

## Case study: windmill

# The windmill data

- Engineer: does amount of electricity generated by windmill depend on how strongly wind blowing?
- Measurements of wind speed and DC current generated at various times.
- Assume the “various times” to be randomly selected — aim to generalize to “this windmill at all times”.
- Research questions:
  - ▶ Relationship between wind speed and current generated?
  - ▶ If so, what kind of relationship?
  - ▶ Can we model relationship to do predictions?

## Packages for this section

```
library(tidyverse)  
library(broom)
```

## Reading in the data

```
my_url <-  
  "http://ritsokiguess.site/datafiles/windmill.csv"  
windmill <- read_csv(my_url)  
windmill
```

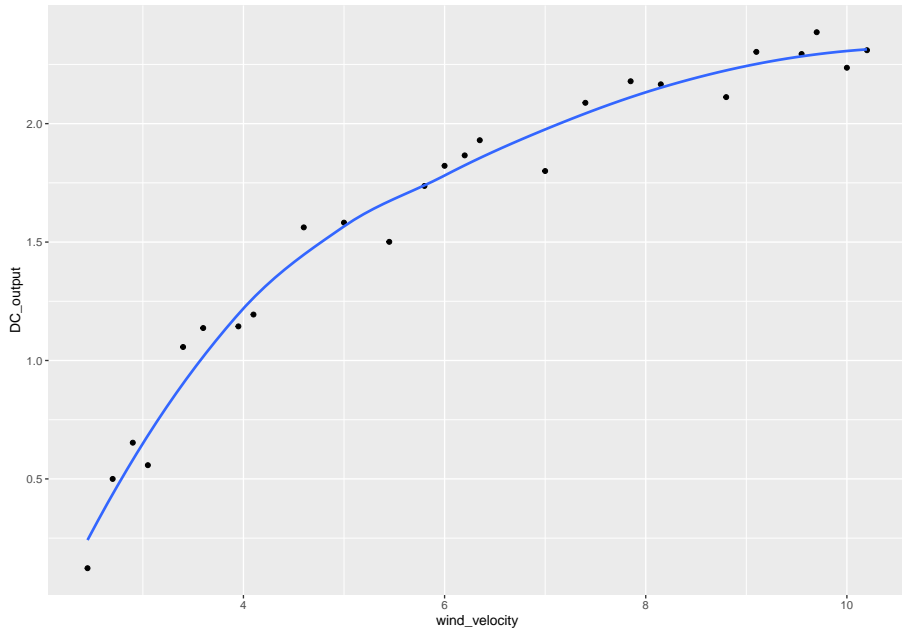
```
# A tibble: 25 x 2  
  wind_velocity DC_output  
      <dbl>      <dbl>  
1         5      1.58  
2         6      1.82  
3        3.4      1.06  
4        2.7      0.5  
5        10      2.24  
6         9.7      2.39  
7        9.55      2.29  
8        3.05      0.558  
9        8.15      2.17  
10       6.2      1.87  
# i 15 more rows
```

# Strategy

- Two quantitative variables, looking for relationship: regression methods.
- Start with picture (scatterplot).
- Fit models and do model checking, fixing up things as necessary.
- Scatterplot:
  - ▶ 2 variables, DC\_output and wind\_velocity.
  - ▶ First is output/response, other is input/explanatory.
  - ▶ Put DC\_output on vertical scale.
- Add trend, but don't want to assume linear:

```
ggplot(windmill, aes(y = DC_output, x = wind_velocity)) +  
  geom_point() + geom_smooth()
```

# Scatterplot



# Comments

- Definitely a relationship: as wind velocity increases, so does DC output. (As you'd expect.)
- Is relationship linear? To help judge, `geom_smooth` smooths scatterplot trend. (Trend called “loess”, “Locally weighted least squares” which downweights outliers. Not constrained to be straight.)
- Trend more or less linear for while, then curves downwards (levelling off?). Straight line not so good here.

## Fit a straight line (and see what happens)

```
DC.1 <- lm(DC_output ~ wind_velocity, data = windmill)
summary(DC.1)
```

Call:

```
lm(formula = DC_output ~ wind_velocity, data = windmill)
```

Residuals:

|  | Min      | 1Q       | Median  | 3Q      | Max     |
|--|----------|----------|---------|---------|---------|
|  | -0.59869 | -0.14099 | 0.06059 | 0.17262 | 0.32184 |

Coefficients:

|               | Estimate | Std. Error | t value | Pr(> t )     |
|---------------|----------|------------|---------|--------------|
| (Intercept)   | 0.13088  | 0.12599    | 1.039   | 0.31         |
| wind_velocity | 0.24115  | 0.01905    | 12.659  | 7.55e-12 *** |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2361 on 23 degrees of freedom

