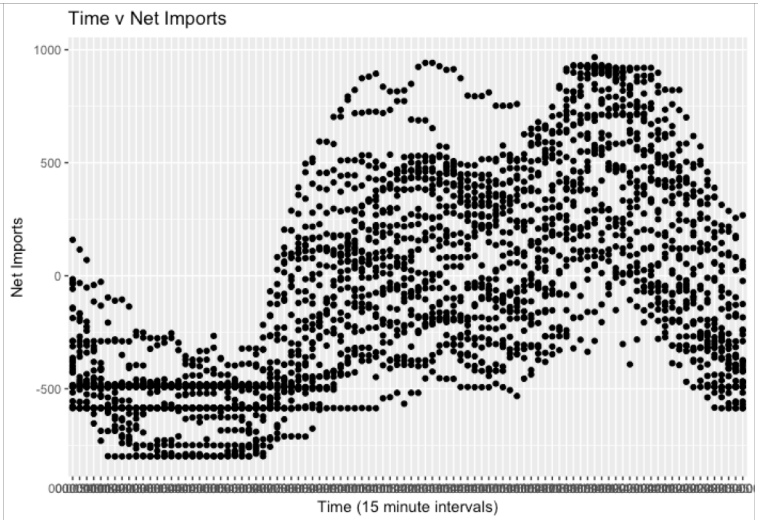
- Joining Data Sets
- Weather Data (Mace Head)
- Impact on Wind Generation
- Simple Prediction Model

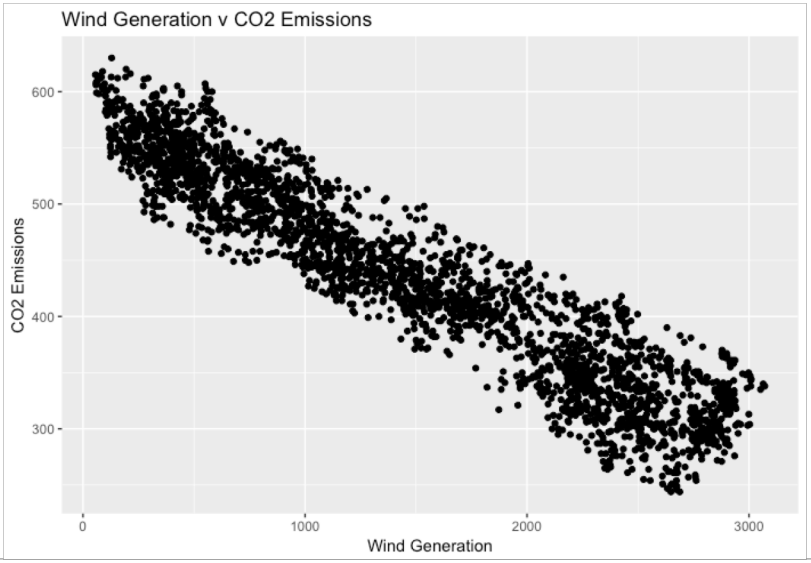


Recap – The Grid Data Set









Relational Data with dplyr (Wickham and Grolemund 2017)

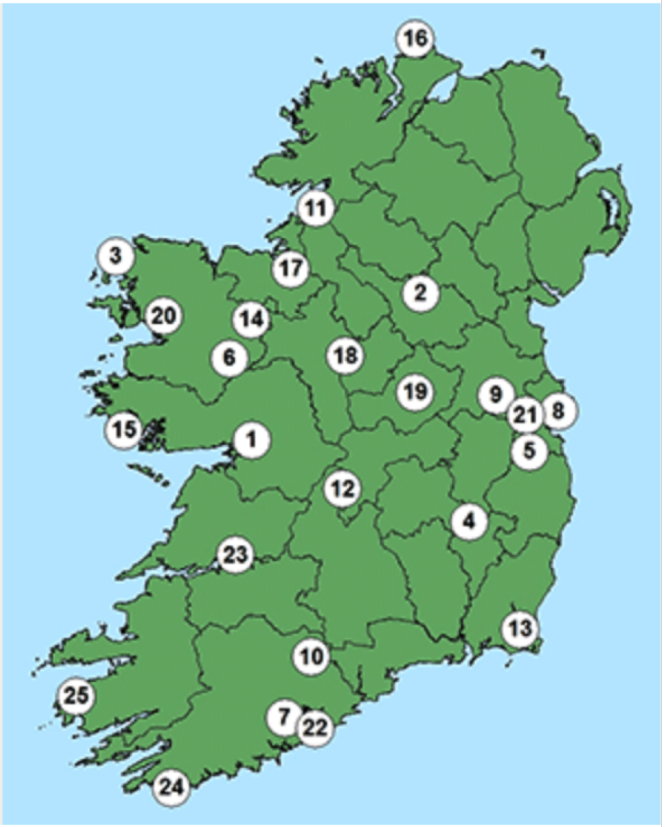
- It is rare that analysis only involves a single table of data
- Typically, you may have many tables of data and they must be combined to answer the questions you're interested in
- Multiple tables of data are called relational data, because the relations (not just the individual data sets) are important

