



# STOR 320 Modeling II&III

Lecture 16

Yao Li

Department of Statistics and Operations Research

UNC Chapel Hill



# Tutorial 11

- Instructions
  - Download Tutorial Zip
  - Unzip Folder
  - Required Packages
    - `library(tidyverse)`
    - `library(modelr)`
  - Open .Rmd File and Knit
- Daily Spanish River Data
  - W = Max Water Temperature
  - A = Max Air Temperature
  - L = River Identifier (31 Rivers)



# Introduction

- Questions About RMarkdown
  - What Does the Following Code Do When Knitted?

```
`r length(unique(DATA$L))`
```

- What Does the following Code Chunk Option Do When Knitted?

```
echo=F
```



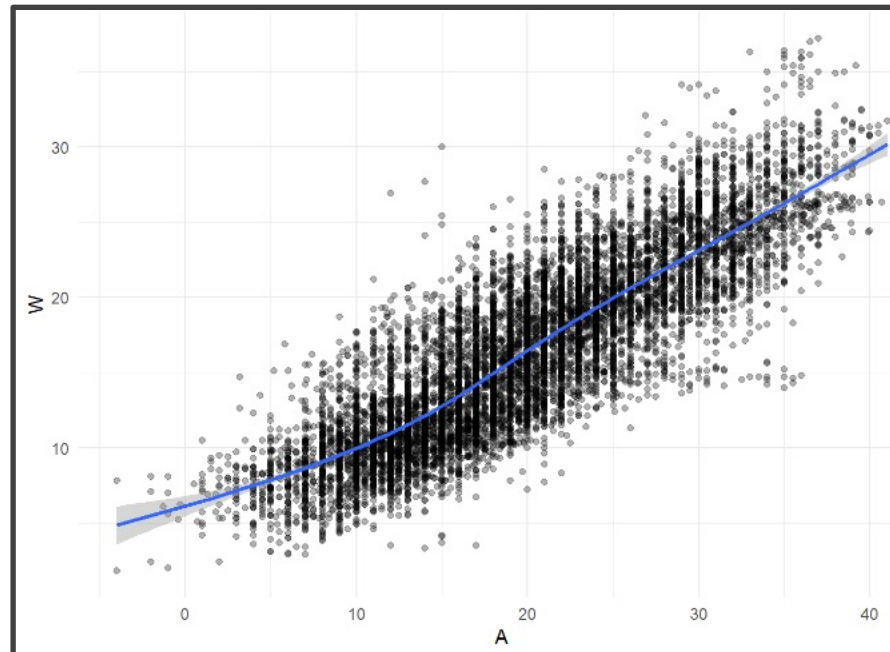
# Introduction

- Goal: Build a Model to Predict Max Water Temp Given Max Air Temp
  - What Do You Know About the Relationship of These Variables?
  - Who Would Care About this Relationship?
  - Why Would Someone Want to Predict the Max Water Temp?
  - Why Would this Model Be Useful?



# Part 1: Examining the Relationship

- Run Chunk 1
  - What Do You Notice About the Overall Relationship?

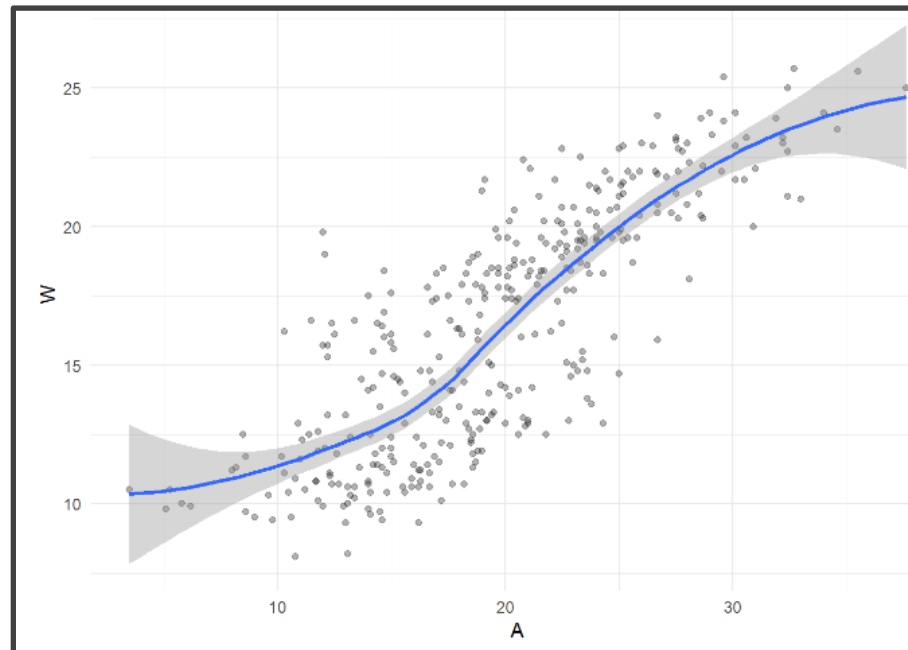


- Do You Think This Relationship is the Same for All Locations?
- Why? `message=F`



# Part 1: Examining the Relationship

- Run Chunk 2
  - Location is a Numeric Variable
  - What Do You Notice About the Relationship for  $L=103$ ?



