SIT718 Real World Analytics

Lecturer: Dr Ye Zhu

School of Information Technology
Deakin University

Week 6: Fitting aggregation functions to data

Fitting aggregation functions to data

For this week we have the following learning aims

- Be able to apply data fitting code in order to define parameters and interpret datasets
- To be able to make reasonable assessments of goodness of fit and accuracy of models

Read Chapter 5 of the reference book (**An Introduction to Data Analysis using Aggregation Functions in R** by Simon James)

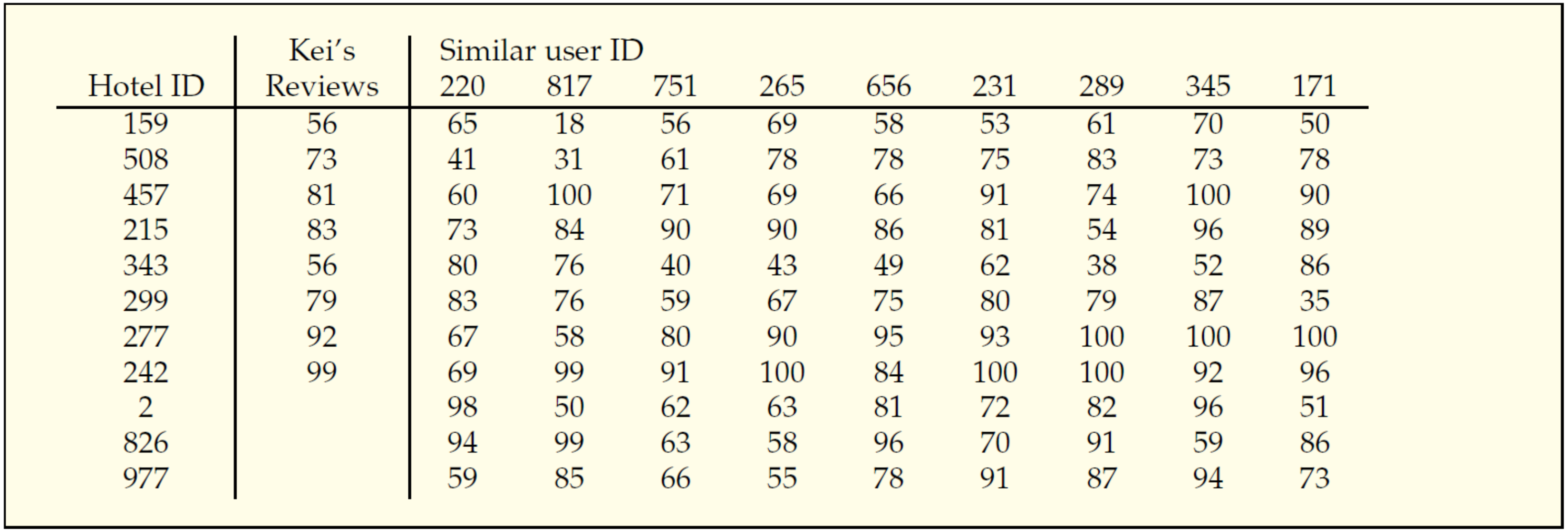
What can we do in R?

Throughout these last few weeks we have already learnt to do a lot with R and aggregation functions. You should be able to:

- Use aggregation functions defined with respect to weighting vectors to calculate the output of one or multiple input vectors
- For a given dataset determine by comparison whether one function (or set of weights) fits better than another

PREDICTING THE OUTPUTS FOR UNKNOWN/NEW DATA

Example: Collaborative recommenders Kei's Hotel Ratings
"based on similar users, we believe you will enjoy staying at Hotel X".



Hotel ID	Kei's Reviews	Similar user ID 220	817	751	265	656	231	289	345	171
159	56	65	18	56	69	58	53	61	70	50
508	73	41	31	61	78	78	75	83	73	78
457	81	60	100	71	69	66	91	74	100	90
215	83	73	84	90	90	86	81	54	96	89
343	56	80	76	40	43	49	62	38	52	86
299	79	83	76	59	67	75	80	79	87	35
277	92	67	58	80	90	95	93	100	100	100
242	99	69	99	91	100	84	100	100	92	96
2		98	50	62	63	81	72	82	96	51
826		94	99	63	58	96	70	91	59	86
977		59	85	66	55	78	91	87	94	73

We can define weight in terms of Distance (Generalised distance measure)

$$d(kei, j) = \left(\sum_{i=1}^{n} |x_{i,kei} - x_{i,j}|^p \right)^{1/p}$$

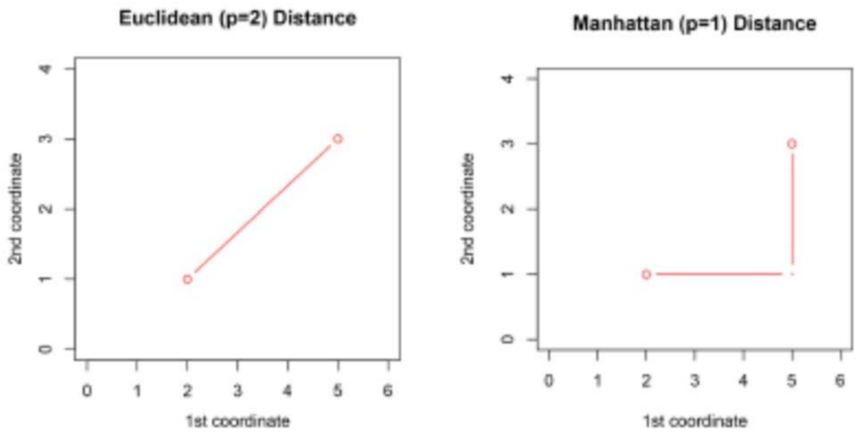
For example, the distance between Kei and user 220 when $p = 2$ would be

$$|56 - 65|^2 + |73 - 41|^2 + |81 - 60|^2 + ... + |99 - 69|^2 \, ^{1/2} .$$

Minkowski distance

$$d(kei, j) = \left(\sum_{i=1}^{n} |x_{i,kei} - x_{i,j}|^p \right)^{1/p}$$

- The above generalized distance with parameter p is called the Minkowski distance.
- When $p = 1$, this distance is called the Manhattan distance.
- When $p = 2$, this distance is called the Euclidean distance.