Linking wind data and wind generation

Monthly Data

Please choose a monthly data report from any station by clicking one of the links below:

1	Athenry	14	Knock Airport
2	Ballyhaise	15	Mace Head
3	Belmullet	16	Malin Head
4	Carlow Oakpark	17	Markree
5	Baldonnel - Casement Aerodrome	18	Mount Dillon
6	Claremorris	19	Mullingar
7	Cork Airport	20	Newport
8	Dublin Airport	21	Phoenix Park
9	Dunsany	22	Roches Point
10	Fermoy Moorepark	23	Shannon Airport
11	Finner	24	Sherkin Island
12	Gurteen Agri College	25	Valentia Observatory
13	Johnstown		



http://www.met.ie

Linking wind data and wind generation

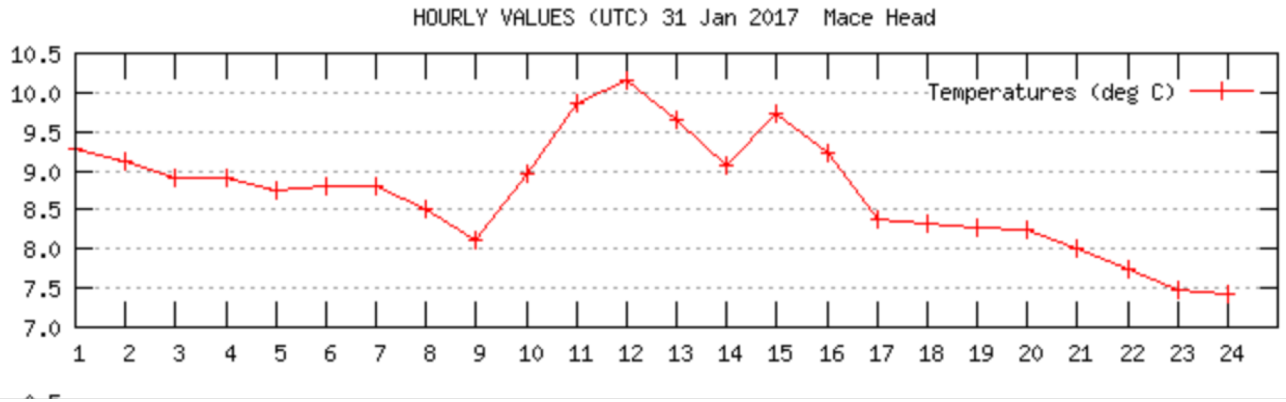
Daily Data

Weather Station Data - From 13/03/2015 to 12/03/2017

Please Select a Station and Date from the menu on the right.

REPORTS FROM MACE HEAD (A)

Date	Rainfall (mm)	Max Temp (°C)	Min Temp (°C)	Grass Min Temp (°C)	Mean Wind Speed (knots)	Maximum Gust (if >= 34 knots)	Sunshine (hours)
31/1/2017	0	10.3	7.3	5.8	10		



Select Station & Date

Station

Mace Head (A)