- What do You Notice Now?



# Part 1: Examining the Relationship

- Chunk 2 Modified
  - Modify Chunk 2 to Create a Function Called `WAPlot.func` With 1 Argument Location
  - Function Usage: You Specify the Location as an Integer and the Function Outputs a Figure of the Relationship
  - Use Your Function For Three Different Locations
  - Knit the Document to Observe and Compare



# Part 1: Examining the Relationship

- Chunk 2 Discussion
  - What are the Differences in the Relationship Between W and A for the Various Locations?
  - Why do You Think These Differences Exist?
  - How do You Suggest We Handle the Differences?





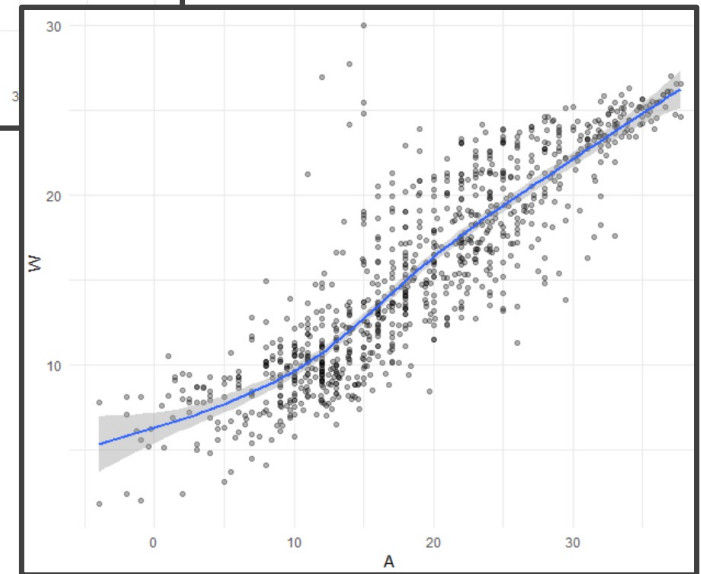
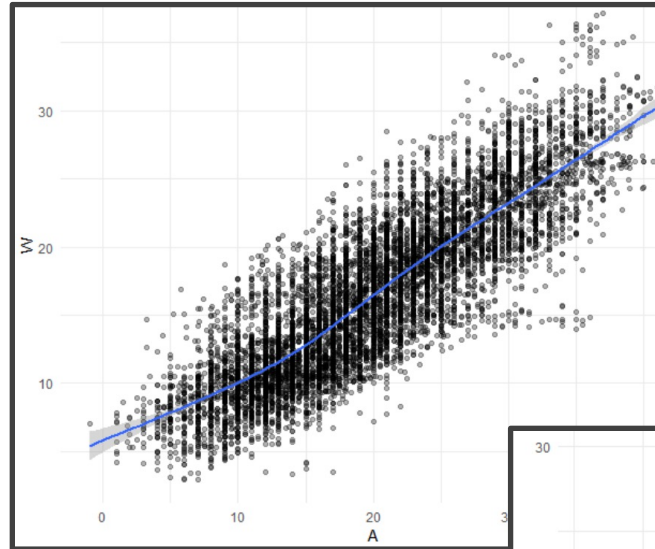
# Part 1: Examining the Relationship

- Chunk 3
  - Randomly Samples 3 Locations
  - Plant Your Seed and Run Code
  - Usage:
    - `anti_join()`
    - `semi_join()`
  - Why Don't We Handpick the Three Locations?



# Part 1: Examining the Relationship

- Run Chunk 4
  - Train Plot
  
- Test Plot





# Part 2: Linear Model

- Linear Model
  - Simplest Relationship that is Easily Explained
  - For every 1 Degree Change in  $A$ ,  $W$  changes by  $b$  Degrees
  - When  $A=0$  Degrees, the Expected Water Temperature is  $a$  Degrees



# Part 2: Linear Model

- Run Chunk 1
  - Fits Linear Model to Train Data
  - What is Your Intercept?
  - What is Your Slope?
- Run Chunk 2
  - Saves Predictions to Train/Test

```
add_predictions(MODEL,var="NAME")
```

- Run Chunk 3
  - Saves Residuals to Train/Test

```
add_residuals(MODEL,var="NAME")
```