Another popular measure of similarity is known as cosine similarity (from vector arithmetic) and is given by,

$$\cos(kei, j) = \frac{\sum_{i}^{n} x_{i,kei} \cdot x_{i,j}}{\left(\left(\sum_{i}^{n} x_{i,kei}^2\right) \cdot \left(\sum_{i}^{n} x_{i,j}^2\right)\right)^{1/2}}$$

FITTING AGGREGATION FUNCTIONS

Just like with regression in statistics, we can find the best fitting parameters for an aggregation function. The idea is usually to minimise the sum of differences between predicted and observed values, with respect to the possible choices of weighting vector **w**.

Formula

$$\text{minimize}_{\mathbf{w}} \sum_{j=1}^{D} (A(\mathbf{x}_j) - y_j)^2$$

where $j = 1, 2, \ldots, D$ are the data points that we have observations for, y_j is our observed output, $\mathbf{x}_j$ is the input vector associated with that output and $A(\mathbf{x}_j)$ is our predicted value.

FITTING AGGREGATION FUNCTIONS

Usually, however it's a slightly harder problem than in the usual statistical approach because we have restrictions on our weights.

$$w_i \geq 0, \text{ for all } i$$

$$\sum_{i=1}^{n} w_i = 1$$

Fitting aggregation functions

We can also minimize least absolute deviation.

Formula

$$\text{minimize} \sum_{i=1}^{m} |A(\mathbf{x}_i) - y_i|$$

Fitting aggregation functions

We can essentially use this process for one of two purposes:

- Predicting the outputs for unknown/new data
- Making inferences about the ‘importance’ of each variable based on the fitted weighting vector

Predicting the outputs for unknown/new data

Example

1. *We have a data set such as:*

x_1	x_2	x_3	x_4	y
3	4	2.4	8	7.2
6.7	3	4.5	6	6.1
4.3	7	8.2	4.5	8
...				
2	3.2	1.9	8.1	?
4.5	7.1	7.2	6.3	?

2. *We use our fitting procedure (after performing any necessary data transformations and choosing which mean we wish to use) to find that* $\mathbf{w} = \langle 0.3, 0.2, 0.11, 0.39 \rangle$
3. *We then use our* **w** *along with the function to find the ? values.*

Making inferences about variables

Example

1. *We have a data set such as:*

x_1	x_2	x_3	x_4	y
3	4	2.4	8	7.2
6.7	3	4.5	6	6.1
4.3	7	8.2	4.5	8
...				

2. *We use our fitting procedure (after performing any necessary data transformations and choosing which mean we wish to use) to find that* $\mathbf{w} = \langle 0.3, 0.2, 0.11, 0.39 \rangle$

3. *From this we interpret that variables* 1 *and* 4 *seem more influential in determining the y-values.*

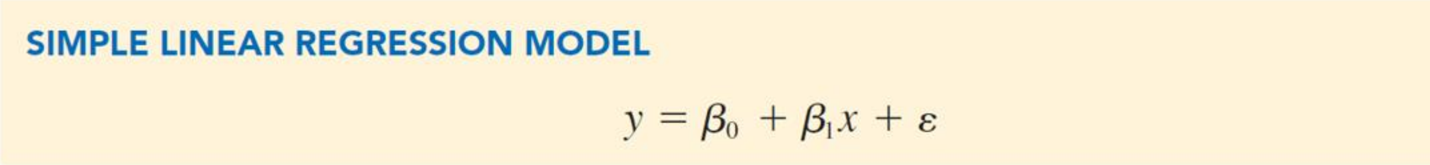
LINEAR REGRESSION

If data can be obtained, a statistical procedure called **regression analysis** can be used to develop an equation showing how the variables are related.

- **Dependent variable** or response: Variable being predicted.
- **Independent variables** or predictor variables: Variables being used to predict the value of the dependent variable.
- **Simple linear regression**: A regression analysis for which any one unit change in the independent variable, x, is assumed to result in the same change in the dependent variable, y.
- **Multiple linear regression**: A regression analysis involving two or more independent variables.

Simple Linear Regression Model

Regression Model:
The equation that describes how y is related to x and an error term.