Date

31/01/2017

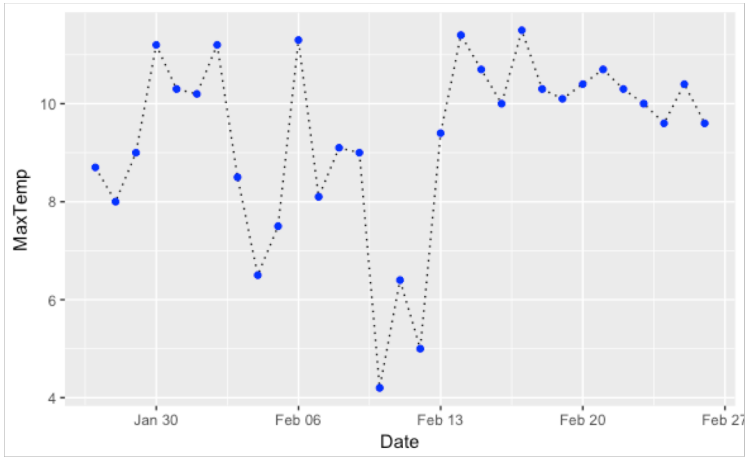
Go ->

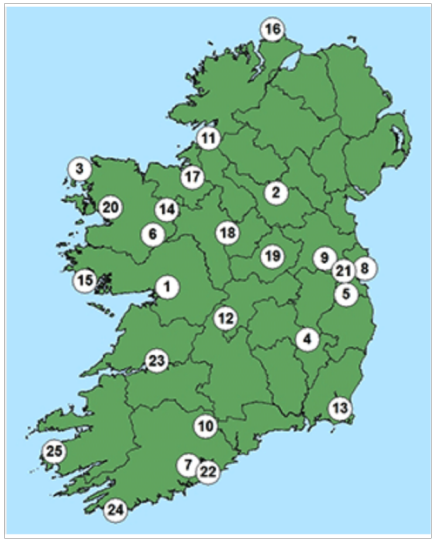
Synoptic Stations

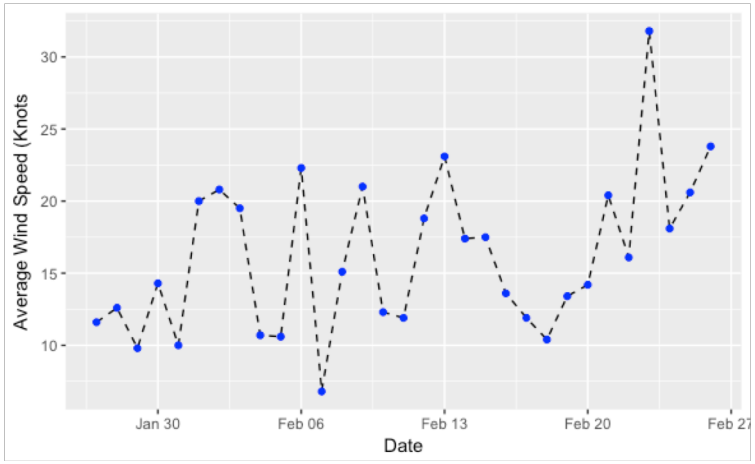
2011

http://www.met.ie

Mace Head Daily Data







Date	Rainfall	MaxTemp	MinTemp	GrassMinTemp	AVRWind	MaxWindGust
27/01/17	7.9	8.7	4.3	-0.7	11.6	
28/01/17	3.5	8	4.5	2.9	12.6	
29/01/17	4.7	9	4.9	3.7	9.8	
30/01/17	7.8	11.2	7.1	5.8	14.3	
31/01/17	0	10.3	7.3	5.8	10	
01/02/17	0.6	10.2	6.1	5.2	20	38
02/02/17	4.9	11.2	7.4	6.4	20.8	45
03/02/17	2.2	8.5	3.6	2.1	19.5	46
04/02/17	5.3	6.5	1.8	-1.3	10.7	

Exploring Data Sets:
Different time recordings of observations

```
> ener[1:5,]
# A tibble: 5 × 13
             DateTime       Date     Time Demand Generation  Wind   CO2 NetImports  EWIC Moyle
               <dttm>      <chr>    <chr>  <int>      <int> <int> <int>      <int> <int> <int>
1 2017-01-29 00:00:00 2017-01-29 00:00:00   3834       4041   449   552       -145   -33  -112
2 2017-01-29 00:15:00 2017-01-29 00:15:00   3785       4041   505   548       -200  -108   -92
3 2017-01-29 00:30:00 2017-01-29 00:30:00   3708       4130   521   544       -294  -183  -111
4 2017-01-29 00:45:00 2017-01-29 00:45:00   3634       4181   492   543       -419  -258  -161
5 2017-01-29 01:00:00 2017-01-29 01:00:00   3581       4211   538   555       -503  -333  -170
# ... with 3 more variables: HourOfDay <int>, MinuteOfDay <int>, DayOfWeek <ord>
>
>
> weather[1:5,]
# A tibble: 5 × 7
        Date Rainfall MaxTemp MinTemp GrassMinTemp AVRWind MaxWindGust
      <date>    <dbl>   <dbl>   <dbl>        <dbl>   <dbl>       <int>
1 2017-01-27      7.9     8.7     4.3         -0.7    11.6          NA
2 2017-01-28      3.5     8.0     4.5          2.9    12.6          NA
3 2017-01-29      4.7     9.0     4.9          3.7     9.8          NA
4 2017-01-30      7.8    11.2     7.1          5.8    14.3          NA
5 2017-01-31      0.0    10.3     7.3          5.8    10.0          NA
```

Approach

Need to find the average generation by wind from grid data

```
avr_daily_wind <- ener %>% group_by(Date) %>%
  summarise(AverageWindGeneration=mean(Wind)) %>%
  mutate(Date=ymd(Date))
```

```
> avr_daily_wind[1:5,]
# A tibble: 5 × 2
        Date AverageWindGeneration
      <date>                  <dbl>
1 2017-01-29               431.3125
2 2017-01-30              1725.9375
3 2017-01-31               330.3333
4 2017-02-01              2046.5521
5 2017-02-02              2647.0000
```

