

Modeling with Deterministic Functions - Capturing Signal

Zack Treisman

Spring 2021

Philosophy

Data analysis and statistics are tools used in modeling.

- ▶ A **model** is a proposed distribution of a variable or variables
- ▶ Separate any model into two parts: the **signal** and the **noise**.
- ▶ This week we are focused on **signal**.

The most fundamental setting is a pair of variables, x and y . We know something about x , and would like to leverage this to learn something about y , to the extent that this is possible. We call x the **predictor** and y the **response**. Write

$$y = f(x) + \epsilon$$

where the model function $f(x)$ is what we call the **signal** and ϵ is the **noise**.

“All models are wrong but some are useful” - George Box

The usefulness of a model comes when the signal is not drowned out by the noise. In other words the model is **statistically significant**.

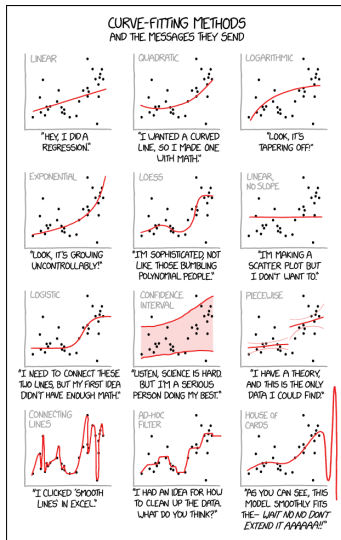


Figure 1: <https://xkcd.com/2048>

Model misuse is not a joke

United States Daily COVID-19 Deaths: Actual Data, IHME/UW Model Projections, & Cubic Fit.

Updated today (5/5/20), data through yesterday (5/4/20).

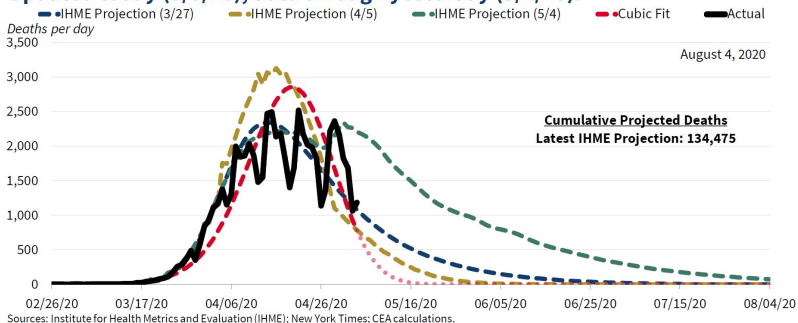


Figure 2:

<https://twitter.com/WhiteHouseCEA/status/1257680258364555264>

Reducible and irreducible error

If we had infinite knowledge, we could choose for our model function the expected value, or mean, of all y such that (x, y) is a possible data point. Write $\mu(x)$ for this expected value.

$$y = \mu(x) + \varepsilon$$

We call ε the **irreducible error** or **intrinsic variance**. It is the uncertainty that exists because of natural variation in the system described.

Any actual model function that we come up with will differ from this optimal function. Suppose we have a model function $f(x)$. We call the difference $f(x) - \mu(x)$ the **reducible error**. With a better f we can reduce the reducible error.

Bias and Variance

The reducible error can be broken down into two parts.

- ▶ The error due to **bias** is that part of the reducible error that comes from a model function's inability to change when it needs to.
- ▶ The error due to **variance** is that part of the reducible error that comes from a model function's excessive flexibility to match the particular data that are observed.

A model function with high bias error is said to **underfit** the data, and one with high variance error is an **overfit**. Whenever we are choosing a model, we must consider this **bias-variance tradeoff**.