Multiple R-squared: 0.8745, Adjusted R-squared: 0.869

F-statistic: 160.3 on 1 and 23 DF, p-value: 7.546e-12



## Another way of looking at the output

- The standard output tends to go off the bottom of the page rather easily. Package broom has these:

```
glance(DC.1)
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
  <dbl>      <dbl> <dbl>    <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl>
1    0.874    0.869 0.236    160. 7.55e-12     1   1.66   2.68   6.33
# i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

showing that the R-squared is 87%, and

```
tidy(DC.1)
```

```
# A tibble: 2 x 5
  term          estimate std.error statistic p.value
  <chr>          <dbl>    <dbl>    <dbl>    <dbl>
1 (Intercept)    0.131    0.126     1.04 3.10e- 1
2 wind_velocity  0.241    0.0190    12.7 7.55e-12
```

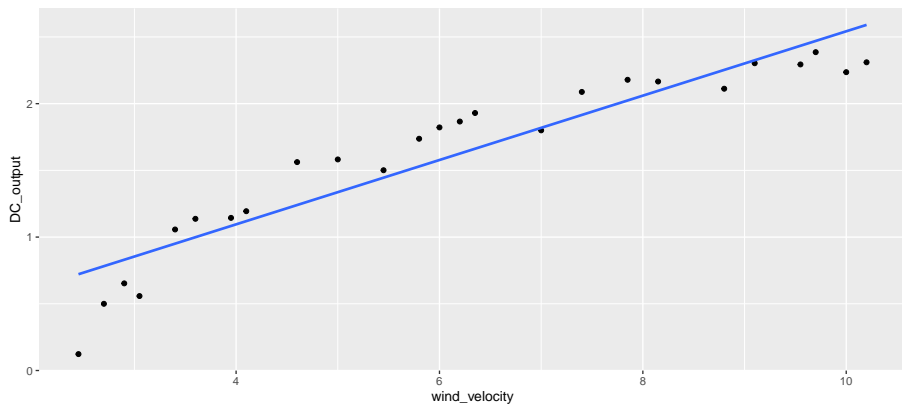
showing the intercept and slope and their significance.

# Comments

- Strategy: `lm` actually fits the regression. Store results in a variable. Then look at the results, eg. via `summary` or `glance/tidy`.
- My strategy for model names: base on response variable (or data frame name) and a number. Allows me to fit several models to same data and keep track of which is which.
- Results actually pretty good: `wind.velocity` strongly significant, R-squared (87%) high.
- How to check whether regression is appropriate? Look at the residuals, observed minus predicted, plotted against fitted (predicted).
- Plot using the regression object as “data frame” (in a couple of slides).

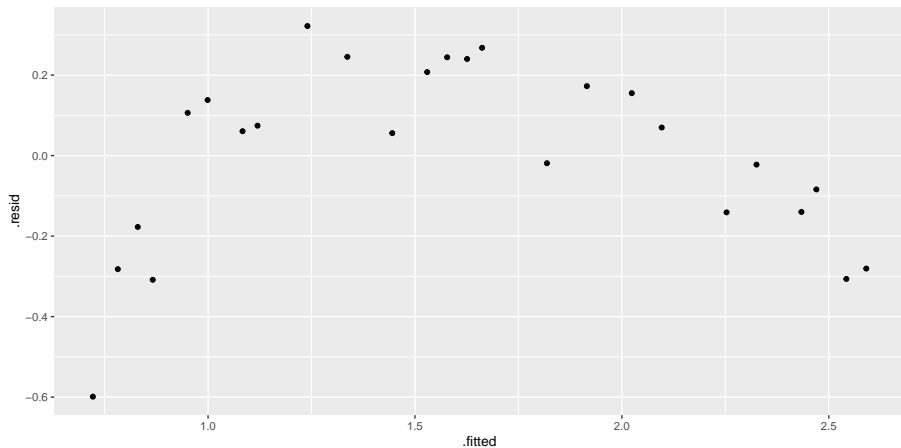
## Scatterplot, but with line

```
ggplot(windmill, aes(y = DC_output, x = wind_velocity)) +  
  geom_point() + geom_smooth(method="lm", se = FALSE)
```



# Plot of residuals against fitted values