Need to combine the data sets

```
> avr_daily_wind[1:5,]
# A tibble: 5 × 2
        Date AverageWindGeneration
       <date>                 <dbl>
1 2017-01-29              431.3125
2 2017-01-30             1725.9375
3 2017-01-31              330.3333
4 2017-02-01             2046.5521
5 2017-02-02             2647.0000
```

```
> weather[1:5,]
# A tibble: 5 × 7
        Date Rainfall MaxTemp MinTemp GrassMinTemp AVRWind MaxWindGust
       <date>    <dbl>   <dbl>   <dbl>        <dbl>   <dbl>       <int>
1 2017-01-27      7.9     8.7     4.3         -0.7    11.6          NA
2 2017-01-28      3.5     8.0     4.5          2.9    12.6          NA
3 2017-01-29      4.7     9.0     4.9          3.7     9.8          NA
4 2017-01-30      7.8    11.2     7.1          5.8    14.3          NA
5 2017-01-31      0.0    10.3     7.3          5.8    10.0          NA
```

```
> avr_daily_wind
# A tibble: 29 × 2
         Date AverageWindGeneration
       <date>                  <dbl>
1  2017-01-29               431.3125
2  2017-01-30              1725.9375
3  2017-01-31               330.3333
4  2017-02-01              2046.5521
5  2017-02-02              2647.0000
6  2017-02-03              1049.7500
7  2017-02-04               590.7708
8  2017-02-05               439.4688
9  2017-02-06              1970.5833
10 2017-02-07               393.7604
# ... with 19 more rows
```

```
> select(weather,Date,AVRWind,everything())
# A tibble: 31 × 7
         Date AVRWind Rainfall MaxTemp MinTemp GrassMinTemp
       <date>   <dbl>    <dbl>   <dbl>   <dbl>        <dbl>
1  2017-01-27    11.6      7.9     8.7     4.3         -0.7
2  2017-01-28    12.6      3.5     8.0     4.5          2.9
3  2017-01-29     9.8      4.7     9.0     4.9          3.7
4  2017-01-30    14.3      7.8    11.2     7.1          5.8
5  2017-01-31    10.0      0.0    10.3     7.3          5.8
6  2017-02-01    20.0      0.6    10.2     6.1          5.2
7  2017-02-02    20.8      4.9    11.2     7.4          6.4
8  2017-02-03    19.5      2.2     8.5     3.6          2.1
9  2017-02-04    10.7      5.3     6.5     1.8         -1.3
10 2017-02-05    10.6      6.9     7.5     2.2         -1.4
# ... with 21 more rows, and 1 more variables:
#   MaxWindGust <int>
```

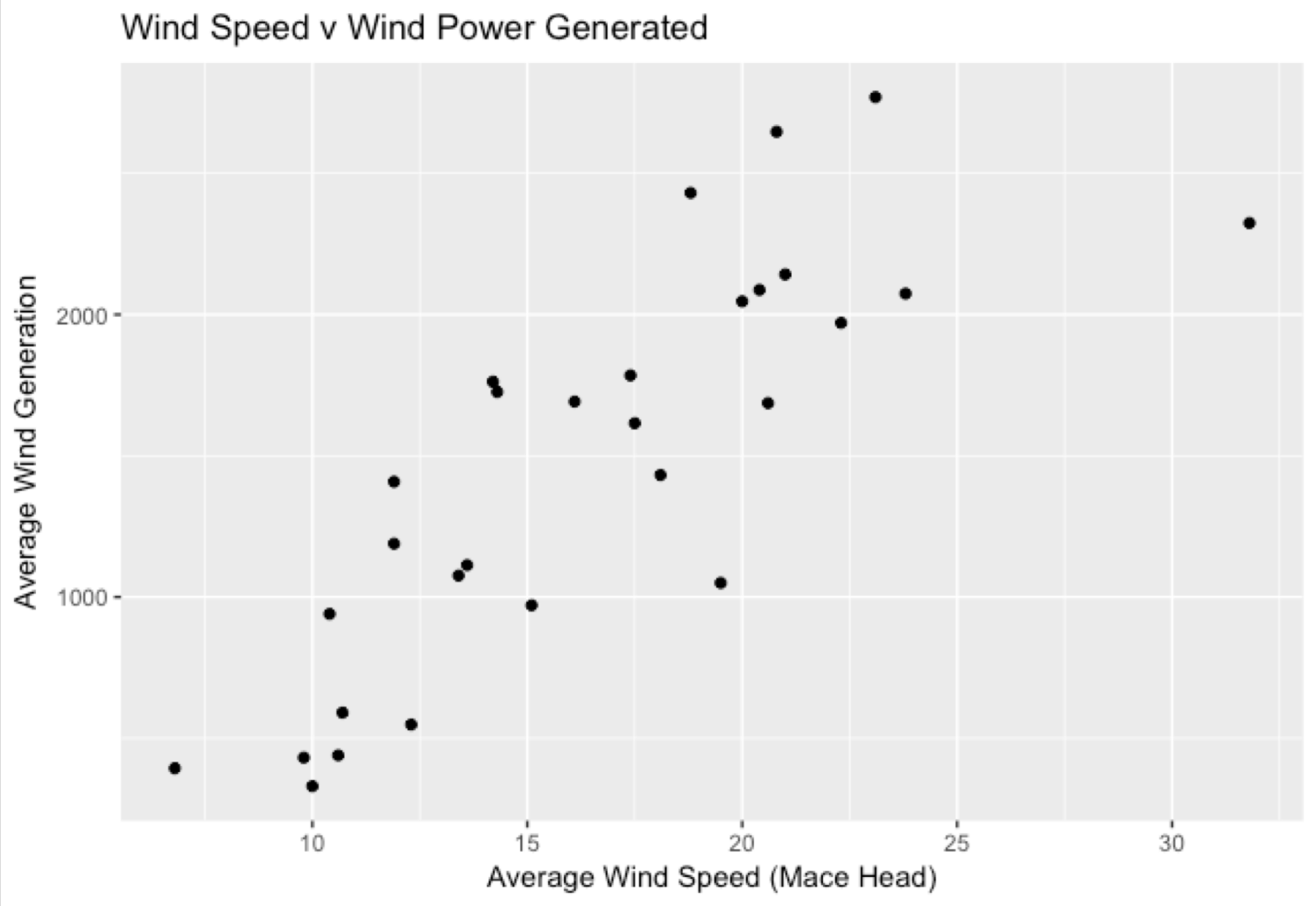
```
gen_weather <- left_join(avr_daily_wind,weather) %>%
                select(Date,AVRWind,AverageWindGeneration)
```

```
> left_join(avr_daily_wind,weather)
Joining, by = "Date"
# A tibble: 29 × 8
         Date AverageWindGeneration Rainfall MaxTemp MinTemp GrassMinTemp AVRWind MaxWindGust
       <date>                 <dbl>    <dbl>   <dbl>   <dbl>        <dbl>   <dbl>       <int>
1  2017-01-29              431.3125      4.7     9.0     4.9          3.7     9.8          NA
2  2017-01-30             1725.9375      7.8    11.2     7.1          5.8    14.3          NA
3  2017-01-31              330.3333      0.0    10.3     7.3          5.8    10.0          NA
4  2017-02-01             2046.5521      0.6    10.2     6.1          5.2    20.0          38
5  2017-02-02             2647.0000      4.9    11.2     7.4          6.4    20.8          45
6  2017-02-03             1049.7500      2.2     8.5     3.6          2.1    19.5          46
```