Parametric vs Non-parametric models

- ▶ A **parametric** model function is one defined in terms of arithmetic and analytic functions, such as logarithms, polynomials, or anything else you might have encountered in a math class like Calculus. The numbers such as coefficients and exponents defining the function are called the **parameters**.
- ▶ A **non-parametric** model function is defined in some other way. The first example we will encounter and use is *local regression* or *loess*. Trees and random forests are also non-parametric models.

Linear function models

If x and y are both numeric variables, then the simplest possible relationship between them is a linear relationship.

$$y = \beta_0 + \beta_1 x + \epsilon$$

- ▶ β_0 is the *intercept*, the value we predict for y when x is zero.
- ▶ β_1 is the *slope*, or predicted rate of change in y with respect to x . Often written $\frac{\Delta y}{\Delta x}$ or $\frac{dy}{dx}$.

The tremendous advantage of the linear function model over all others is its simplicity. A disadvantage is a tendency towards high bias error.

Note: This model is linear in the variable x **and** in the *parameters* β_i . The term “linear model” is frequently applied with both meanings, but the latter is more useful. The R command `lm` refers to linearity in the parameters.

Linear models with multiple predictors

Extending to multiple predictor variables is straightforward.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon$$

- ▶ β_i is the expected change in y as x_i changes and all else is constant.
- ▶ If the x_i are correlated, these models can be unreliable.

Linear models with categorical predictors

If a predictor variable is categorical, the linear model can still be used.

- ▶ One level is set as the **reference** level of the variable.
- ▶ For every other level of the variable, an **indicator variable** is defined, taking the value 1 when the variable has that level and 0 otherwise.
- ▶ The coefficient β_i is the expected effect from observing level i instead of the reference level.

Examples:

- ▶ Control/ Treatment: Control is reference, x_{treat}
- ▶ Low/ Medium/ High: Low is reference, $x_{\text{med}}, x_{\text{high}}$

Polynomial functions

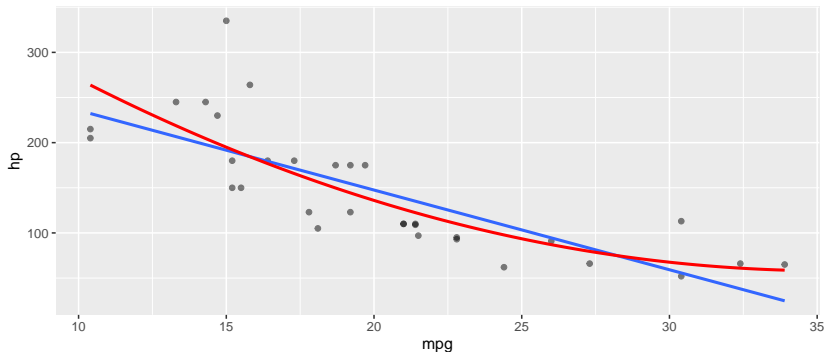
Including powers of x such as x^2 or x^3 can reduce bias.

Example:

$$y = \beta_0 + \beta_1x + \beta_2x^2 + \epsilon$$

This is still a linear model even though it includes the x^2 term.

```
ggplot(mtcars, aes(mpg, hp)) + geom_point(alpha=0.5) +  
  geom_smooth(method = lm, formula = y~x, se=F) +  
  geom_smooth(method = lm, formula = y~poly(x,2), se=F, color="red")
```



Power functions

A power function model is of the form

$$y = ax^k + \epsilon$$

Note that k can be any number, not just a positive integer. This is *not* a linear model because of the parameter k .

- ▶ Power models can have explanatory meaning if x and y have relevant dimensionality, like mass or area.

Power function models can be fit using linear model techniques by taking logarithms. Ignoring the error term for a moment:

$$\hat{y} = ax^k \quad \Longleftrightarrow \quad \log(\hat{y}) = \log(a) + k \log(x)$$

Back-transforms and error

Models fit on log-transformed variables can be exponentiated back to the original variables. How the error transforms can cause issues.