SIMPLE LINEAR REGRESSION MODEL

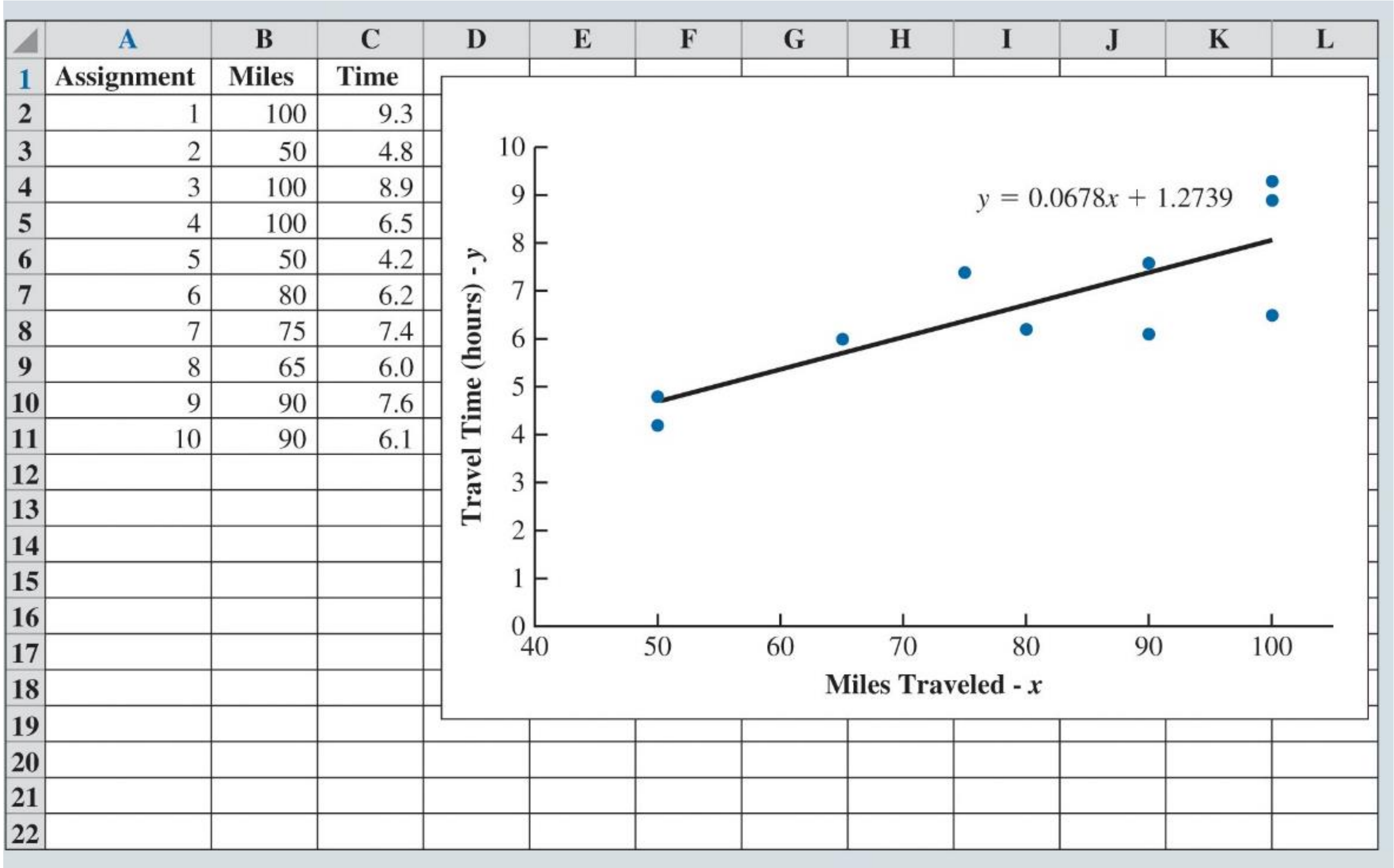
$$y = \beta_0 + \beta_1 x + \varepsilon$$

Parameters: The characteristics of the population, β_0 and β_1.

Random variable: Error term, ε.

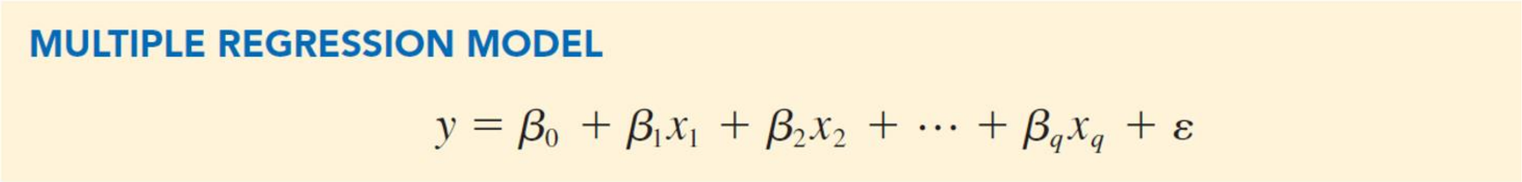
The error term accounts for the variability in y that cannot be explained by the linear relationship between x and y.

Simple Linear Regression Model (cont.)



	A	B	C
1	Assignment	Miles	Time
2	1	100	9.3
3	2	50	4.8
4	3	100	8.9
5	4	100	6.5
6	5	50	4.2
7	6	80	6.2
8	7	75	7.4
9	8	65	6.0
10	9	90	7.6
11	10	90	6.1

Multiple Regression Model



MULTIPLE REGRESSION MODEL

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q + \varepsilon$$

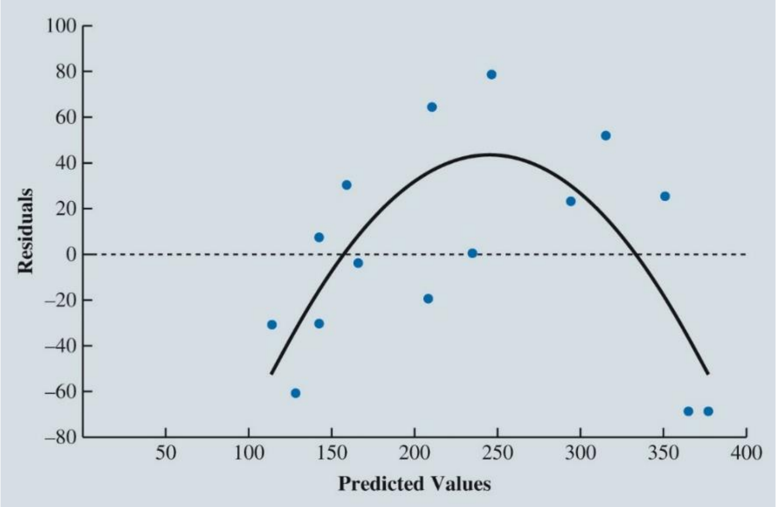
y = dependent variable.
$x_1, x_2, . . ., x_q$ = independent variables.
$\beta_0, \beta_1, \beta_2, \ldots, \beta_q$ = parameters.
ε = error term (accounts for the variability in y that cannot be explained by the linear effect of the q independent variables).