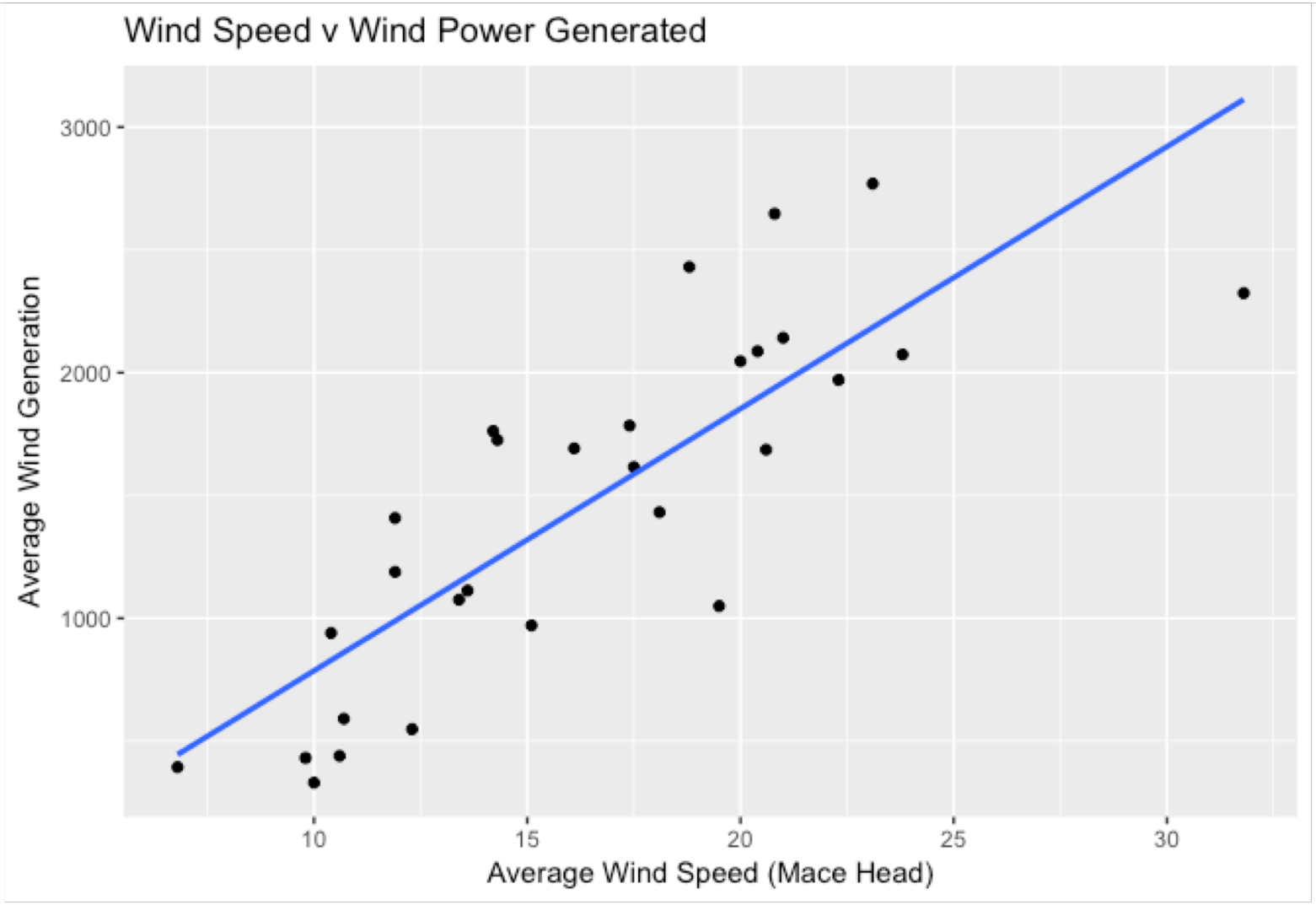
```
gen_weather <- left_join(avr_daily_wind,weather) %>%
                select(Date,AVRWind,AverageWindGeneration)
```

```
> gen_weather
# A tibble: 29 × 3
         Date AVRWind AverageWindGeneration
       <date>   <dbl>                 <dbl>
1  2017-01-29     9.8              431.3125
2  2017-01-30    14.3             1725.9375
3  2017-01-31    10.0              330.3333
4  2017-02-01    20.0             2046.5521
5  2017-02-02    20.8             2647.0000
6  2017-02-03    19.5             1049.7500
```