- ▶ Exponentiating the predicted mean of a log-transformed variable does **not** predict the untransformed mean.
 - ▶ When back-transforming, add half the variance in the residuals before exponentiating to recover the mean.
 - ▶ Diagnostics such as R^2 and p values apply to the transformed variables, not after back-transformation.
- ▶ Linear regression assumes that the error is additive. Exponentiation changes this addition into multiplication.

Suppose we fit a model:

$$\log(y) \sim N\left(\hat{\beta}_0 + \hat{\beta}_1 \log(x), \sigma^2\right).$$

Then the prediction for the mean of y is

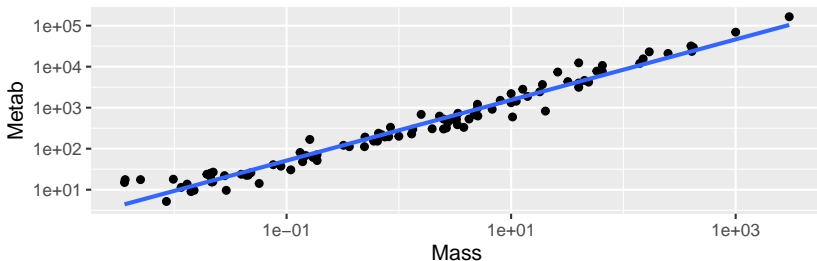
$$e^{\hat{\beta}_0 + \hat{\beta}_1 \log(x) + \sigma^2/2} = e^{\hat{\beta}_0 + \sigma^2/2} x^{\hat{\beta}_1}$$

and the variance is dependent on x .

Kleiber's law

Mass and metabolic rate of mammals relate via a power law.

```
ggplot(ex0826, aes(Mass, Metab)) + geom_point() + # data in Sleuth3  
  scale_x_log10() + scale_y_log10() + geom_smooth(method = lm, se=F)
```



```
lm1 <- lm(log(Metab)~log(Mass), data = ex0826)  
lm1$coefficients; var(lm1$residuals)
```

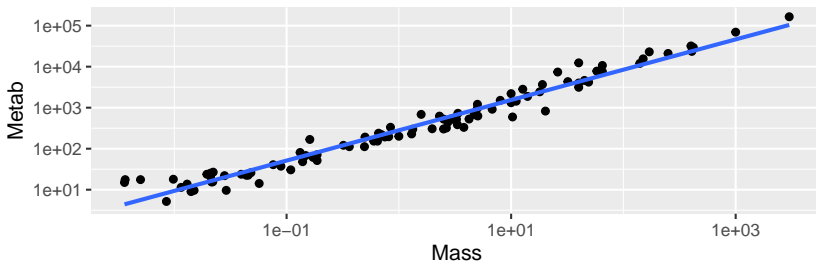
```
## (Intercept)    log(Mass)  
##    5.6383307    0.7387436  
  
## [1] 0.2068395
```

$$\text{Metab} = e^{5.64+0.21/2} \times (\text{Mass})^{0.74} \times \epsilon$$

Kleiber's law

Mass and metabolic rate of mammals relate via a power law.

```
ggplot(ex0826, aes(Mass, Metab)) + geom_point() + # data in Sleuth3  
  scale_x_log10() + scale_y_log10() + geom_smooth(method = lm, se=F)
```



```
lm1 <- lm(log(Metab)~log(Mass), data = ex0826)  
lm1$coefficients; var(lm1$residuals)
```

```
## (Intercept)    log(Mass)  
##    5.6383307    0.7387436  
  
## [1] 0.2068395
```