# Part 3: Polynomial Model

- Polynomial Model
  - “Feature Engineering”
  - Generalized Additive Model
  - `Geom_smooth()` Fits a GAM when Fitting a Curve
  - Useful for Approximating Nonlinear Relationships
  - Dependent on Degree “k”
  - Goal: Choose Best “k”



# Part 3: Polynomial Model

- Formula Object in R
  - Special Notation
  - Helpful Table:

Symbol	Example	Meaning
+	+X	include this variable
-	-X	delete this variable
:	X:Z	include the interaction between these variables
*	X*Y	include these variables and the interactions between them
	X   Z	conditioning: include x given z
^	(X + Z + W) ^ 3	include these variables and all interactions up to three way
I	I (X*Z)	as is: include a new variable consisting of these variables multiplied
1	X - 1	intercept: delete the intercept (regress through the origin)

- We will Use the I() Function to Create New Variables Based Off Variables We Have



# Part 3: Polynomial Model

- Run Chunk 1
  - Fits 2<sup>nd</sup> Degree Polynomial
  - Fits 3<sup>rd</sup> Degree Polynomial
  - Fits 4<sup>th</sup> Degree Polynomial
- Run Chunk 2
  - Obtains Predictions Under the Different Polynomial Models

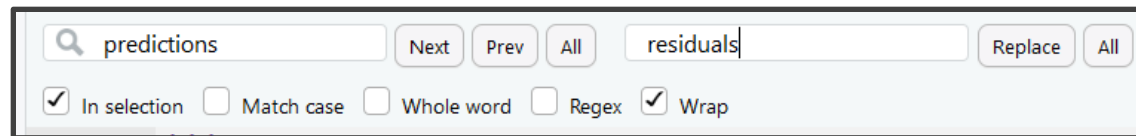


# Part 3: Polynomial Model

- Chunk 3
  - Code Needs Modification
  - Highlight Code

```
TRAIN4 =TRAIN3 %>%  
  add_predictions(poly2mod, var="poly2pred") %>%  
  add_predictions(poly3mod, var="poly3pred") %>%  
  add_predictions(poly4mod, var="poly4pred")  
  
TEST4 =TEST3 %>%  
  add_predictions(poly2mod, var="poly2pred") %>%  
  add_predictions(poly3mod, var="poly3pred") %>%  
  add_predictions(poly4mod, var="poly4pred")
```

- TRAIN3 -> TRAIN4 and etc.
- Use Ctrl+F (Find and Replace)
  - 'predictions' -> 'residuals'
  - 'pred' -> 'res'



- Run Chunk 3 After Modifying





# Intermission

- Run Code Chunk
  - `save.image()` = Used to Save Workspace into .Rdata File
  - `load()` = Used to Load Workspace from .Rdata File
  - .Rdata = File Extension of R Workspace File (All Objects in Global Environment)



# Tutorial 12

- Instructions
  - Download Tutorial Zip
  - Unzip Folder
  - Required Packages
    - `library(tidyverse)`
    - `library(modelr)`
  - Open .Rmd File and Knit
- Daily Spanish River Data
  - W = Max Water Temperature
  - A = Max Air Temperature
  - L = River Identifier (31 Rivers)



# Part 3: Polynomial Model

- Polynomial Model
  - “Feature Engineering”
  - Generalized Additive Model
  - `Geom_smooth()` Fits a GAM when Fitting a Curve
  - Useful for Approximating Nonlinear Relationships
  - Dependent on Degree “k”
  - Goal: Choose Best “k”



# Part 3: Polynomial Model

- Formula Object in R
  - Special Notation
  - Helpful Table:

Symbol	Example	Meaning
+	+X	include this variable
-	-X	delete this variable
:	X:Z	include the interaction between these variables
*	X*Y	include these variables and the interactions between them
	X   Z	conditioning: include x given z
^	(X + Z + W) ^ 3	include these variables and all interactions up to three way
I	I (X*Z)	as is: include a new variable consisting of these variables multiplied
1	X - 1	intercept: delete the intercept (regress through the origin)

- We will Use the I() Function to Create New Variables Based Off Variables We Have



# Part 3: Polynomial Model

- Run Chunk 1
  - Fits 2<sup>nd</sup> Degree Polynomial
  - Fits 3<sup>rd</sup> Degree Polynomial
  - Fits 4<sup>th</sup> Degree Polynomial
- Run Chunk 2
  - Obtains Predictions Under the Different Polynomial Models