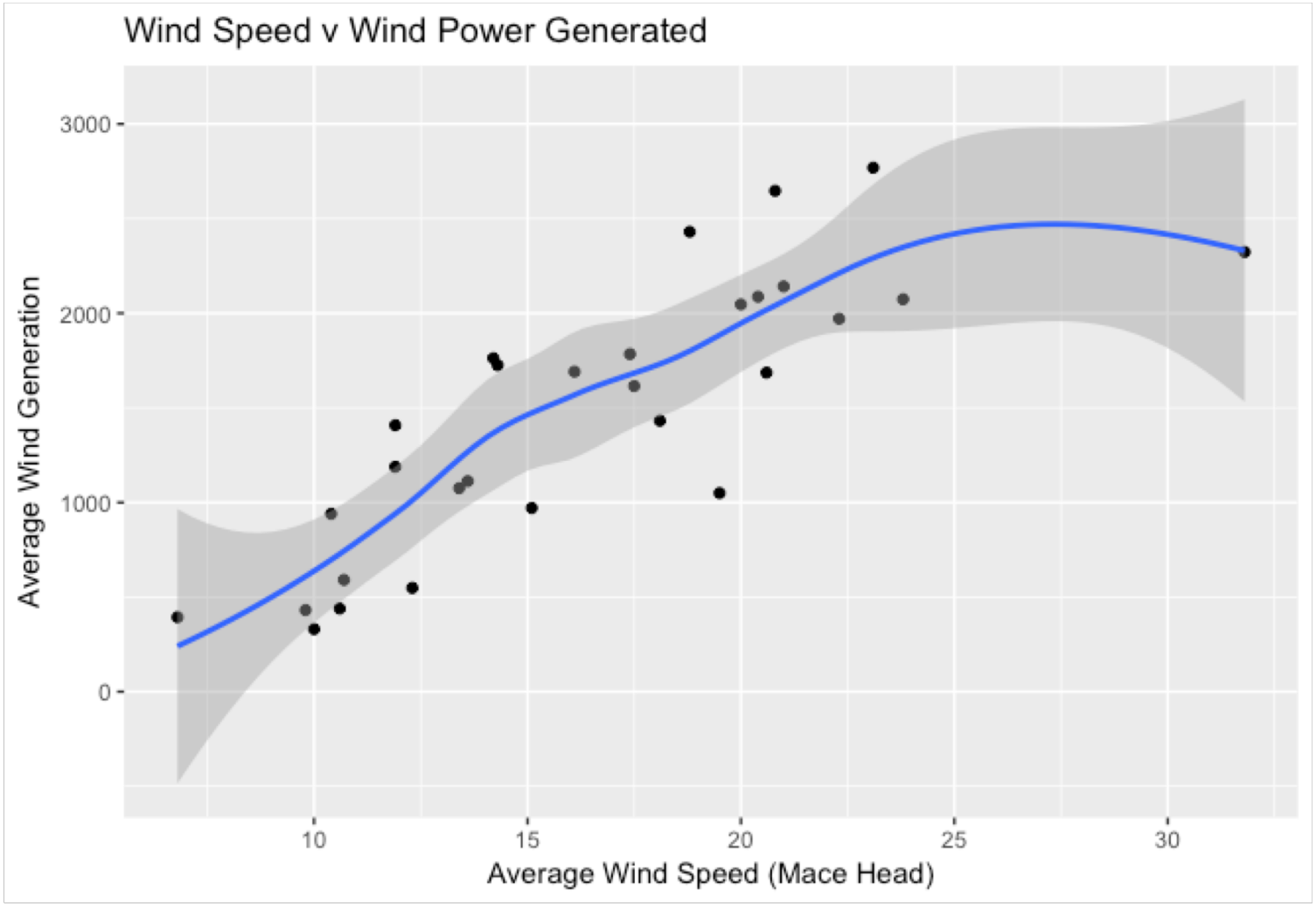
```
ggplot(data = gen_weather,mapping = aes(x=AVRWind,y=AverageWindGeneration)) +
  geom_point() +
  xlab("Average Wind Speed (Mace Head)") + ylab("Average Wind Generation") +
  ggtitle("Wind Speed v Wind Power Generated")
```



```
ggplot(data = gen_weather,mapping = aes(x=AVRWind,y=AverageWindGeneration)) +
  geom_point() +
  geom_smooth(method = "lm",se=F)+
  xlab("Average Wind Speed (Mace Head)") + ylab("Average Wind Generation") +
  ggtitle("Wind Speed v Wind Power Generated")
```



```
ggplot(data = gen_weather,mapping = aes(x=AVRWind,y=AverageWindGeneration)) +
  geom_point() +
  geom_smooth()+
  xlab("Average Wind Speed (Mace Head)") + ylab("Average Wind Generation") +
  ggtitle("Wind Speed v Wind Power Generated")
```