$$\text{Metab} = e^{5.64+0.21/2} \times (\text{Mass})^{0.74} \times \epsilon$$

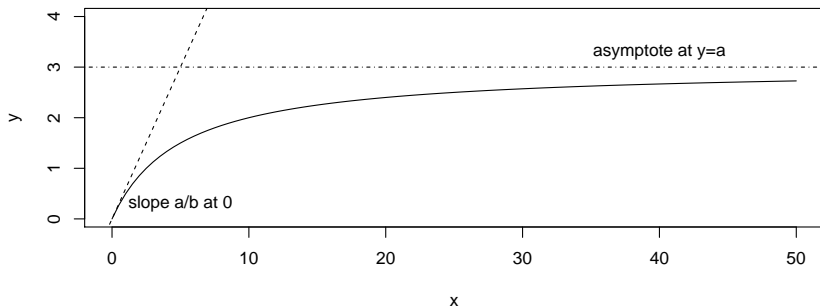
Rational models

The ratio of two polynomials is called a **rational function**. They can have asymptotes. Not generally linearizable.

Example: Michaelis-Menten/ Holling (McNickle and Brown (2014))

$$f(x) = \frac{ax}{b+x}$$

```
curve(3*x/(5+x), from = 0, to = 50, ylim = c(0, 4), ylab = "y")  
abline(h=3, lty = 4); abline(0, 3/5, lty = 2)  
text(5, 0.3, "slope a/b at 0"); text(40, 3.3, "asymptote at y=a")
```



Exponential models

Exponential growth and decay are very common.

- ▶ $\frac{dy}{dx}$ means change in y as x changes.
- ▶ $\frac{dy}{dx} = k \cdot y$ means y changes by a fixed fraction (k) of itself.

The solution is

$$y = a e^{kx} + \epsilon$$

Growth if $k > 0$, decline if $k < 0$.

- ▶ Exponential growth is usually bad for extrapolation. Something else tends to take over.
- ▶ With a little algebra, exponential decay can be used to model convergence to any asymptote.

These models are not linear, but taking logs make them so:

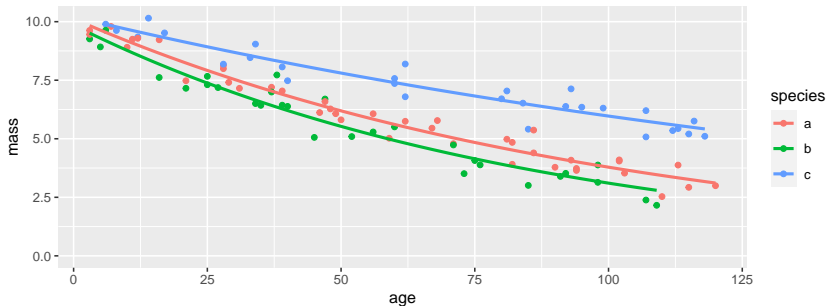
$$\hat{y} = a e^{kx} \quad \Longleftrightarrow \quad \log(\hat{y}) = \log(a) + kx$$

Exponential decay example

Exponential decay has long been used to model decomposition of biotic material. Olson (1963)

Simulated data, see RMarkdown for code.

```
ggplot(decay, aes(age, mass, color=species))+  
  expand_limits(y=0)+  
  geom_point()+  
  geom_smooth(method="glm",  
              method.args = list(family=gaussian(link="log")),  
              se=FALSE)
```



Decay model using log-transformed mass

In this model `speciesb` and `speciesc` are not significant at $\alpha = 0.05$ but the interaction terms are, so their initial quantities do not appear to differ from species a, but their decay rates do.

```
lm2 <- lm(log(mass)~age*species, data=decay)
round(summary(lm2)$coefficients,4)
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	2.3225	0.0283	81.9712	0.0000
## age	-0.0100	0.0004	-24.5589	0.0000
## speciesb	-0.0059	0.0434	-0.1363	0.8919
## speciesc	0.0019	0.0455	0.0410	0.9674
## age:speciesb	-0.0023	0.0007	-3.3980	0.0010
## age:speciesc	0.0046	0.0006	7.5162	0.0000