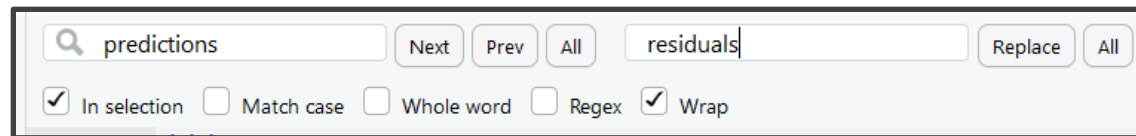


# Part 3: Polynomial Model

- Chunk 3
  - Code Needs Modification
  - Highlight Code

```
TRAIN4 =TRAIN3 %>%  
  add_predictions(poly2mod, var="poly2pred") %>%  
  add_predictions(poly3mod, var="poly3pred") %>%  
  add_predictions(poly4mod, var="poly4pred")  
  
TEST4 =TEST3 %>%  
  add_predictions(poly2mod, var="poly2pred") %>%  
  add_predictions(poly3mod, var="poly3pred") %>%  
  add_predictions(poly4mod, var="poly4pred")
```

- TRAIN3 -> TRAIN4 and etc.
- Use Ctrl+F (Find and Replace)
  - 'predictions' -> 'residuals'
  - 'pred' -> 'res'



- Run Chunk 3 After Modifying



# Intermission

- Run Code Chunk
  - `save.image()` = Used to Save Workspace into .Rdata File
  - `load()` = Used to Load Workspace from .Rdata File
  - .Rdata = File Extension of R Workspace File (All Objects in Global Environment)



# Part 4: Logistic Model

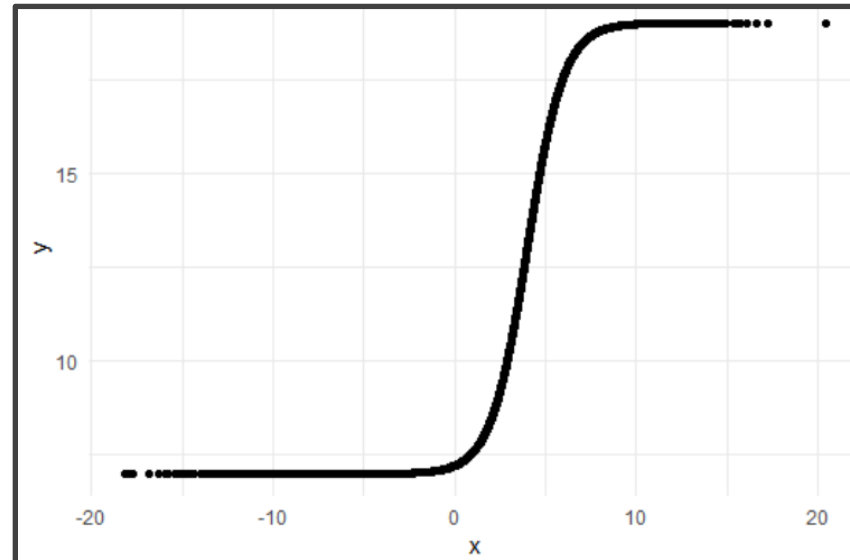
- Logistic Model
  - “Smart” Model Based On Physical Relationship Between  $A$  and  $W$
  - Four Parameters
    - Controls the Shape of the Relationship
    - $a$  and  $b$
    - $c$  and  $d$
  - What Shape Do You Think This Function Makes?
    - Idea: Precalculus





# Part 4: Logistic Model

- Run Chunk 1
  - Plant that Seed
  - Example Model



- Parameter Investigation
  - What Does 7 Represent?
  - What Does 12 Represent?
  - What Does 4 Represent?
  - What Does 1 Represent?



# Part 4: Logistic Model

- Run Chunk 2
  - Creation of Modeling Function
  - Creation of MSE Function Specific to this Model
- Run Chunk 3
  - Use `optim()` Function With Smart Starting Values Based on Understanding of The Model
  - Finds Estimates Based on Minimization of MSE