Creating a linear model (lm)

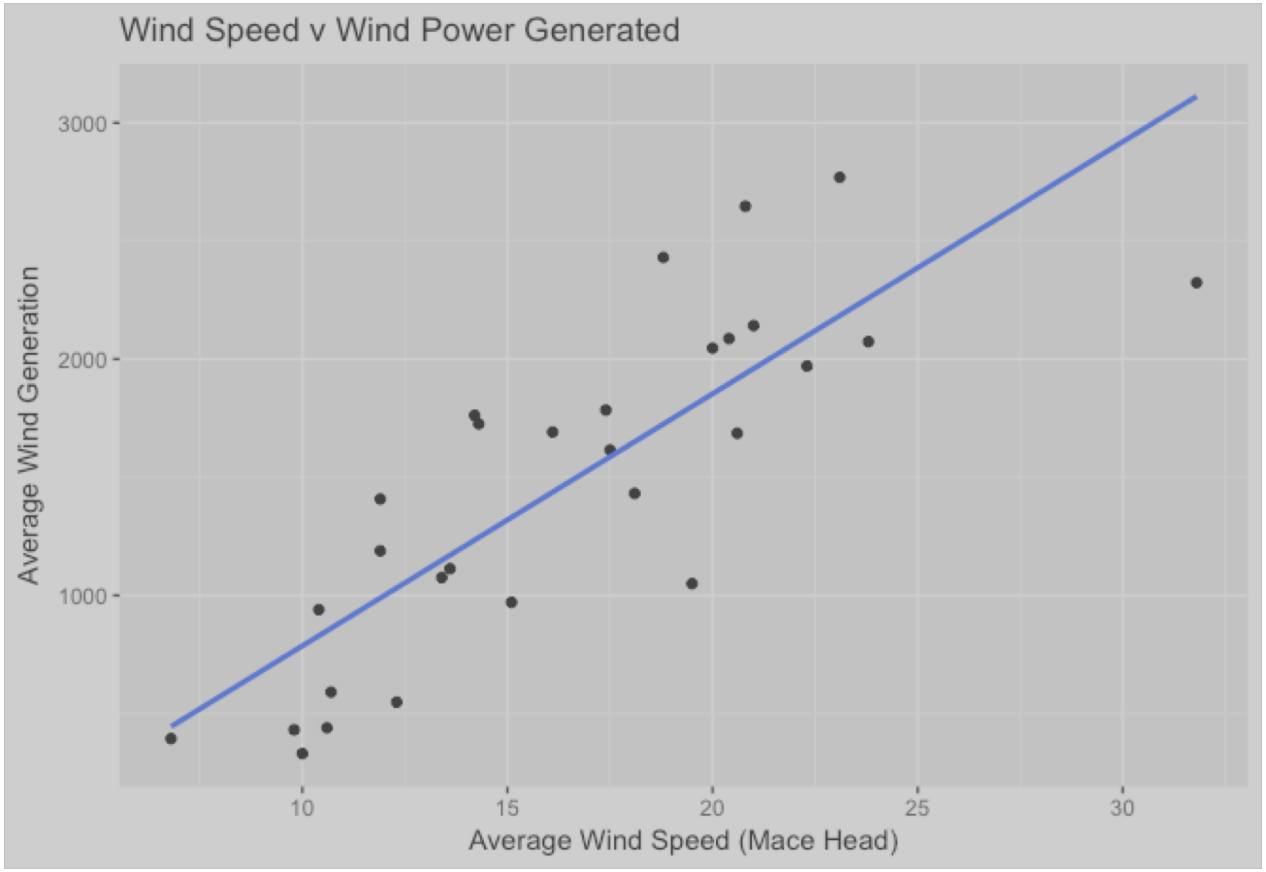
```
> gen_weather
# A tibble: 29 × 3
         Date AVRWind AverageWindGeneration
        <date>   <dbl>                 <dbl>
1  2017-01-29     9.8                431.3125
2  2017-01-30    14.3               1725.9375
3  2017-01-31    10.0                330.3333
4  2017-02-01    20.0               2046.5521
5  2017-02-02    20.8               2647.0000
```

```
> mod <- lm(data = gen_weather,AverageWindGeneration~AVRWind)
>
> mod

Call:
lm(formula = AverageWindGeneration ~ AVRWind, data = gen_weather)

Coefficients:
(Intercept)      AVRWind
     -280.8        106.7
```