```
round(var(lm2$residuals),4)
```

```
## [1] 0.008
```

The model for the mean is:

$$\text{mass} = 10.242 \times e^{(-0.01 - 0.0023\chi_b + 0.0046\chi_c)\text{age}} \times \epsilon$$

$10.242 = e^{2.3225 + 0.008/2}$ and χ_b and χ_c are indicator functions.

Link functions

Most people who fit models to log transformed variables don't do the back-transformation. But this means that they aren't actually talking about the variables in the system they want to model.

Generalized linear models (GLMs) offer a solution: link functions. The mathematics of computing the parameters is more complicated, but they give a model that for untransformed variables.

A GLM with a log link for y in terms of x fits:

$$\log(\mu_y) = \beta_0 + \beta_1 x$$

or equivalently

$$\mu_y = e^{\beta_0 + \beta_1 x}$$

and the error is homoscedastic, as we like it.

Using GLMs and link functions means we can talk about our variables, not their transforms.

Decay model using glm and log link

Specifying a link function for a GLM in R is done in the family argument, which is used to describe the shape of the error. For normally distributed error, use family=gaussian.

```
lm2a <- glm(mass~age*species, family=gaussian(link=log), data=decay)
round(summary(lm2a)$coefficients, 4)
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	2.3150	0.0185	125.4186	0.0000
## age	-0.0098	0.0004	-25.8990	0.0000
## speciesb	-0.0280	0.0300	-0.9342	0.3526
## speciesc	0.0082	0.0278	0.2947	0.7689
## age:speciesb	-0.0017	0.0007	-2.5257	0.0132
## age:speciesc	0.0045	0.0005	8.9756	0.0000

Note the increase in the p-value for the age:speciesb term.

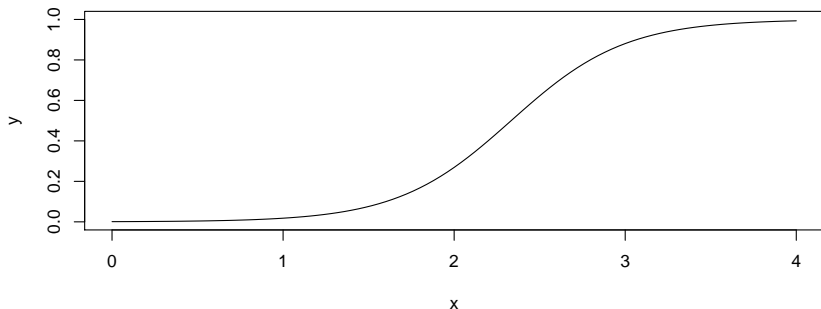
- It is possible that a difference between groups can appear significant for log transformed variables but not when we look at the variables directly.

Logistic models

The logistic function makes a transition from $y = 0$ to $y = 1$.

$$y = \frac{e^{a+bx}}{1 + e^{a+bx}}$$

```
curve(exp(-7+3*x)/(1+exp(-7+3*x)), from=0, to=4, ylim=c(0, 1), ylab="y")
```