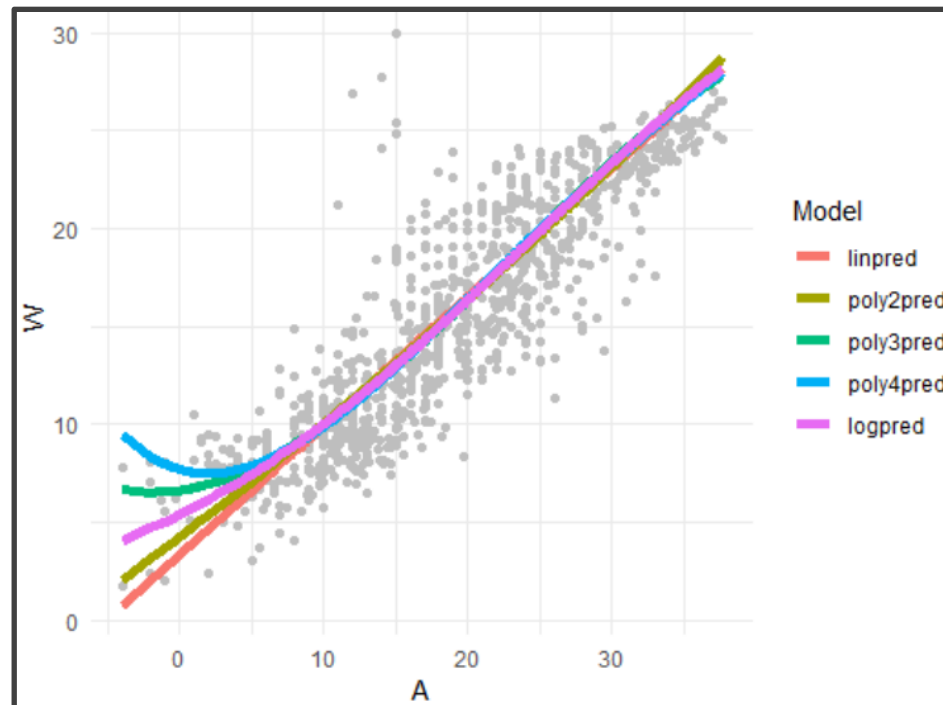
# Part 4: Logistic Model

- Run Chunk 4
  - Use Logistic Model Function and Estimated Parameters from `optim()` to Obtain
    - Predictions
    - Residuals



# Part 5: Evaluation by Visualization

- Run Chunk 1
  - Plots of Different Models
  - What Can We Say About the Different Models?

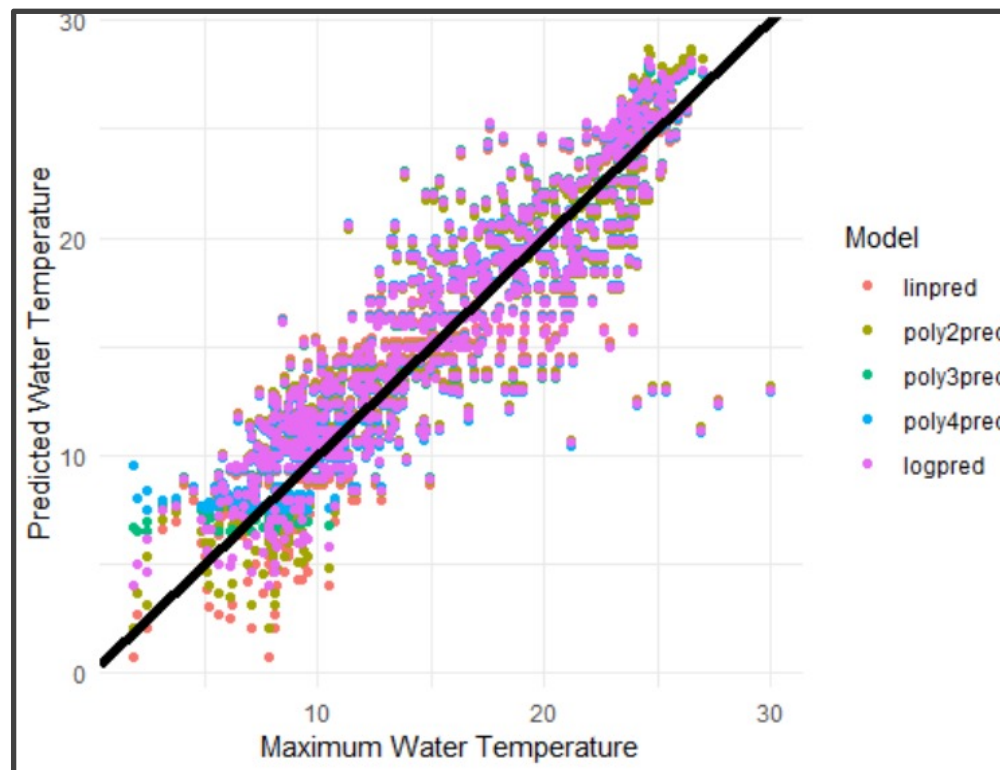


- Which Model Would You Use?