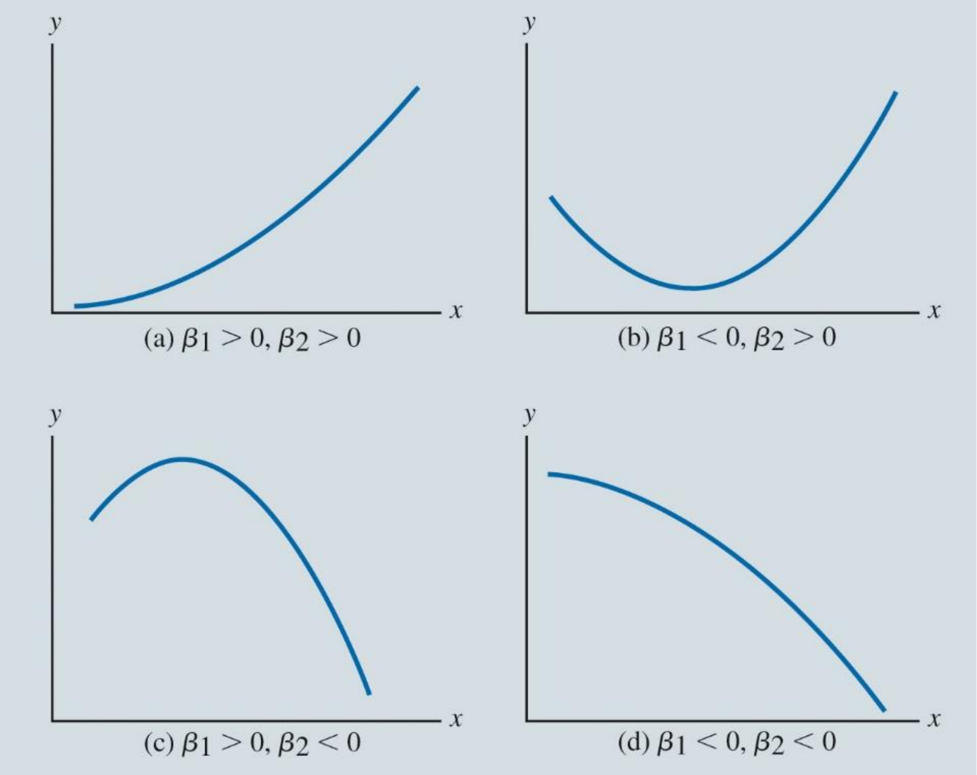
Modelling Nonlinear Relationships

A quadratic regression model

$$y = b_0 + b_1x_1 + b_2x_1^2$$

Relationships That Can Be Fit with a Quadratic Regression Model





(a) $\beta 1 > 0, \beta 2 > 0$

(b) $\beta 1 < 0, \beta 2 > 0$

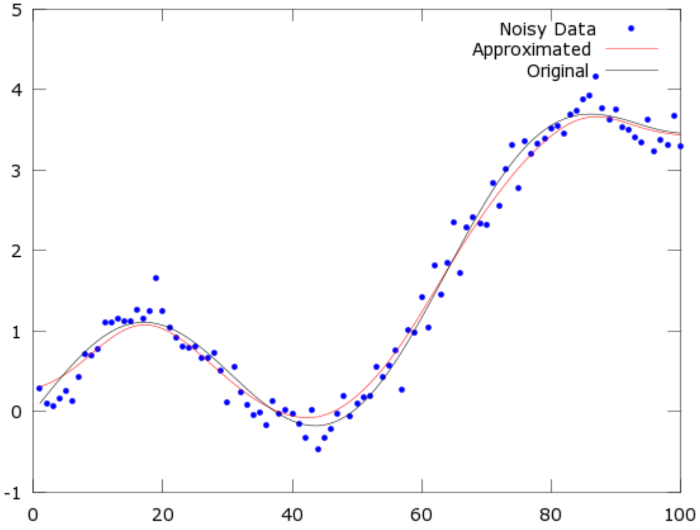
(c) $\beta 1 > 0, \beta 2 < 0$

(d) $\beta 1 < 0, \beta 2 < 0$

Modelling Nonlinear Relationships (cont.)

- Interaction Between Independent Variables:
  - **Interaction**: This occurs when the relationship between the dependent variable and one independent variable is different at various values of a second independent variable.
  - The estimated multiple linear regression equation is given as:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_1x_2$$

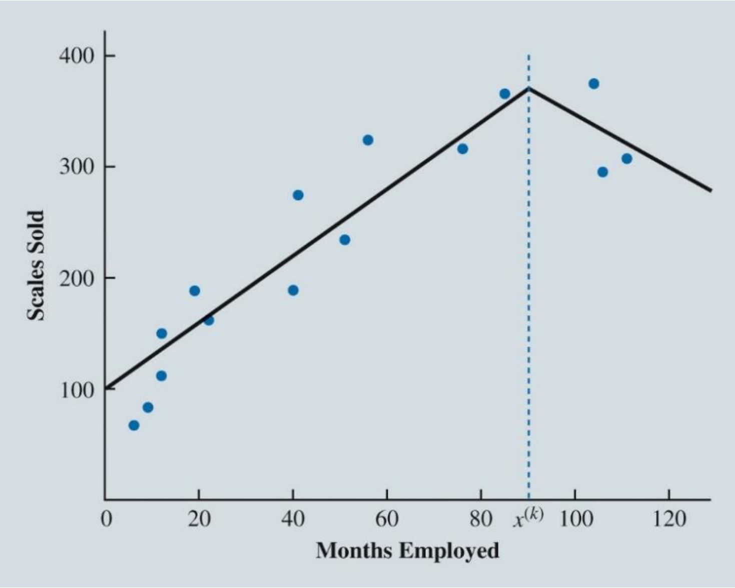


Modelling Nonlinear Relationships (cont.)

- **Piecewise linear regression model**: This model will allow us to fit these relationships as two linear regressions that are joined at the value of Months at which the relationship between Months Employed and Sales changes.
- **Knot**: The value of the independent variable at which the relationship between dependent variable and independent variable changes; also called *breakpoint*.

$$x_k = \begin{cases} 0 \text{ if } x_1 \leq x^{(k)} \\ 1 \text{ if } x_1 > x^{(k)} \end{cases}$$

$$y = b_0 + b_1 x_1 + b_2 (x_1 - x^{(k)}) x_k$$



Reliability

If we want to be able to make inferences or estimate the value of new data, we need to make sure our model is reliable. In statistics there is the notion of hypothesis testing, which is based on the probability of obtaining models with prior assumptions. Here we will consider more general evaluations:

- Are we evaluating an approach, or just the parameters of the model?
- Do we have a 'ground truth' dataset we can use?
- Do we have enough data to make reliable models?
- How can we account for overfitting?

We don't always have good conditions for being certain about our model, but we have to make sure any arguments we make can be justified both by the data and theoretically.