```
ggplot(DC.1, aes(y = .resid, x = .fitted)) + geom_point()
```

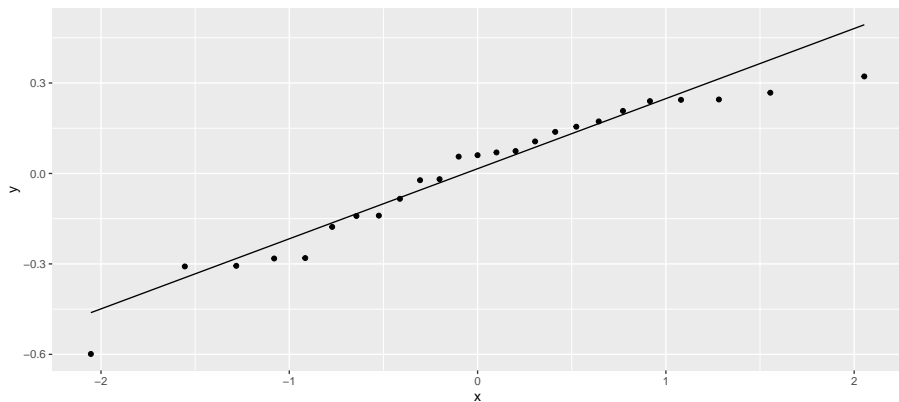


## Comments on residual plot

- Residual plot should be a random scatter of points.
- Should be no pattern “left over” after fitting the regression.
- Smooth trend should be more or less straight across at 0.
- Here, have a curved trend on residual plot.
- This means original relationship must have been a curve (as we saw on original scatterplot).
- Possible ways to fit a curve:
  - ▶ Add a squared term in explanatory variable.
  - ▶ Transform response variable (doesn't work well here).
  - ▶ See what science tells you about mathematical form of relationship, and try to apply.

## normal quantile plot of residuals

```
ggplot(DC.1, aes(sample = .resid)) +  
  stat_qq() + stat_qq_line()
```



# Parabolas and fitting parabola model

- A parabola has equation

$$y = ax^2 + bx + c$$

with coefficients  $a, b, c$ . About the simplest function that is not a straight line.

- Fit one using `lm` by adding  $x^2$  to right side of model formula with `+`:

```
DC.2 <- lm(DC_output ~ wind_velocity + I(wind_velocity^2),  
  data = windmill  
)
```

- The `I()` necessary because `^` in model formula otherwise means something different (to do with interactions in ANOVA).
- Call it *parabola model*.

# Parabola model output

```
summary(DC.2)
```

Call:

```
lm(formula = DC_output ~ wind_velocity + I(wind_velocity^2),  
    data = windmill)
```

Residuals:

|  | Min      | 1Q       | Median  | 3Q      | Max     |
|--|----------|----------|---------|---------|---------|
|  | -0.26347 | -0.02537 | 0.01264 | 0.03908 | 0.19903 |

Coefficients:

|                    | Estimate  | Std. Error | t value | Pr(> t ) |     |
|--------------------|-----------|------------|---------|----------|-----|
| (Intercept)        | -1.155898 | 0.174650   | -6.618  | 1.18e-06 | *** |
| wind_velocity      | 0.722936  | 0.061425   | 11.769  | 5.77e-11 | *** |
| I(wind_velocity^2) | -0.038121 | 0.004797   | -7.947  | 6.59e-08 | *** |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1227 on 22 degrees of freedom

Multiple R-squared: 0.9676, Adjusted R-squared: 0.9646

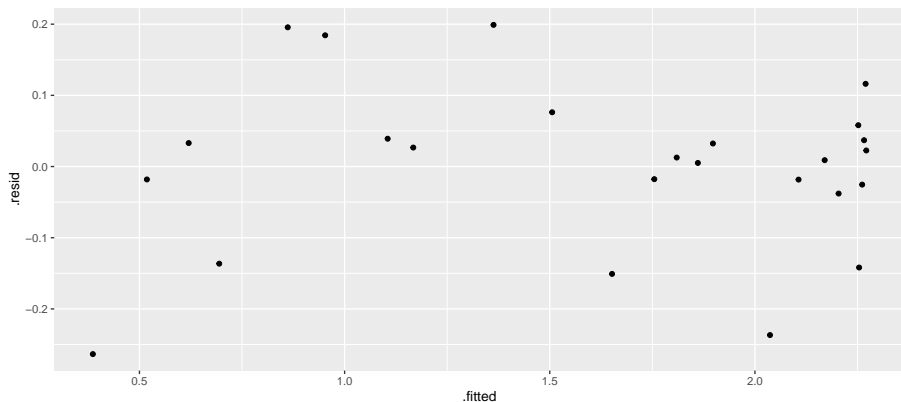


## Comments on output

- R-squared has gone up a lot, from 87% (line) to 97% (parabola).
- Coefficient of squared term strongly significant (P-value  $6.59 \times 10^{-8}$ ).
- Adding squared term has definitely improved fit of model.
- Parabola model better than linear one.
- But...need to check residuals again.

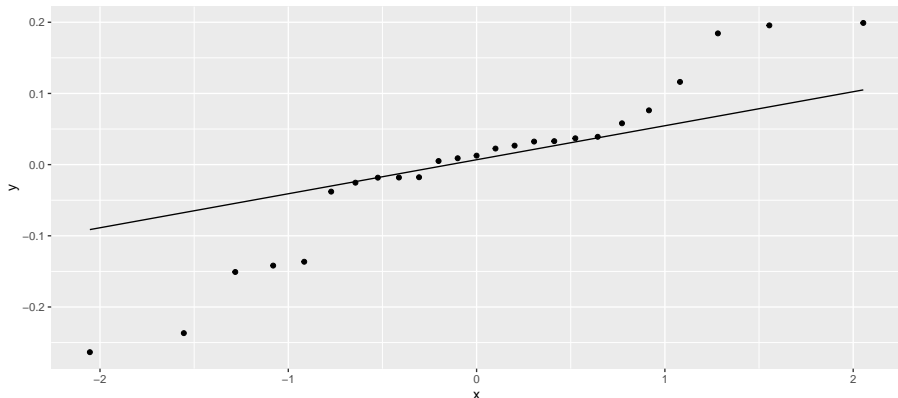
# Residual plot from parabola model