# Part 5: Evaluation by Visualization

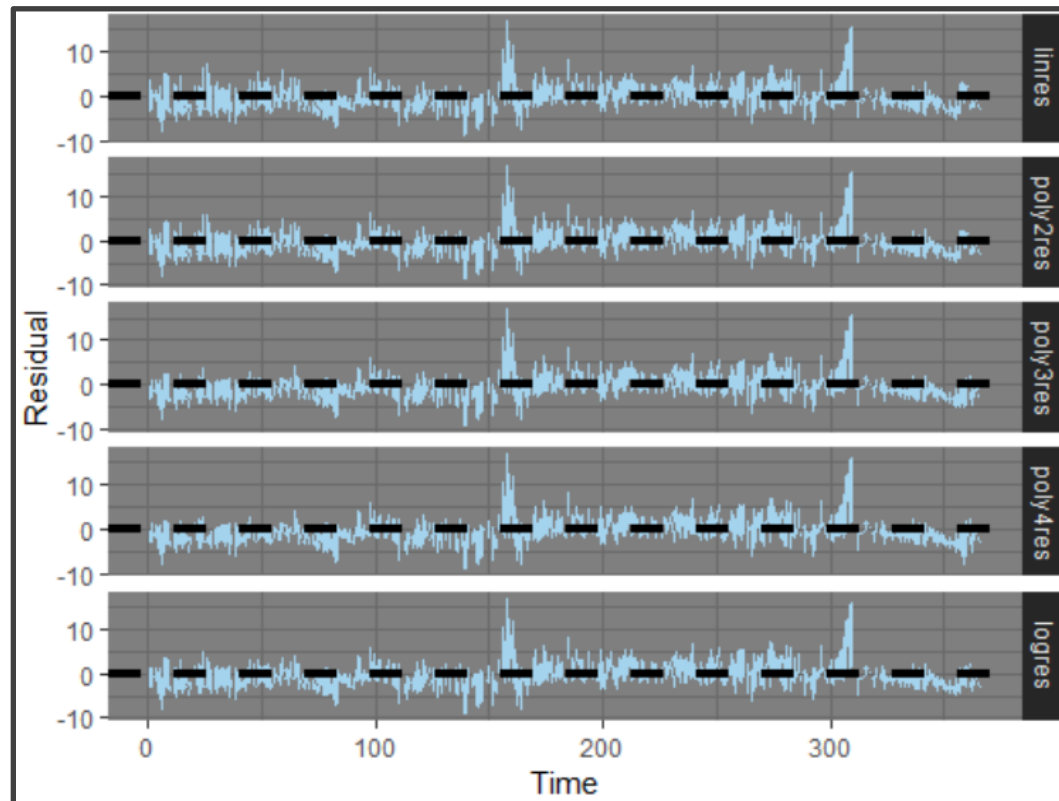
- Run Chunk 2
  - Comparing Predictions vs Actual Maximum Water Temperatures
  - Models Give Similar Predictions





# Part 5: Evaluation by Visualization

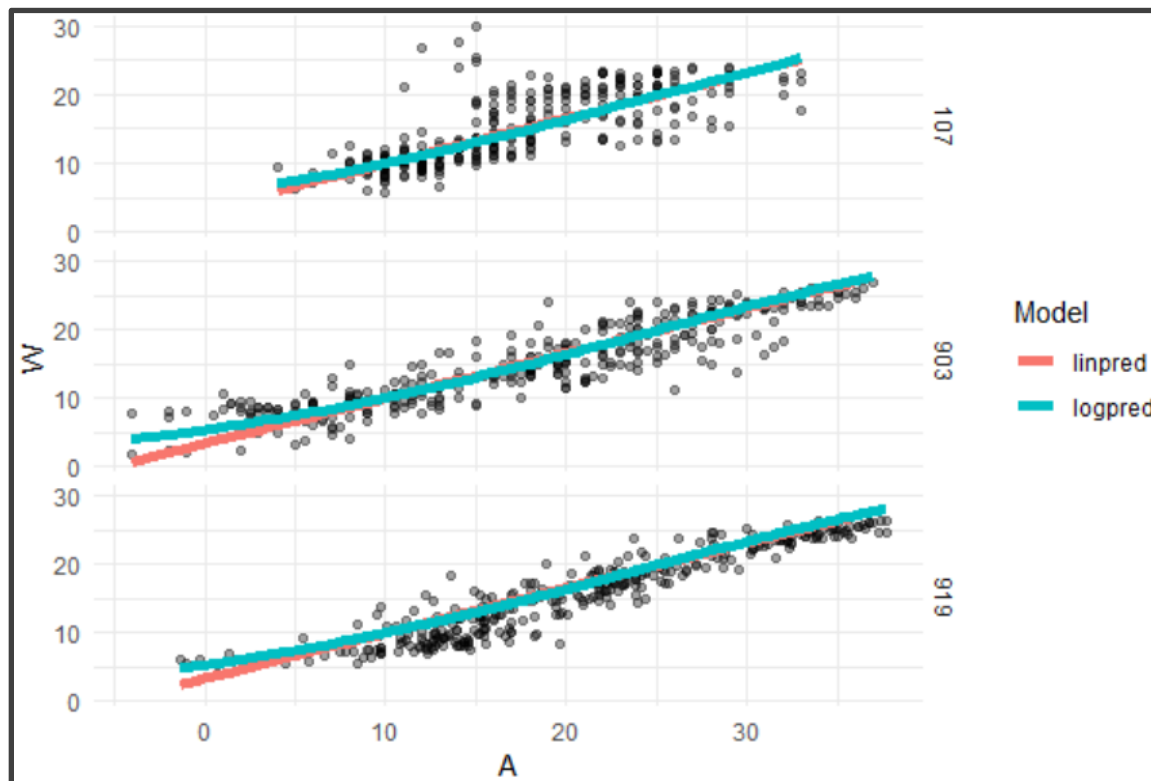
- Run Chunk 3
  - Shows Residuals Under the 4 Models Plotted Over Time
  - What is the Problem?