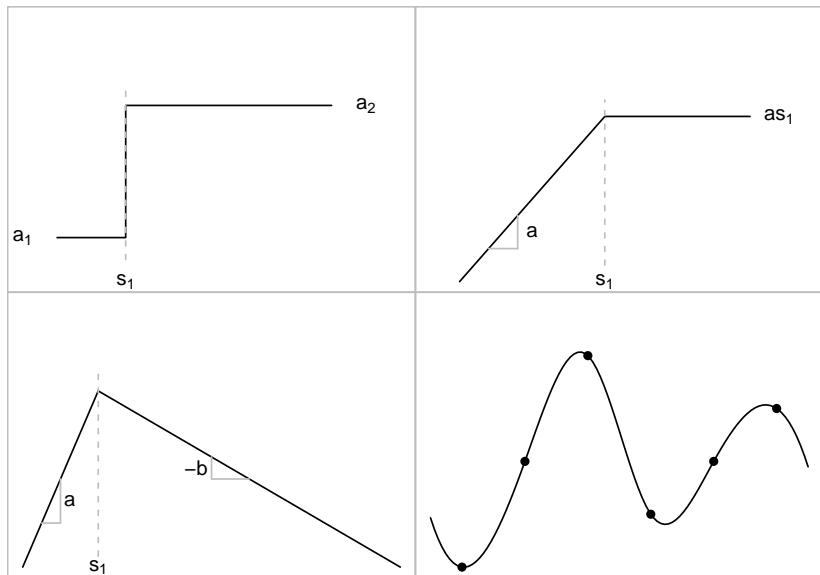


- ▶ Mostly used for binary classification. (logistic regression)
- ▶ Also useful for populations with a carrying capacity.

Model with `link=logit` in `glm`. Default for `family=binomial`.

Piecewise models

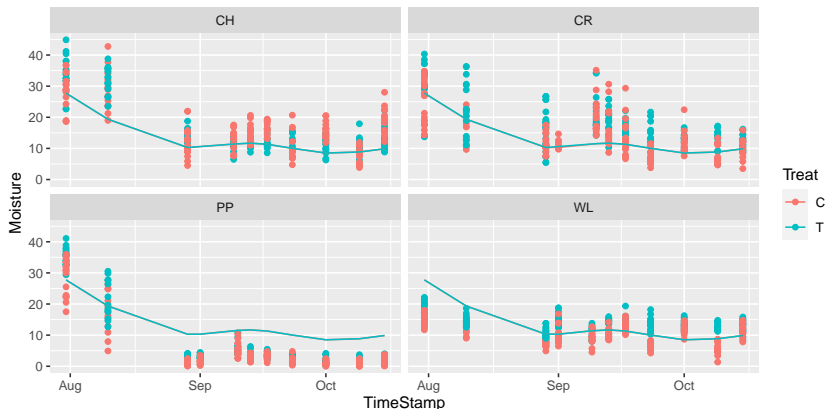
Bolker (2008) Figure 3.7.



Splines

Weather drives the response. A spline can help account for this.

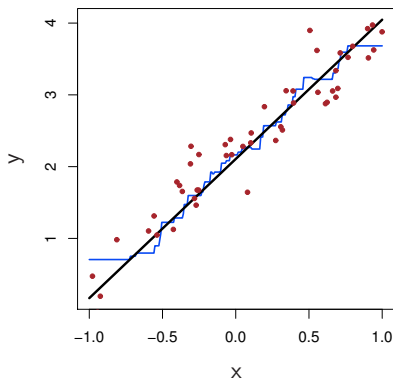
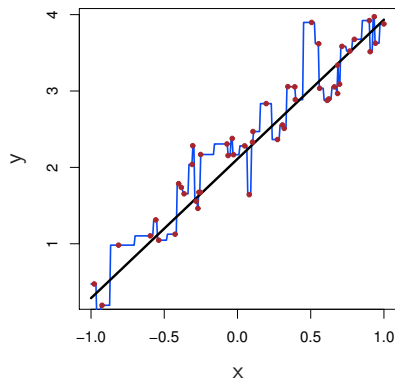
```
SoilMoist <- read.csv("data/SoilMoisture_ALL.csv") # Alexia Cooper's data
SoilMoist$TimeStamp <- mdy(SoilMoist$TimeStamp)
splineMod <- lm(Moisture~ns(TimeStamp,4), data=SoilMoist)
SoilMoist$pred <- predict(splineMod)
ggplot(SoilMoist, aes(TimeStamp,Moisture, color = Treat))+
  facet_wrap(~Site)+geom_point()+geom_line(aes(y=pred))
```



Nearest neighbor averaging (knn)

- Assume the actual expected value function doesn't change too quickly: $\mu(x - h) \approx \mu(x) \approx \mu(x + h)$.