```
ggplot(DC.2, aes(y = .resid, x = .fitted)) +  
  geom_point()
```



## normal quantile plot of residuals

```
ggplot(DC.2, aes(sample = .resid)) + stat_qq() + stat_qq_line()
```



This distribution has long tails, which should worry us at least some.

# Make scatterplot with fitted line and curve

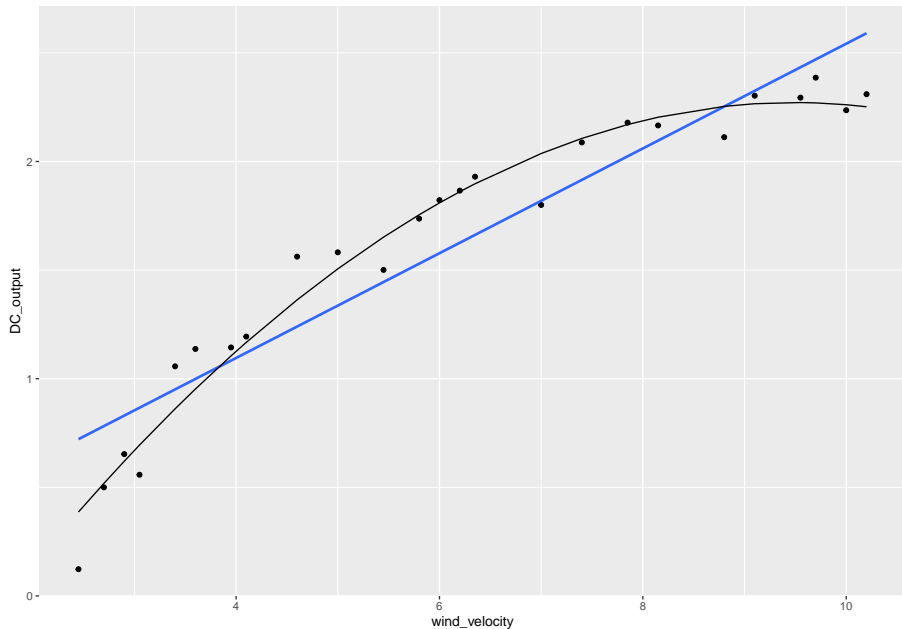
- Residual plot basically random. Good.
- Scatterplot with fitted line and curve like this:

```
ggplot(windmill, aes(y = DC_output, x = wind_velocity)) +  
  geom_point() + geom_smooth(method = "lm", se = F) +  
  geom_line(data = DC.2, aes(y = .fitted))
```

# Comments

- This plots:
  - ▶ scatterplot (`geom_point`);
  - ▶ straight line (via tweak to `geom_smooth`, which draws best-fitting line);
  - ▶ fitted curve, using the predicted `DC_output` values, joined by lines (with points not shown).
- Trick in the `geom_line` is use the predictions as the y-points to join by lines (from `DC.2`), instead of the original data points. Without the data and `aes` in the `geom_line`, original data points would be joined by lines.

# Scatterplot with fitted line and curve



## Another approach to a curve

- There is a problem with parabolas, which we'll see later.
- Ask engineer, “what should happen as wind velocity increases?”:
  - ▶ Upper limit on electricity generated, but otherwise, the larger the wind velocity, the more electricity generated.
- Mathematically, *asymptote*. Straight lines and parabolas don't have them, but eg.  $y = 1/x$  does: as  $x$  gets bigger,  $y$  approaches zero without reaching it.
- What happens to  $y = a + b(1/x)$  as  $x$  gets large?
  - ▶  $y$  gets closer and closer to  $a$ : that is,  $a$  is asymptote.
- Fit this, call it asymptote model.
- Fitting the model here because we have math to justify it.
  - ▶ Alternative,  $y = a + be^{-x}$ , approaches asymptote faster.

## How to fit asymptote model?

- Define new explanatory variable to be  $1/x$ , and predict  $y$  from it.
- $x$  is velocity, distance over time.
- So  $1/x$  is time over distance. In walking world, if you walk 5 km/h, take 12 minutes to walk 1 km, called your pace. So 1 over wind\_velocity we call wind\_pace.
- Make a scatterplot first to check for straightness (next page).