ARE WE EVALUATING AN APPROACH, OR JUST THE PARAMETERS OF THE MODEL?

We might be looking at an overall approach to prediction or parameter estimation. In this case, we want to be able to isolate, as much as possible, the influence of our contributions.

Example

Suppose we have a website that evaluates relevance of a website to a search query by aggregating the last 2 weeks of visits. Since website hits usually follow an exponential distribution, it has been standard practice in your company to perform a log transform of the data. You want to see if doing a piecewise linear transform of the data can achieve better results. How could you

determine/show that your new approach is better?

ARE WE EVALUATING AN APPROACH, OR JUST THE PARAMETERS OF THE MODEL? (cont.)

Considerations

- Two approaches? You can compare log transform and linear data transform, but it is also worth seeing what happens with raw data - so compare all three.
- How can you objectively say one approach is better? You need data that shows some kind of ground truth relationship between website hits and relevance for a number of cases.

day_1	day_2	day_3	...	relevance
3	4	24	8	72
67	3	45	6	61
43	7	82	45	8
...				

- Then we need to be able to perform a sufficient number of experiments on varied data to show that our approach is better

DO WE HAVE A 'GROUND TRUTH' DATASET WE CAN USE?

There are basically two types of datasets we can use for performing experiments: synthetic and real. Both have their own drawbacks.

- Real
  - Difficult to obtain (permission, privacy, availability, existence)
  - Can be incomplete
  - Can involve variables that are difficult to transform to numerical data
  - Can be too small or specific
- Synthetic
  - Need to be generated in a theoretically sound way appropriate to the problem
  - Always carry some skepticism - what bias and distributions exist in the real dataset that would not be captured by our generated one?

Do we have a 'ground truth' dataset we can use?

Real data

- Ensure the integrity of the dataset (keep good records)
- Make sure we understand the units
- Have experiments been carried out elsewhere on the data that we can replicate?

DO WE HAVE A 'GROUND TRUTH' DATASET WE CAN USE?

Synthetically generated data

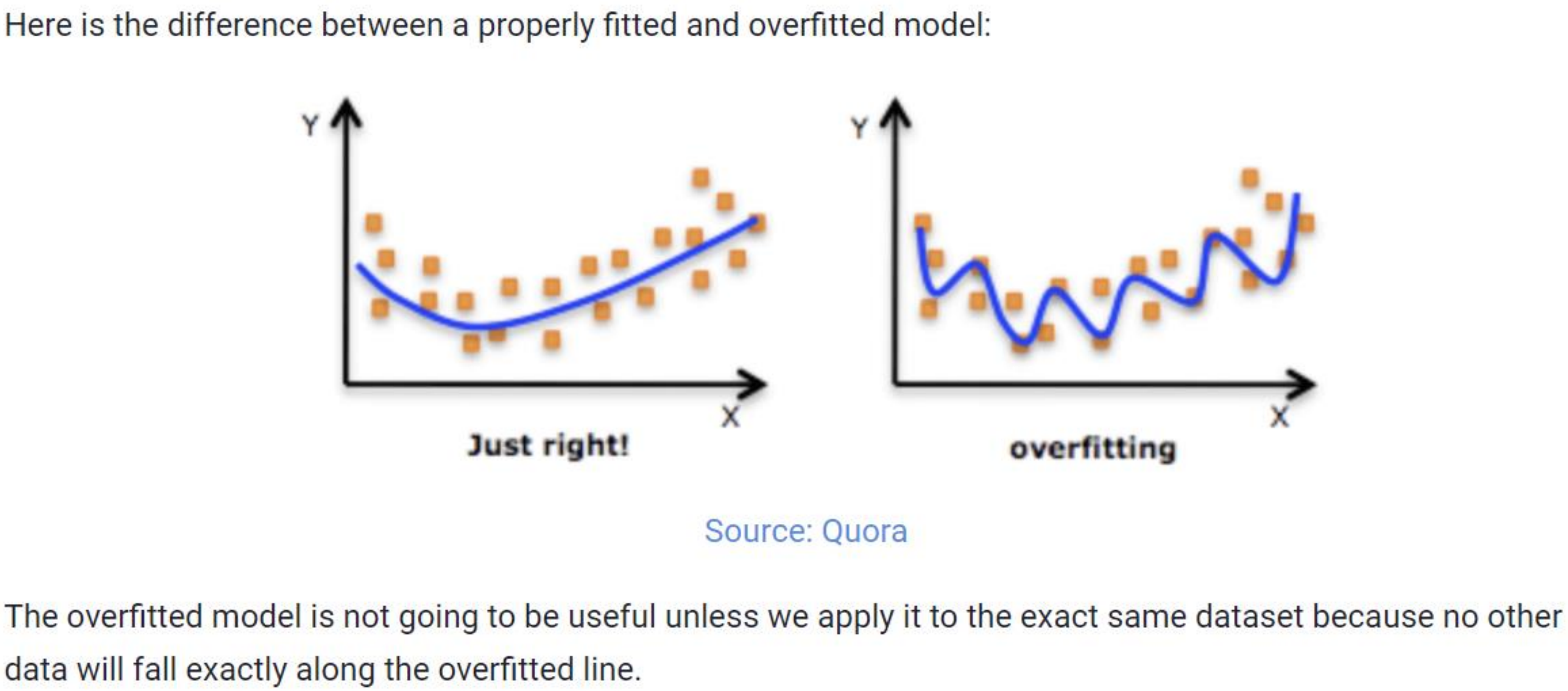
- Should we model randomness using a uniform distribution? normal distribution? exponential distribution? or some specialised way?
- Can we assume an underlying model and add noise? The type of noise we have, or the generation of the model, can often influence which functions will be robust in the first place (e.g. normally distributed noise with independent sources will often be aggregated best using the mean)

Do we have enough data to make reliable models?

- In general, you would need at least as many data points as you have parameters - this is the bare minimum! Ideally you'd want much more. Remember with the Choquet integral, we actually have 2^n parameters for n variables.
- Often more data will contribute to longer time to process and fit the data. Too few data will either mean we can't interpret our model because several other models would fit just as well, or we can sometimes overfit our data.
- In prediction, we can split any 'ground truth' dataset into training and test data. We use the training data to find our parameters and then see how well it goes in predicting values for test data.