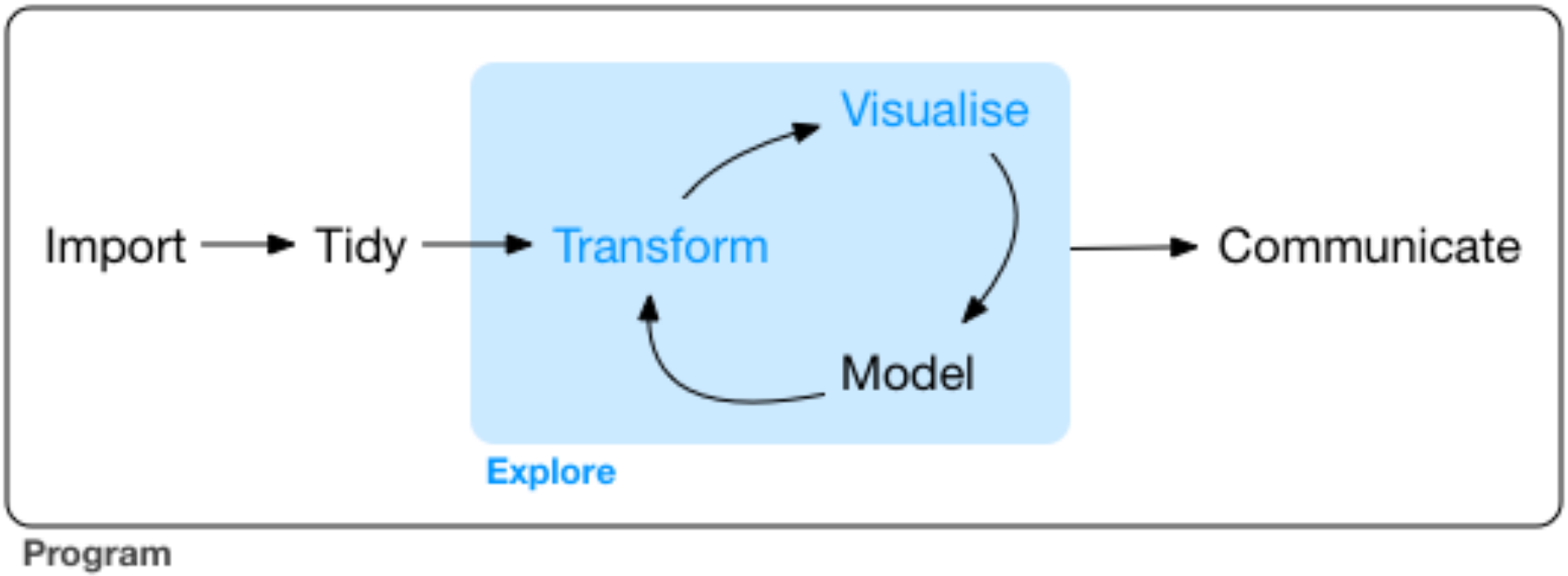
Predicting values…




```
> p1 <- predict(mod,newdata = data.frame(AVRWind=25))
>
> p1
       1
2386.727
```

Summary

"Data exploration is the art of looking at your data, rapidly generating hypotheses, quickly testing them, then repeating again and again and again." (Wickham and Grolemund 2017).



*"**Analytics** involves the systematic use of data and related insights developed through **applied analytical methods** to drive **evidence-based decision making** for planning, management, measurement and learning" (Zodpey 2016)*