```
windmill %>% mutate(wind_pace = 1 / wind_velocity) -> windmill
ggplot(windmill, aes(y = DC_output, x = wind_pace)) +
  geom_point() + geom_smooth(se = F)
```



and run regression like this:

```
DC.3 <- lm(DC_output ~ wind_pace, data = windmill)
summary(DC.3)
```

Call:

```
lm(formula = DC_output ~ wind_pace, data = windmill)
```

Residuals:

|  | Min      | 1Q       | Median  | 3Q      | Max     |
|--|----------|----------|---------|---------|---------|
|  | -0.20547 | -0.04940 | 0.01100 | 0.08352 | 0.12204 |

Coefficients:

|             | Estimate | Std. Error | t value | Pr(> t )   |
|-------------|----------|------------|---------|------------|
| (Intercept) | 2.9789   | 0.0449     | 66.34   | <2e-16 *** |
| wind_pace   | -6.9345  | 0.2064     | -33.59  | <2e-16 *** |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

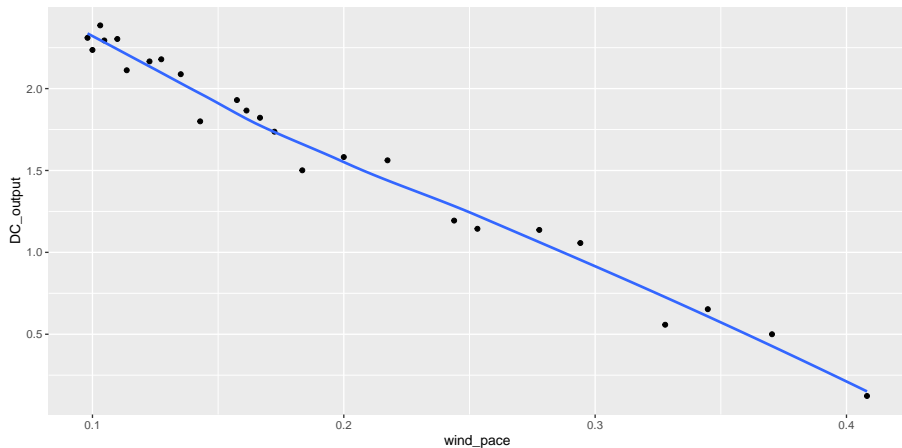
Residual standard error: 0.09417 on 23 degrees of freedom

Multiple R-squared: 0.98, Adjusted R-squared: 0.9792

F-statistic: 1128 on 1 and 23 DF, p-value: < 2.2e-16

## Scatterplot for wind\_pace

Pretty straight. Blue actually smooth curve not line:



# Regression output

```
summary(DC.3)
```

Call:

```
lm(formula = DC_output ~ wind_pace, data = windmill)
```

Residuals:

|  | Min      | 1Q       | Median  | 3Q      | Max     |
|--|----------|----------|---------|---------|---------|
|  | -0.20547 | -0.04940 | 0.01100 | 0.08352 | 0.12204 |

Coefficients:

|             | Estimate | Std. Error | t value | Pr(> t )   |
|-------------|----------|------------|---------|------------|
| (Intercept) | 2.9789   | 0.0449     | 66.34   | <2e-16 *** |
| wind_pace   | -6.9345  | 0.2064     | -33.59  | <2e-16 *** |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09417 on 23 degrees of freedom

Multiple R-squared: 0.98, Adjusted R-squared: 0.9792

F-statistic: 1128 on 1 and 23 DF, p-value: < 2.2e-16

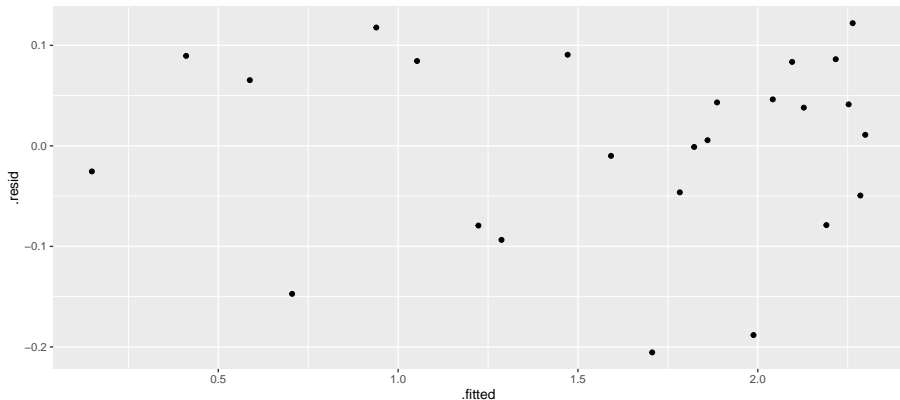
```
tidy(DC.3)
```

# Comments

- R-squared, 98%, even higher than for parabola model (97%).
- Simpler model, only one explanatory variable (`wind.pace`) vs. 2 for parabola model (`wind.velocity` and its square).
- `wind.pace` (unsurprisingly) strongly significant.
- Looks good, but check residual plot (over).

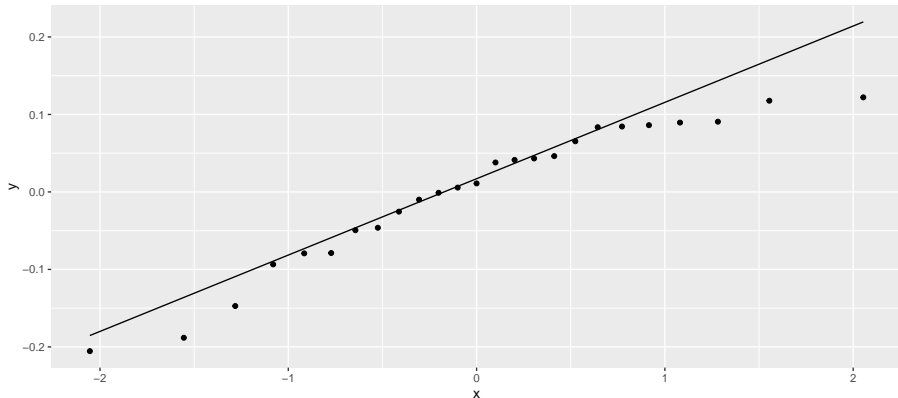
# Residual plot for asymptote model