Make sure the data will be able to let you know how effective your approach is. Do some pre-testing to see if you can understand it's behaviour.

FLEXIBILITY AND OVERFITTING



Here is the difference between a properly fitted and overfitted model:

Source: Quora

The overfitted model is not going to be useful unless we apply it to the exact same dataset because no other data will fall exactly along the overfitted line.

In overfitting, a statistical model describes random error or noise instead of the underlying relationship. Overfitting occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. A model that has been overfit has poor predictive performance, as it overreacts to minor fluctuations in the training data.

Measures of accuracy/ goodness of fit measure

Total squared error/sum of squared error
All of the differences between the predicted and observed output values, squared and then added together

$$SSE = \sum_{i=1}^{m} (A_w(x_i) - y_i)^2$$

MEASURES OF ACCURACY

Root mean squared error (RMSE)

After calculating the Total squared error, we can divide by the number of observations (m) and then take the square root

$$RMSE = \sqrt{\sum_{i=1}^{m} \frac{(A_w(x_i) - y_i)^2}{n}}$$

- interpreted as the average difference between each prediction and the output
- it is actually the quadratic mean, a power mean with p = 2, so it will be affected more by larger differences.
- If a fitted function performs better than another in terms of RMSE, then it will also have a lower total squared error.

Measures of accuracy

Total least absolute deviation (LAD) / sum of absolute errors
The sum of all the absolute differences between predicted and observed outputs,

$$SAE = \sum_{i=1}^{m} |A_w(x_i) - y_i|$$

Average L1 error / Average absolute error
This is the SAE divided by the number of observations,

$$Av.AE = \sum_{i=1}^{m} \frac{|A_w(x_i) - y_i|}{n}$$

MEASURES OF ACCURACY

Pearson Correlation (*r*)

- This value between -1 and 1, gives an idea of how close the relationship between the two variables is to a linear relationship.
- A perfect positive relationship will have $r = 1$, a perfect negative relationship will have $r = -1$ and no relationship will correspond with $r = 0$.
- When using Pearson correlation as a goodness-of-fit measure for aggregation functions, we would be not be expecting to find a negative relationship, and in general we want the predicted and observed outputs to be as close as possible

Measures of accuracy

Spearman Correlation (ρ)

- Similar to Pearson’s correlation, the Spearman correlation coefficient (r, pronounced ′rho′) gives an indication of whether there is a monotone relationship between the observed and predicted outputs.
- We may not have a linear relationship such that increases in the observed outputs correspond with equal increases to our predicted outputs, however it may be the case that if one observation y_1 is higher than another y_2, then we might want it to hold that our predicted output for $A_w(x_1)$ is higher than that predicted for $A_w(x_2)$.
- For example, if our observed outputs were $< 0.5, 0.95, 0.3, 0.72 >$ and the predicted outputs were $< 0.7, 0.89, 0.66, 0.72 >$ the Spearman’s rank correlation is r = 1 since the relative orderings in both cases are $y_3 < y_1 < y_4 < y_2$.   More examples: https://bit.ly/2YaHJFf

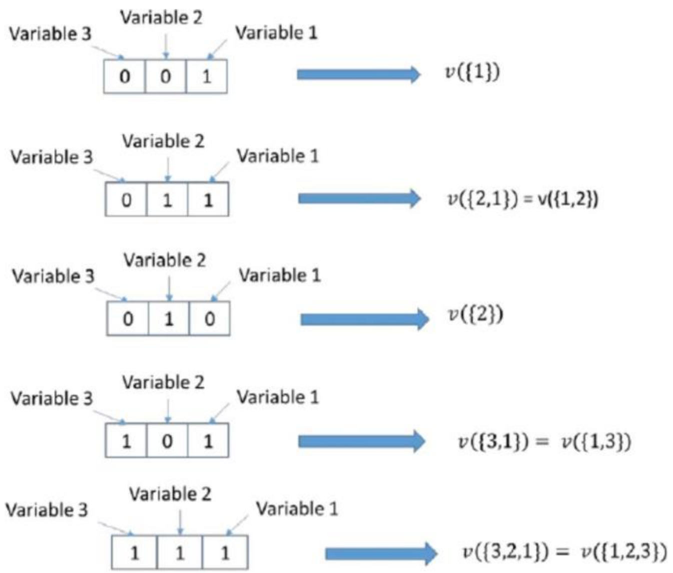
Measures of accuracy

- The evaluation measure we choose to use may vary depending on our application.
- If we plan to use the aggregation model we constructed to rank different alternatives, then we may be more interested in using Spearman correlation, while if we want to create an aggregation function that predicts the average temperature in a room, we may be more interested in RMSE or Av.AE.

BINARY REPRESENTATION OF THE FUZZY MEASURES (CHOQUET INTEGRALS)

- In order to represent the fuzzy measures, Binary ordering can be used.
- Binary ordering uses the form of a binary number to decide which elements are in the set at a given position
- For example, let us consider that we want to represent the following 8 fuzzy measures (with three variables 1, 2, and 3).
  - $v(\varphi), v(1), v(2), v(3), v(1,2), v(1,3), v(2,3), v(1,2,3)$
  - In order to represent these 8 fuzzy measures, we need three digit binary number (hence it has 8 possibilities). The binary numbers are 000, 001, 010, 011, 100, 101, 110, 111
  - We look at where the "1" occurs from the far "right" in order to decide which element/variable is included in the set. See the diagrams below

BINARY REPRESENTATION OF THE FUZZY MEASURES (CHOQUET INTEGRALS)...