Choose a positive integer k and define $f(x)$ as the average of y_i for the points (x_i, y_i) where $x - x_i$ is among the k smallest.

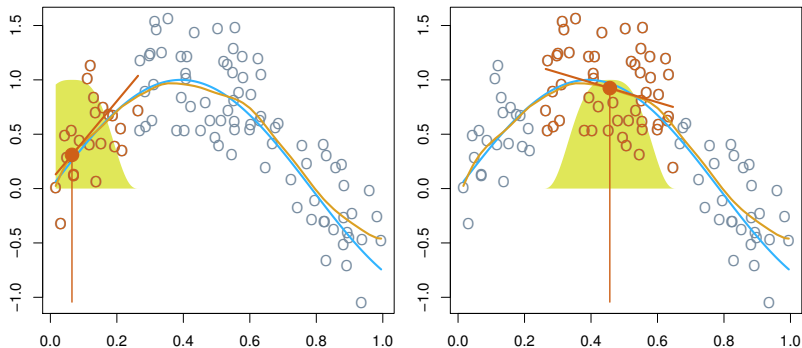


Left: knn with $k = 1$. Variance is high. Right: knn with $k = 9$.

Weighted averaging and local regression (loess)

Choose a distance and define $f(x)$ by linear regression using the data within that distance of x or weighted based on that distance.

Local Regression



Simulated data. Blue curve is the true signal, orange is the weighted local regression.

Loess in R

One way to do loess in R is to use `gam` and `lo`.

```
moistmod0 <- gam(Moisture~lo(TimeStamp), data=SoilMoist)
moistmod1 <- gam(Moisture~lo(TimeStamp)+Site, data=SoilMoist)
moistmod2 <- gam(Moisture~lo(TimeStamp)+Site+Treat, data=SoilMoist)
moistmod3 <- gam(Moisture~lo(TimeStamp)+Site*Treat, data=SoilMoist)
anova(moistmod0, moistmod1, moistmod2, moistmod3, test="F")
```

```
## Analysis of Deviance Table
```

```
##
```

```
## Model 1: Moisture ~ lo(TimeStamp)
```

```
## Model 2: Moisture ~ lo(TimeStamp) + Site
```

```
## Model 3: Moisture ~ lo(TimeStamp) + Site + Treat
```

```
## Model 4: Moisture ~ lo(TimeStamp) + Site * Treat
```

```
##   Resid. Df Resid. Dev Df Deviance      F      Pr(>F)
## 1      1362.8      56413
## 2      1359.8      38604  3  17809.1 225.919 < 2.2e-16 ***
## 3      1358.8      36577  1   2026.9  77.138 < 2.2e-16 ***
## 4      1355.8      35627  3    950.3  12.056 8.642e-08 ***
## ---
```

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

Using loess instead of a spline, we ask if Site, Treat, and their interaction are significant drivers of soil moisture after accounting for the common temporal variability. It appears that the answer is yes.

Acknowledgements

Some figures in this presentation are taken from “An Introduction to Statistical Learning, with applications in R” (Springer 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani

Other figures are created using code provided by Ben Bolker related to his text “Ecological Models and Data in R” (Princeton 2008)

References

Bolker, Benjamin M. 2008. *Ecological Models and Data in R*. Princeton University Press.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning: With Applications in R*. Springer.
<https://faculty.marshall.usc.edu/gareth-james/ISL/>.

McNickle, Gordon G., and Joel S. Brown. 2014. "When Michaelis and Menten met Holling: towards a mechanistic theory of plant nutrient foraging behaviour." *AoB PLANTS* 6 (December).
<https://doi.org/10.1093/aobpla/plu066>.

Olson, Jerry S. 1963. "Energy Storage and the Balance of Producers and Decomposers in Ecological Systems." *Ecology* 44 (2): 322–31. <https://doi.org/https://doi.org/10.2307/1932179>.