```
ggplot(DC.3, aes(y = .resid, x = .fitted)) + geom_point()
```



## normal quantile plot of residuals

```
ggplot(DC.3, aes(sample = .resid)) +  
  stat_qq() + stat_qq_line()
```



This is skewed (left), but is not bad (and definitely better than the one for the parabola model).

## Plotting trends on scatterplot

- Residual plot not bad. But residuals go up to 0.10 and down to  $-0.20$ , suggesting possible skewness (not normal). I think it's not perfect, but OK overall.
- Next: plot scatterplot with all three fitted lines/curves on it (for comparison), with legend saying which is which.
- First make data frame containing what we need, taken from the right places:

```
w2 <- tibble(  
  wind_velocity = windmill$wind_velocity,  
  DC_output = windmill$DC_output,  
  linear = fitted(DC.1),  
  parabola = fitted(DC.2),  
  asymptote = fitted(DC.3)  
)
```

# What's in w2

w2

```
# A tibble: 25 x 5
```

|    | wind_velocity | DC_output | linear | parabola | asymptote |
|----|---------------|-----------|--------|----------|-----------|
|    | <dbl>         | <dbl>     | <dbl>  | <dbl>    | <dbl>     |
| 1  | 5             | 1.58      | 1.34   | 1.51     | 1.59      |
| 2  | 6             | 1.82      | 1.58   | 1.81     | 1.82      |
| 3  | 3.4           | 1.06      | 0.951  | 0.861    | 0.939     |
| 4  | 2.7           | 0.5       | 0.782  | 0.518    | 0.411     |
| 5  | 10            | 2.24      | 2.54   | 2.26     | 2.29      |
| 6  | 9.7           | 2.39      | 2.47   | 2.27     | 2.26      |
| 7  | 9.55          | 2.29      | 2.43   | 2.27     | 2.25      |
| 8  | 3.05          | 0.558     | 0.866  | 0.694    | 0.705     |
| 9  | 8.15          | 2.17      | 2.10   | 2.20     | 2.13      |
| 10 | 6.2           | 1.87      | 1.63   | 1.86     | 1.86      |

```
# i 15 more rows
```

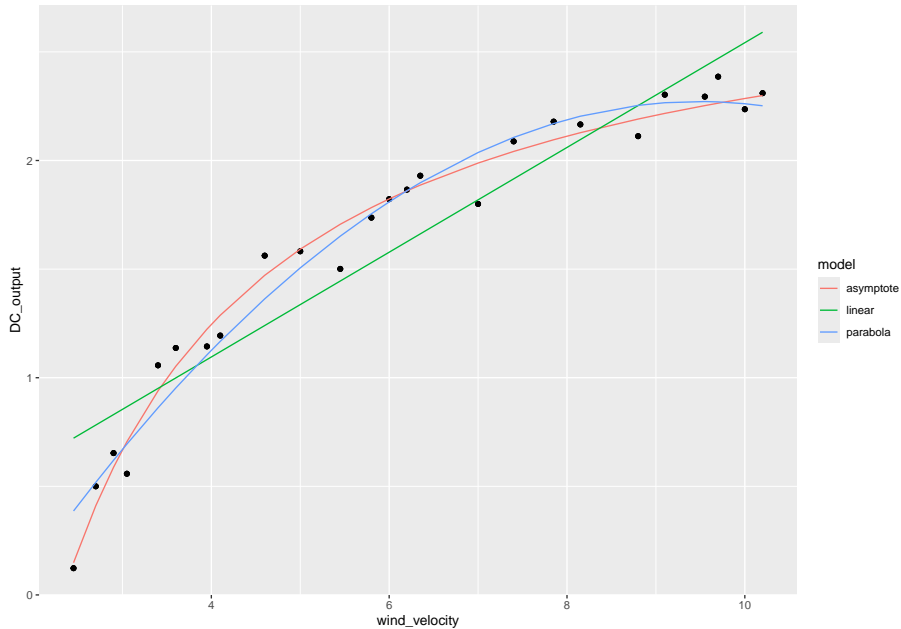


# Making the plot

- ggplot likes to have one column of  $x$ 's to plot, and one column of  $y$ 's, with another column for distinguishing things.
- But we have three columns of fitted values, that need to be combined into one.
- `pivot_longer`, then plot:

```
w2 %>%  
  pivot_longer(linear:asymptote, names_to="model",  
               values_to="fit") %>%  
  ggplot(aes(x = wind_velocity, y = DC_output)) +  
  geom_point() +  
  geom_line(aes(y = fit, colour = model))
```

# Scatterplot with fitted curves



# Comments

- Predictions from curves are very similar.
- Predictions from asymptote model as good, and from simpler model (one  $x$  not two), so prefer those.
- Go back to asymptote model summary.

# Asymptote model summary

```
tidy(DC.3)
```

```
# A tibble: 2 x 5
```

|   | term        | estimate | std.error | statistic | p.value  |
|---|-------------|----------|-----------|-----------|----------|
|   | <chr>       | <dbl>    | <dbl>     | <dbl>     | <dbl>    |
| 1 | (Intercept) | 2.98     | 0.0449    | 66.3      | 8.92e-28 |
| 2 | wind_pace   | -6.93    | 0.206     | -33.6     | 4.74e-21 |

# Comments

- Intercept in this model about 3.
- Intercept of asymptote model is the asymptote (upper limit of DC.output).
- Not close to asymptote yet.
- Therefore, from this model, wind could get stronger and would generate appreciably more electricity.
- This is extrapolation! Would like more data from times when wind.velocity higher.
- Slope  $-7$ . Why negative?
  - ▶ As wind.velocity increases, wind.pace goes down, and DC.output goes up. Check.
- Actual slope number hard to interpret.

# Checking back in with research questions

- Is there a relationship between wind speed and current generated?
  - ▶ Yes.
- If so, what kind of relationship is it?
  - ▶ One with an asymptote.
- Can we model the relationship, in such a way that we can do predictions?
  - ▶ Yes, see model DC.3 and plot of fitted curve.
- Good. Job done.

## Job done, kinda

- Just because the parabola model and asymptote model agree over the range of the data, doesn't necessarily mean they agree everywhere.
- Extend range of wind.velocity to 1 to 16 (steps of 0.5), and predict DC.output according to the two models:

```
wv <- seq(1, 16, 0.5)
wv
```

```
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0 10.5 11.0 11.5 12.0 12.5 13.0 13.5 14.0 14.5 15.0 15.5 16.0
```

- R has `predict`, which requires what to predict for, as data frame. The data frame has to contain values, with matching names, for all explanatory variables in `regression(s)`.

## Setting up data frame to predict from

- Linear model had just `wind_velocity`.
- Parabola model had that as well (squared one will be calculated)
- Asymptote model had just `wind_pace` (reciprocal of velocity).
- So create data frame called `wv_new` with those in:

```
wv_new <- tibble(wind_velocity = wv, wind_pace = 1 / wv)
```



wv\_new

wv\_new

```
# A tibble: 31 x 2
  wind_velocity wind_pace
      <dbl>      <dbl>
1           1         1
2          1.5       0.667
3           2         0.5
4          2.5       0.4
5           3       0.333
6          3.5       0.286
7           4       0.25
8          4.5       0.222
9           5         0.2
10         5.5       0.182
# i 21 more rows
```

## Doing predictions, one for each model

- Use same names as before:

```
linear <- predict(DC.1, wv_new)
parabola <- predict(DC.2, wv_new)
asymptote <- predict(DC.3, wv_new)
```

- Put it all into a data frame for plotting, along with original data:

```
my_fits <- tibble(
  wind_velocity = wv_new$wind_velocity,
  linear, parabola, asymptote
)
```

## my\_fits

```
my_fits
```

```
# A tibble: 31 x 4
```

|    | wind_velocity | linear | parabola | asymptote |
|----|---------------|--------|----------|-----------|
|    | <dbl>         | <dbl>  | <dbl>    | <dbl>     |
| 1  | 1             | 0.372  | -0.471   | -3.96     |
| 2  | 1.5           | 0.493  | -0.157   | -1.64     |
| 3  | 2             | 0.613  | 0.137    | -0.488    |
| 4  | 2.5           | 0.734  | 0.413    | 0.205     |
| 5  | 3             | 0.854  | 0.670    | 0.667     |
| 6  | 3.5           | 0.975  | 0.907    | 0.998     |
| 7  | 4             | 1.10   | 1.13     | 1.25      |
| 8  | 4.5           | 1.22   | 1.33     | 1.44      |
| 9  | 5             | 1.34   | 1.51     | 1.59      |
| 10 | 5.5           | 1.46   | 1.67     | 1.72      |