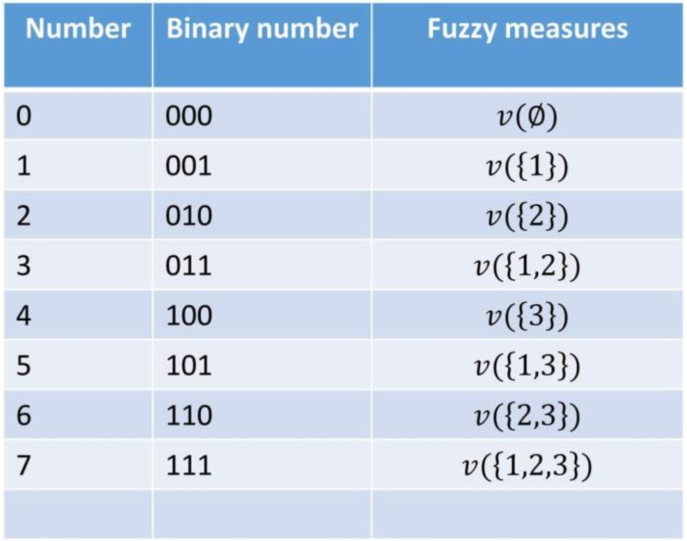
Variable 3 Variable 2 Variable 1
0 0 1 → $v(\{1\})$

Variable 3 Variable 2 Variable 1
0 1 1 → $v(\{2,1\})$ = v({1,2})

Variable 3 Variable 2 Variable 1
0 1 0 → $v(\{2\})$

Variable 3 Variable 2 Variable 1
1 0 1 → $v(\{3,1\}) = v(\{1,3\})$

Variable 3 Variable 2 Variable 1
1 1 1 → $v(\{3,2,1\}) = v(\{1,2,3\})$



Number	Binary number	Fuzzy measures
0	000	$v(\emptyset)$
1	001	$v(\{1\})$
2	010	$v(\{2\})$
3	011	$v(\{1,2\})$
4	100	$v(\{3\})$
5	101	$v(\{1,3\})$
6	110	$v(\{2,3\})$
7	111	$v(\{1,2,3\})$

- The table above shows the full binary representation of the 8 fuzzy measures.
- Remember that we look at where the “1” occurs from the far **”right”** in the binary representation

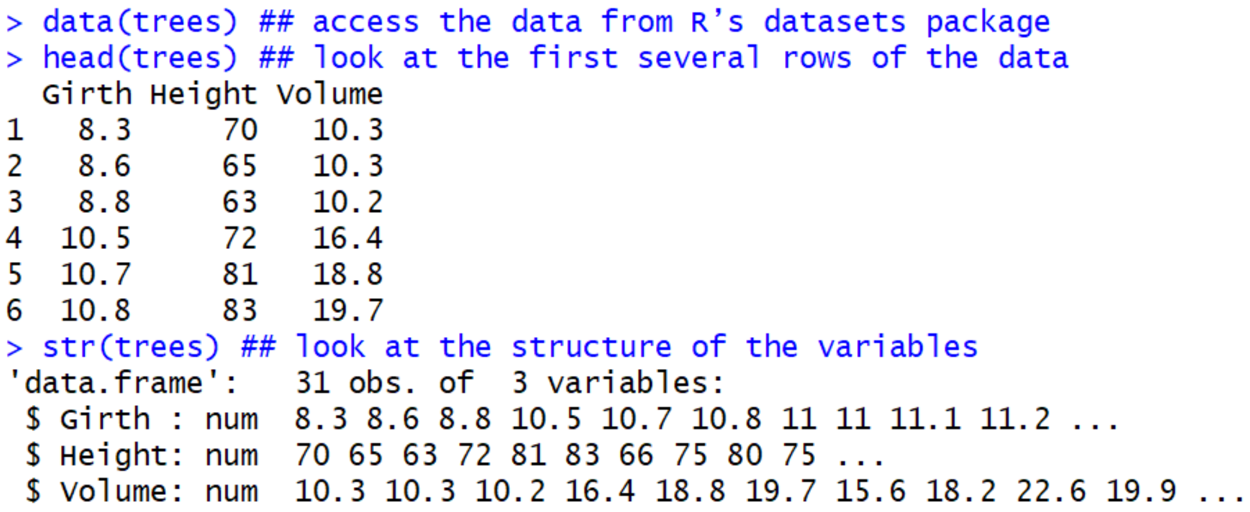
We don't need to input the empty set for the functions provided in workshops, i.e., start with v({1}),v({2}) and so on.

EXAMPLE: FITTING AGGREGATION FUNCTION TO ”KEIHOTELS.TXT” DATA (USE OF PACKAGE ”AGGWAFIT718.R”)

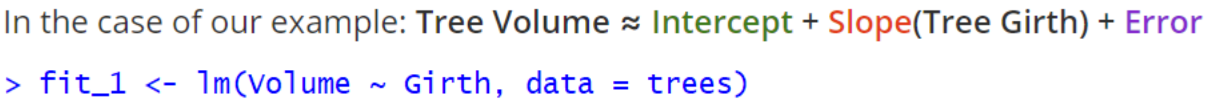
Check Workshop 6 materials for instruction

- To fit aggregation functions, we use the package **AggWaFit718.R**
- Fitting aggregation function to ”KeiHotels.txt” data.
  - Will show a quick demo in R
  - Will be done in this week’s prac as well.
- Another good example using ’bicycle share systems’ (”BikeShare231.txt”) data is provided in section 5.4 of the reference book.

Linear Regression with R



```
> data(trees) ## access the data from R’s datasets package
> head(trees) ## look at the first several rows of the data
  Girth Height Volume
1   8.3     70   10.3
2   8.6     65   10.3
3   8.8     63   10.2
4  10.5     72   16.4
5  10.7     81   18.8
6  10.8     83   19.7
> str(trees) ## look at the structure of the variables
'data.frame':   31 obs. of  3 variables:
 $ Girth : num  8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...
 $ Height: num  70 65 63 72 81 83 66 75 80 75 ...
 $ Volume: num  10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...
```