# Part 5: Evaluation by Visualization

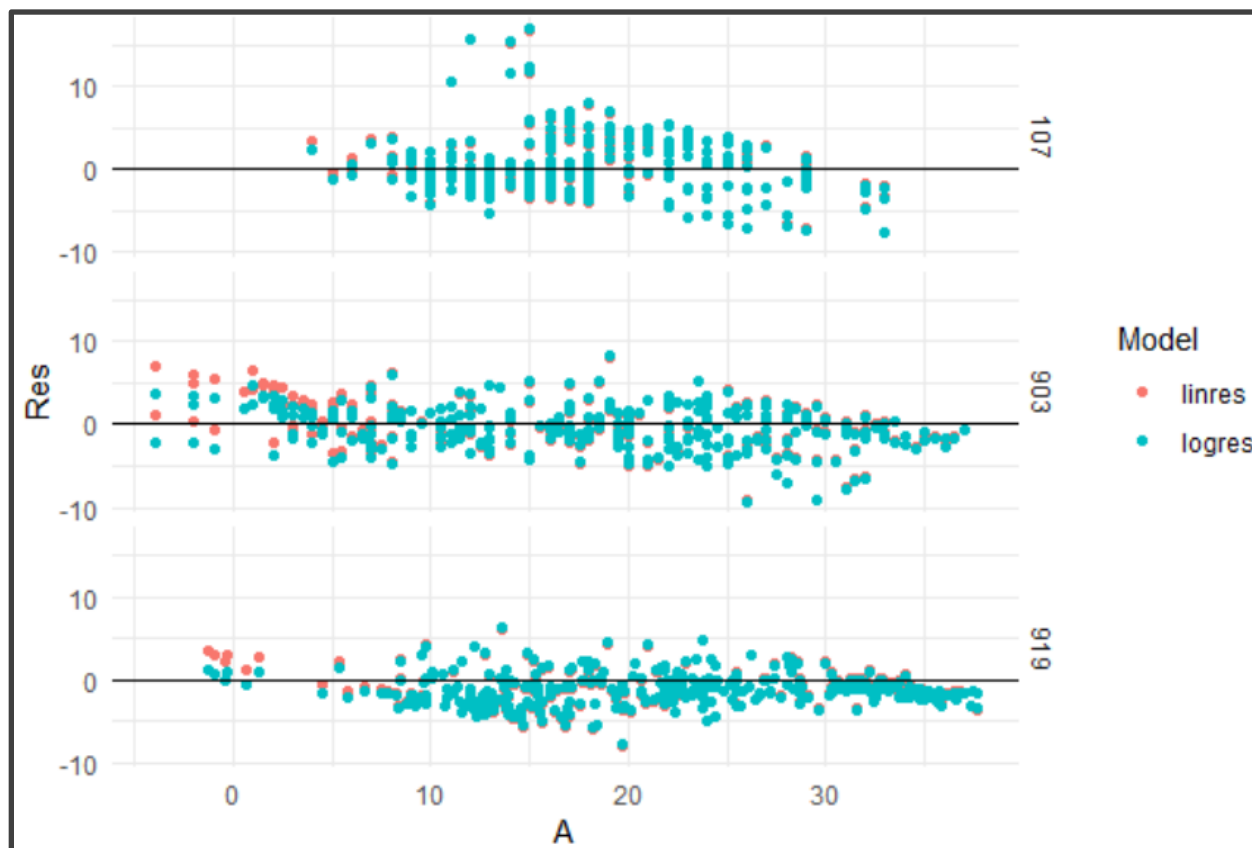
- Run Chunk 4
  - Evaluate Models For the Three Locations Separately





# Part 5: Evaluation by Visualization

- Run Chunk 5
  - Evaluate Error For the Three Locations Separately (by A)