```
# i 21 more rows
```

## Making a plot 1/2

- To make a plot, we use the same trick as last time to get all three predictions on a plot with a legend (saving result to add to later):

```
my_fits %>%  
  pivot_longer(  
    linear:asymptote,  
    names_to="model",  
    values_to="fit"  
  ) %>%  
  ggplot(aes(  
    y = fit, x = wind_velocity,  
    colour = model  
  )) + geom_line() -> g
```

## Making a plot 2/2

- The observed wind velocities were in this range:

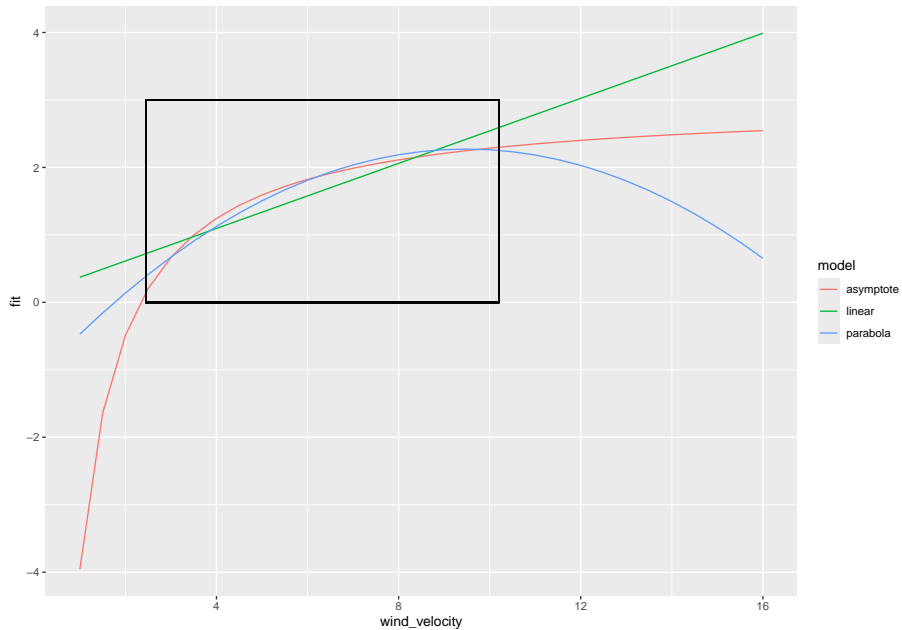
```
(vels <- range(windmill$wind_velocity))
```

```
[1] 2.45 10.20
```

- DC.output between 0 and 3 from asymptote model. Add rectangle to graph around where the data were:

```
g + geom_rect(  
  xmin = vels[1], xmax = vels[2], ymin = 0, ymax = 3,  
  alpha = 0, colour = "black"  
)
```

# The plot



## Comments (1)

- Over range of data, two models agree with each other well.
- Outside range of data, they disagree violently!
- For larger `wind.velocity`, asymptote model behaves reasonably, parabola model does not.
- What happens as `wind.velocity` goes to zero? Should find `DC.output` goes to zero as well. Does it?

## Comments (2)

- For parabola model:

```
tidy(DC.2)
```

```
# A tibble: 3 x 5
```

|   | term<br><chr>      | estimate<br><dbl> | std.error<br><dbl> | statistic<br><dbl> | p.value<br><dbl> |
|---|--------------------|-------------------|--------------------|--------------------|------------------|
| 1 | (Intercept)        | -1.16             | 0.175              | -6.62              | 1.18e- 6         |
| 2 | wind_velocity      | 0.723             | 0.0614             | 11.8               | 5.77e-11         |
| 3 | I(wind_velocity^2) | -0.0381           | 0.00480            | -7.95              | 6.59e- 8         |

- Nope, goes to  $-1.16$  (intercept), actually significantly different from zero.



## Comments (3): asymptote model

```
tidy(DC.3)
```

```
# A tibble: 2 x 5
  term          estimate std.error statistic  p.value
<chr>         <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    2.98      0.0449      66.3 8.92e-28
2 wind_pace     -6.93      0.206     -33.6 4.74e-21
```

- As `wind.velocity` heads to 0, `wind.pace` heads to  $+\infty$ , so `DC.output` heads to  $-\infty$ !
- Also need more data for small `wind.velocity` to understand relationship. (Is there a lower asymptote?)
- Best we can do now is to predict `DC.output` to be zero for small `wind.velocity`.
- Assumes a “threshold” wind velocity below which no electricity generated at all.

# Summary

- Often, in data analysis, there is no completely satisfactory conclusion, as here.
- Have to settle for model that works OK, with restrictions.
- Always something else you can try.
- At some point you have to say “I stop.”