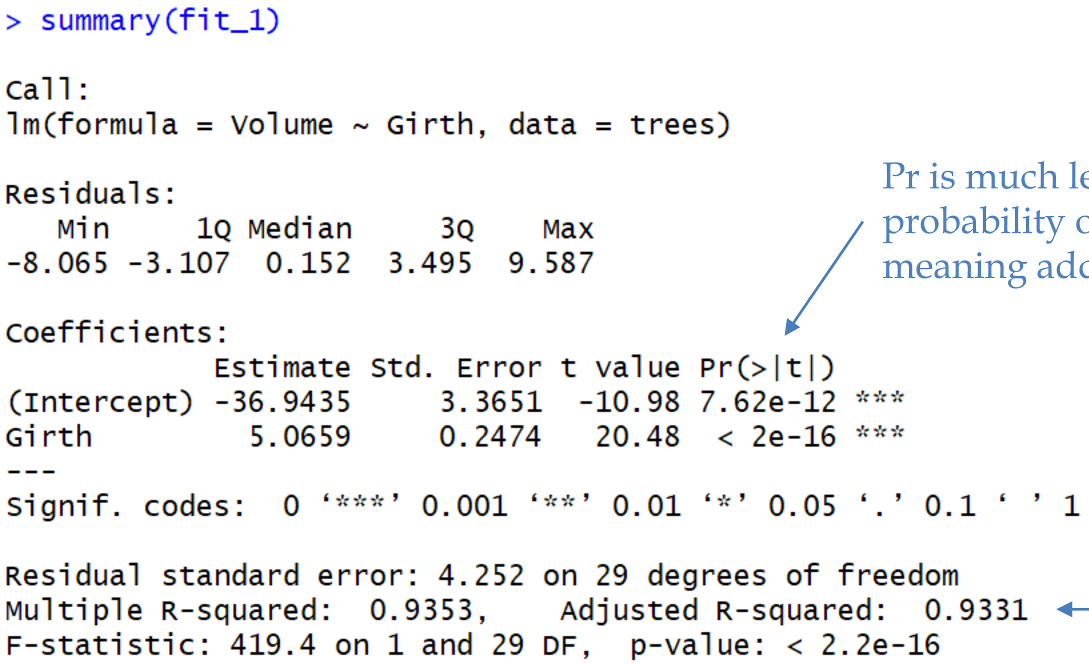
In the case of our example: **Tree Volume ≈ Intercept + Slope(Tree Girth) + Error**

```
> fit_1 <- lm(Volume ~ Girth, data = trees)
```

https://www.dataquest.io/blog/statistical-learning-for-predictive-modeling-r/

Linear Regression with R (cont.)

The model output using **summary()** will provide us with the information we need to test our hypothesis and assess how well the model fits our data.



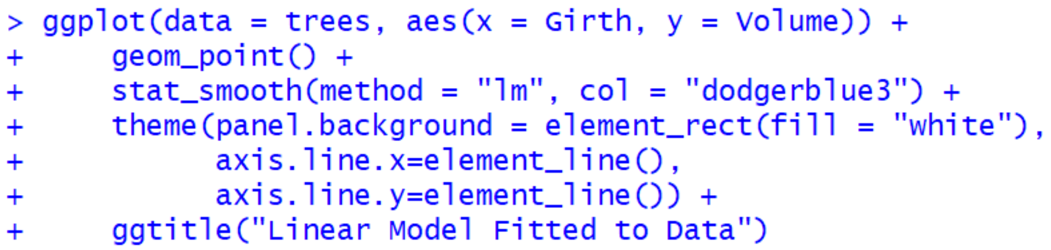
```
> summary(fit_1)

Call:
lm(formula = Volume ~ Girth, data = trees)

Residuals:
   Min      1Q Median      3Q     Max 
-8.065 -3.107  0.152  3.495  9.587 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -36.9435     3.3651  -10.98 7.62e-12 ***
Girth         5.0659     0.2474   20.48  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.252 on 29 degrees of freedom
Multiple R-squared:  0.9353,	Adjusted R-squared:  0.9331 
F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16
```

Pr is much less than 0.05, we can say with a 95% probability of being correct that the variable has a meaning addition to the fitted model

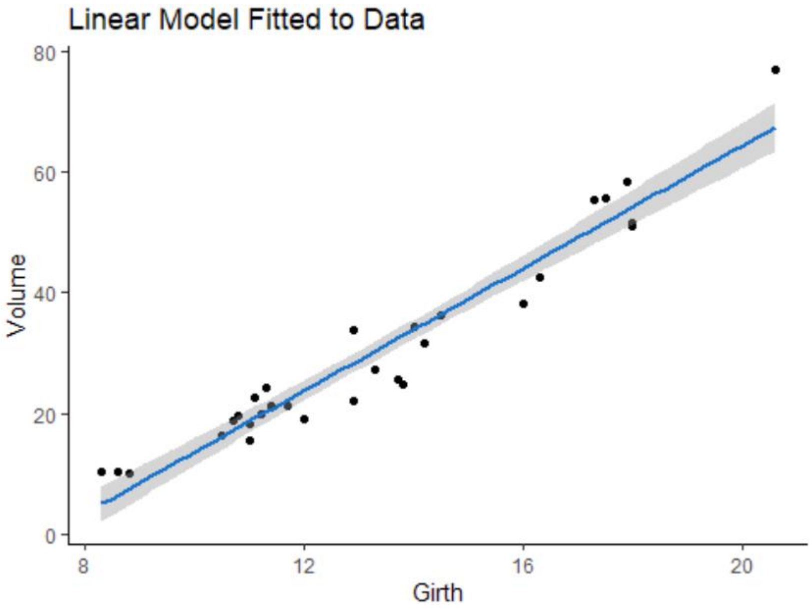
R-squared measures how close the data is to fitting the regression line

A p-value less than 0.05 rejects the null hypothesis that no linear correlation between variables, i.e., the results shows that there is significant positive linear correlation between the variables.

Linear Regression with R (cont.)



```
> ggplot(data = trees, aes(x = Girth, y = Volume)) +
+     geom_point() +
+     stat_smooth(method = "lm", col = "dodgerblue3") +
+     theme(panel.background = element_rect(fill = "white"),
+           axis.line.x=element_line(),
+           axis.line.y=element_line()) +
+     ggtitle("Linear Model Fitted to Data")
```



The gray shading around the line represents a confidence interval of 0.95. This 0.95 confidence interval is the probability that the true linear model for the girth and volume of all black cherry trees will lie within the confidence interval of the regression model fitted to our data. Even though this model fits our data quite well, there is still variability within our observations.