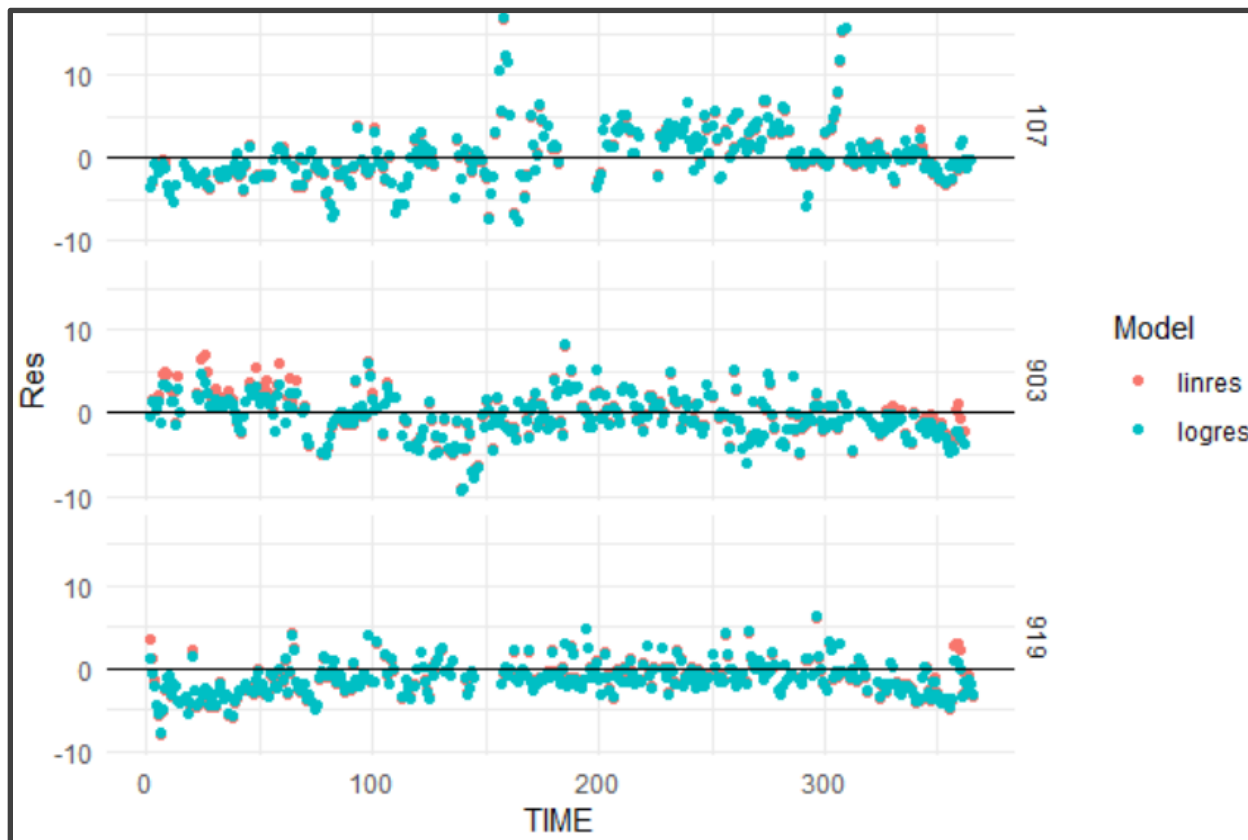






# Part 5: Evaluation by Visualization

- Run Chunk 6
  - Evaluate Error For the Three Locations Separately (by Time)





# Part 6: Evaluation by Numerical Summary

- Run Chunk 1
  - Mean Bias

$$\text{MB} = \frac{1}{N} \sum \hat{\epsilon}_k$$

- Mean Absolute Error

$$\text{MAE} = \frac{1}{N} \sum |\hat{\epsilon}_k|$$

- Root Mean Squared Error

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum \hat{\epsilon}_k^2}$$

- MB, MAE, and RMSE are in Degrees Celsius



# Part 6: Evaluation by Numerical Summary

- Summarizing Table
  - Evaluate MB, MAE, and RMSE on Test Data to Choose Best Model Going Forward
  - Sketch of Table We Want

Model	MB	MAE	RMSE
Linear			
Poly(2)			
Poly(3)			
Poly(4)			
Logistic			

- Before Writing Code, Have a Plan for the Output



# Part 6: Evaluation by Numerical Summary

- Chunk 2
  - Run Line-By-Line
  - Think About Ways to Quickly Apply All 3 Functions to All Residuals
- Run Chunk 3
  - Combine `rename()`, `gather()`, `group_by()`, and `summarize()`
- Chunk 4
  - Change `eval=F` to `eval=T` and Knit the File (What is Seen?)



# Part 6: Evaluation by Numerical Summary

- My Results Based on My Seed

<b>Model</b> <fctr>	<b>MB</b> <dbl>	<b>MAE</b> <dbl>	<b>RMSE</b> <dbl>
Linear	0.9534126	2.750323	3.351594
Poly(2)	0.9742415	2.732399	3.344867
Poly(3)	0.9903951	2.706833	3.328889
Poly(4)	0.9920042	2.715366	3.338710
Logistic	0.2613184	3.135313	3.711664

- When Results Are This Close, Always Consider the Most Simple Model